AWS Security Automation with GuardDuty

Cilla Mary Mathew, Divya Gorivale
Guide: Prof. Bindy Wilson
Model College

Abstract- Security Automation is a Crucial element of todays security architecture which includes monitor, detect and respond. Security Automation should be a vital part of any organization security posture. AWS GuardDuty already does the Intelligent part of Processing the Logs & Events, Apply Machine Learning to detect Malicious Activity and Display it on a Dashboard.

The goal is to automate this process and alert the security team about any new inciident identified by the GuardDuty, this way the security team can organize and prioritize the recent critical incident and also work on detection respond. Security Automation will enable monitoring, detection and response to critical security threats and activities.

Index Terms- AWS GuardDuty Automation, AWS Security, Security Automation, Automate GuardDuty Findings, GuardDuty Slack Alerts, GuardDuty Teams Alert.

I.INTRODUCTION

AWS is one of the Top Cloud Providers where organizations deploy their infrastructure, when people/organizations move to AWS Cloud, they make use of the AWS Resources which are broadly divided into Compute, Storage, Networking & Database. When the Organisation has set up its Application & Infra on AWS Cloud, it is extremely important to have good visibility and be aware of what is happening (i.e what kind of activity is happening)

Example:

1) Have any of the Server connecting to malicious servers or C2C servers
2) Is my S3 Bucket public, has someone accessed it
3) Any Malicious activity in the Network
4) DNS Exfiltration

AWS GuardDuty is AWS Cloud Native Intelligent IDS Solution that takes input from VPC Flow Logs, DNS Logs, AWS CloudTrail, and S3 Data Events.

AWS GuardDuty already does the Intelligent part of Processing the Logs & Events, Apply Machine Learning to detect Malicious Activity, and Displaying it on a Dashboard.

But our Automation Solution will create Alerts on GuardDuty Finding, Alert on Slack & on email.

II.OBJECTIVES

Enable faster incident response and to increase security agility.

Greatly enhance detection and remediation

III.AWS GUARDDUTY

GuardDuty, is a threat detection service. Threat detection, a cybersecurity practice that continuously monitors a system for malicious activity and generates alerts and security events. With GuardDuty, teams can monitor AWS resources and receive alerts and notifications about potential threats. Security teams respond to these notifications and take preventative measures to protect your infrastructure and AWS cloud resources.

IV.HOW GUARDDUTY WORKS?

The threat detection service relies on logs and traces to monitor the system. GuardDuty logs and events are provided by AWS CloudTrail, VPC flow logs, and DNS logs. CloudTrail logs all actions taken with your AWS resources either through the CLI, GUI console, or AWS API. VPC Flow Logs captures and records information about IP traffic traversing network interfaces. AWS Route 53, a DNS service from AWS, generates DNS logs when your EC2 instances use Route 53 for name resolution. GuardDuty implements a comprehensive set of internal rules to analyze these logs and traces to identify threats. As an AWS user, you don't have to worry about these internal rules because AWS takes care of this complexity entirely. However, when you rely on GuardDuty, your team must consider certain configuration and operational settings. We discuss these considerations in the best practices below.

#1: Ensure Full Detection Coverage

GuardDuty uses logs from AWS CloudTrail, VPC flow logs, and DNS logs to describe pitfalls. When you enable GuardDuty, it automatically starts testing data from all these data sources. However, GuardDuty does not automatically enable or manage any configuration in any of these data sources. therefore, it is your responsibility to ensure that all AWS checkouts generate the necessary logs and events for GuardDuty to anatomize. CloudTrail is automatically enabled when you create an AWS account and records all operational events, also known as aircraft control events. CloudTrail will record a functional modification to your AWS vault, like creating a new EC2 instance, attaching a new policy to a stoner, or creating a new subnet, etc. The logs of these operations will be automatically available to GuardDuty without new configurations in CloudTrail. however, VPC flow protocols that cover IP packets transmitted between EC2 instances can be enabled for a VPC, a subnet, or each network interface in a subnet. However, GuardDuty will not gain full visibility into your VPC business if VPC flow protocols are configured for only a subset of your network interfaces. Your squad must ensure that VPC flow protocols are enabled for all areas and necessary interfaces that you plan to cover against traps. gusto ComplyOps can help your squad configure security settings and cover configuration, including logging and intrusion detection settings. AWS Route 53 logs are generated whenever an EC2 instance queries the Route 53 service for name resolution. However, DNS queries form that EC2 instances will not be recorded on Route 53 if you configure Google DNS or any other external DNS garçon address as a nameserver in the EC2 instance. Therefore, GuardDuty will not be suitable for parsing DNS queries from this case. Your squad may consider using AWS Route 53 with EC2 instances for name resolution and other events in GuardDuty.

#2: Enable CloudTrail S3 Data Event Monitoring

In addition to the management events described above, CloudTrail is able to collect events for S3, are known as S3 data events.

CloudTrail automatically accounts for management events in S3 including operations on S3 buckets such as `ListBuckets` or `DeleteBuckets`. But operations such as `GetObject`, `ListObjects`, and `DeleteObject`, which are performed on objects stored inside buckets may be enabled through CloudTrail S3 data events.

When you enable GuardDuty it will start monitoring CloudTrail management events but monitoring of CloudTrail S3 data events will not be enabled by default. You must manually configure CloudTrail S3 data events as a data source in GuardDuty. This can be configured from the GuardDuty console or the API. After this setting is configured, GuardDuty will start monitoring S3 data events for potential threats.

#3: Enable GuardDuty for All Regions

While your team may be running your workloads in only in a limited number of AWS regions, you must enable GuardDuty for all regions, for complete threat visibility.

When GuardDuty is enabled for all regions, it can detect unintended activities in unused regions also. As an example, a malicious user may deploy a new EC2 instance in a region that you have never used. GuardDuty can detect and alert you for such incidents when it is enabled for all regions.

#4: Secure Access to GuardDuty With IAM

Amazon GuardDuty, like all other AWS services, integrates with AWS IAM for authentication and permissions. Because GuardDuty is important to the security of your resources, you must ensure that access to GuardDuty is restricted to specific users only. You must also grant separate permissions to GuardDuty users and administrators.

Organizations must determine which services and resources are required by users using GuardDuty and grant them access based on their job role. Security teams may consider defining a team member to act as a GuardDuty administrator to manage and resolve GuardDuty findings in your organization.

#5: Grant Least Privileges

Granting least privileges ensures that the user has only enough privileges to perform the tasks they are supposed to do. It can prevent human error and also limits what an attacker can do if a user account is compromised.

GuardDuty supports AWS IAM identity-based policies. Identity-based policies are attached to an

IAM identity, such as a user or group. These policies grant users permission to perform certain actions on an AWS resource.

By default, your IAM users will have no permissions to manage GuardDuty resources. When starting GuardDuty, you must grant permissions starting with the lowest permissions. You can then incrementally grant permissions as needed by attaching additional policies.

#6: Analyze GuardDuty Activities with CloudTrail

Just as GuardDuty helps find threats, you need to make sure that no users tamper with GuardDuty unknowingly or maliciously. you'll be able to do that by observance CloudTrail events regarding GuardDuty.

As delineated earlier, CloudTrail logs all operational activity on AWS resources. By default, these logs ar keep for ninety days and you'll be able to read them within the CloudTrail console. you'll be able to more analyze these logs by making trails at intervals CloudTrail. making a trace permits you to send events to CloudWatch for observance and trigger log queries with different services like AWS Greek deity.

So as a best observe, you would like to make a GuardDuty path in CloudTrail and thoroughly monitor all user activities on GuardDuty.

#7: Automate Risk Mitigation with CloudWatch and Lambda

GuardDuty generates a finding once it detects a threat. A finding could be a potential issue that will demand bound actions to mitigate the risks. will| you'll| you'll be able to} read the findings within the GuardDuty console then can take action for every finding.

However, as your team manages massive cloud environments with additional resources, manually analyzing and addressing findings in GuardDuty will become tougher. Your security team might think about searching for opportunities to automatise risk mitigation. integration GuardDuty with AWS CloudWatch Events and Lambda provides team with a good thanks to automatise this method.

GuardDuty will generate CloudWatch Events for every finding. Your team will produce rules/filters in CloudWatch to trigger AWS Lambda functions for various kinds of events. By connecting CloudWatch Events from GuardDuty to Lambda functions, your team will write code to mechanically take corrective actions for every form of GuardDuty finding.

#8: Integrate GuardDuty with AWS Security Hub

In addition to GuardDuty, AWS has alternative cybersecurity services like AWS Config, Amazon Macie, Amazon Inspector, AWS Firewall Manager, etc. AWS Security Hub will integrate with of these services as well as GuardDuty to supply you with a comprehensive read of your security standing.

AWS Security Hub also can mixture security findings from numerous services as well as GuardDuty. By integration GuardDuty with Security Hub, you get to watch your security findings in a very a lot of holistic manner than observance individual services one by one.

#9: Use Suppression Rules To Archive Unnecessary Findings

After facultative GuardDuty, you'll be able to read findings within the GuardDuty console. a number of these findings may well be false positives wherever they are doing not represent actual threats. To filter out any litter, you'll be able to use suppression rules in order that sure findings don't seem to be displayed within the console.

Suppression rules enable you to filter findings by finding kind, specific AWS resource, or additional granular criteria like information science address, ports, etc. Once you produce a suppression rule, the finding won't be displayed on the GuardDuty console. This finding is archived and keep in GuardDuty for ninety days.

It is suggested to form suppression rules incrementally rather than making them direct. you ought to monitor your findings and make rules to suppress specific findings that add noise to your security method. this may make sure that you are doing not suppress real threats with suppression rules.

#10: Configure A Trusted IP List

A sure scientific discipline list could be a list of scientific discipline addresses or subnets you'll be able to outline in GuardDuty, so GuardDuty doesn't generate findings for sure IPs. this will be helpful in an exceedingly hybrid cloud design wherever your EC2 instances are oftentimes communication with sure on-premises resources. you'll be able to trust these on-premises scientific discipline addresses, by process a sure scientific discipline list in GuardDuty. making a sure scientific discipline list will facilitate your team scale back some noise and false positives created in GuardDuty findings.

V.AWS SERVICE INTEGRATIONS WITH GUARDDUTY

Integrating GuardDuty with AWS Security Hub
AWS Security Hub collects security data from your AWS accounts, services, and supported third-party partner products to assess the security posture of your environment against industry standards and best practices. In addition to assessing your security posture, Security Hub creates a central location for discovering all of your integrated AWS services and AWS partner products. Enabling Security Hub with GuardDuty will automatically allow GuardDuty discovery data to be received into Security Hub.

Integrating GuardDuty with Amazon Detective
Amazon Detective uses log knowledge from your AWS accounts to make knowledge visualizations for your resources and information science addresses interacting together with your setting. Detective visualizations assist you quickly and simply investigate security problems. Once each services square measure enabled, you'll go from trying up GuardDuty details to Detective console info.

What can GuardDuty detect?
GuardDuty offers you access to inherent detection techniques developed and optimized for the cloud. Detection algorithms area unit maintained and endlessly improved by GuardDuty Engineers. Primary detection classes embrace the following:
Reconnaissance: Activity indicating intelligence operation by Associate in Nursing aggressor, like uncommon API activity, port scanning within a VPC, uncommon patterns of failing login requests, or unblocked port discovery from a famed dangerous scientific discipline address.
Instance Compromise: Activity indicating instance compromise like cryptocurrency mining, domain generation formula (DGA) malware, departing denial of service activity, unco high volume of network traffic, uncommon network protocols, departing instance communication with a famed malicious scientific discipline address, temporary EC2 credentials employed by Associate in Nursing external scientific discipline address, and knowledge exfiltration exploitation DNS.
Account Compromise: Common patterns indicating account compromise, as well as API calls from Associate in Nursing uncommon geolocation or anonymizing proxy, tries to disable CloudTrail work, uncommon instance or infrastructure startup, infrastructure preparation in Associate in Nursing uncommon region, written document exfiltration, and API calls from a famed malicious scientific discipline address.
Bucket compromise: Activity indicating a bucket compromise, such as suspicious data access patterns indicating credential misuse, unusual S3 API activity from a remote host, unauthorized S3 access from known malicious IP addresses, and API calls to retrieve data in S3 buckets from a user that had no prior history of accessing the bucket or invoked from an unusual location. GuardDuty continuously monitors and analyzes CloudTrail S3 data events (like GetObject, ListObjects, and DeleteObject) to detect suspicious activity across all of your S3 buckets.
Malware detection: GuardDuty begins a malware detection scan when it identifies suspicious behavior indicative of malicious software in EC2 instance or container workloads. GuardDuty generates temporary replicas of EBS volumes attached to such EC2 instance or container workloads and scans the volume replicas for trojans, worms, crypto miners, rootkits, bots, and more, that might be used to compromise the workloads, repurpose resources for malicious use, and gain unauthorized access to data. GuardDuty Malware Protection generates contextualized findings that can validate the source of the suspicious behavior. These findings can be routed to the proper administrators and initiate automated remediation.
Container compromise: Activity identifying possible malicious or suspicious behavior in container workloads is detected by continuously monitoring and profiling Amazon EKS clusters by analyzing its Amazon EKS audit logs.

VI. RESEARCH METHODOLOGIES

A model can include both descriptive and analytical components. A descriptive model’s logical relationships can be examined and conclusions can be drawn. The logical analysis draws quite different conclusions than the quantitative investigations of the properties. We conducted an online poll utilizing an online form creator and data collection service to acquire information regarding people’s awareness.

VII. PUBLIC SURVEY

We deployed our data gathering facility, to a variety of people of all age ranges to collect information on various faces of their understanding of how LiFi will help to make their life better.

VII.I QUESTIONNAIRE

What is your age range ?
Are you aware of AWS ?
Do you think AWS security will be more helpful?
What do you think which is more better ? AWS or any Antivirus
Do you believe that by automating security via AWS would be more secure than manual process?

VII.II RESPONSES

The major age group who responded id from 18-40 years (71%) followed by 40-65 years (16%) and people less than 18 years (7%)

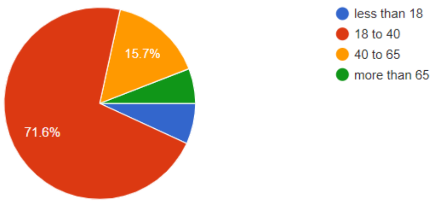

Chart 1: Age group

The respondents who knew about AWS were 71% of the total and 15% were somewhat aware of AWS and the rest were not aware of AWS.

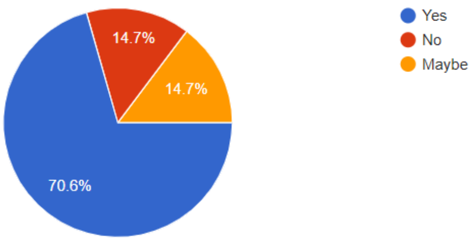

Chart 2: AWS awareness

72% of the respondents thought that AWS was more secure than Antivirus and the rest favored Antivirus being more secure.

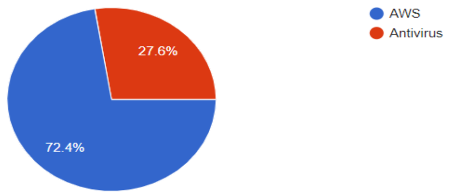

Chart 3: More secure

64% of the respondents believed that AWS security would be more helpful whereas 32% were unsure about it.

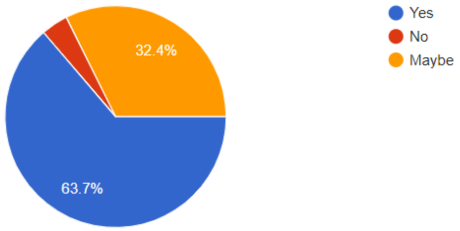

Chart 4: AWS security being more helpful

67% of the respondents thought that by automating security via AWS would be more secure and 27% thought that manual process would be more secure

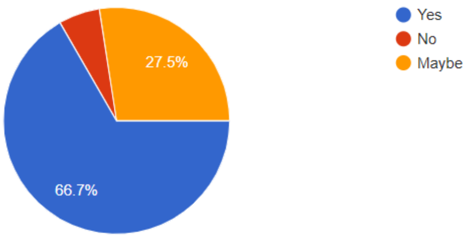

Chart 5: more secure

VIII. HYPOTHESIS TESTING

Hypothesis testing is a systematic process of determining whether research findings support a particular theory that applies to humans. The hypothesis test uses sample data to test a population-based opinion.

The hypothesis testing examines how rare the outcome is, whether the variance is reasonable or whether the outcome is too extreme to be considered a variance of luck.

There are two types of hypotheses:
Null Hypothesis (denoted H0)
Alternate or Research Hypothesis (denoted Ha)

For this paper,
H0: AWS security can improve security
Ha: AWS security cannot improve security

TEST (STATISTICS)

There are three tests available to determine if the null hypothesis is to be accepted or not. They are:

1) Chi-squared Test
2) T-Student test (T-test)
3) Fisher's Z Test

In this paper, I'll be using the Two-Tailed T-student test.

A t-test is an inferential statistic that determines if there is a significant difference in the means of two groups that are related in some manner.

→ Level of Significance

The chance of rejecting the null hypothesis when it is true is the significance level (also known as alpha or α). A significance level of 0.05, for example, means there's a 5% probability of discovering a difference when there is none. Lower significance levels indicate that more evidence is required to reject the null hypothesis.

→ Level of Confidence

The Confidence level indicates the probability that the location of a statistical parameter such as the arithmetic means measured in the sample survey is also true for the entire population.

Sr. No	Data
1	71.6
2	70.6
3	72.5
4	63.7
5	66.7
Mean (x)	69.02
Standard Deviation (s)	3.706

Level of Significance =0.05 i.e. 5%

Level of Confidence= 95%

A t-score (t-value) is the number of standard deviations away from the mean distributions.

The formula to find the t-score is:

$t=(x-\mu) / (s/\sqrt{n})$

where x is the sample mean,
μ is the hypothesized mean,
s is the sample standard deviation,
and n is the sample size.

The Probability value, also known as the p-value, indicates how probable your data is under the null hypothesis. Once we have the value of 't', we can calculate the p-value. If the p-value is less than the alpha level, then we can reject the null hypothesis and conclude that Li-Fi cannot improve healthcare.

→ Calculating 't' value:

Step 1: Determine the Null and Alternate hypothesis
Null Hypothesis (H0): Li-Fi can improve healthcare
Alternate Hypothesis (Ha): Li-Fi cannot improve healthcare
Step 2: Find the test statistic:
In this case, the hypothesized mean is considered 0.
Therefore,

$t=(x-\mu) / (s/\sqrt{n}) = (69.02-0)/(3.706/\sqrt{5})$
$=41.644$

t-value=41.644

→ Calculating p-value:

Step 3: Calculate the test statistic's p-value.

The t-distribution table with n-1 degrees of freedom is used to calculate the p-value. In this paper, the sample size is n=5, so n-1=4.

By feeding the observed value into the calculator, we got the p-value which is 0.00000198. The p-value is less than 0.00001.

Since this p-value is less than our selected alpha level of 0.005, we can reject the null hypothesis. Therefore we can conclude that we have enough evidence to say that AWS security cannot improve security

IX. CONCLUSION

AWS alone might not improve the security but it can help us detect and prevent all possible threats and which would help us in making our organizational or personal systems more secure.

Threat detection is an important aspect of cybersecurity, More Important is to Automate it using AWS Cloud Native Services like Lambda & EventBridge. The best practices described above will help you build a robust threat detection solution with GuardDuty for your resources in AWS.

Once AWS GuardDuty is enabled it is important to work on the monitoring & alerting part, where the security team can be notified of any such incident on AWS.

REFERENCE

[1] https://aws.amazon.com/guardduty/

[2] https://aws.amazon.com/guardduty/faqs/

[3] https://aws.amazon.com/guardduty/resources/

[4] https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_integrations.html

[5] https://docs.amazonaws.cn/en_us/guardduty/latest/ug/infrastructure-security.html

[6] https://docs.aws.amazon.com/whitepapers/latest/architecting-hipaa-security-and-compliance-on-aws/amazon-guardduty.html

[7] https://aws.amazon.com/blogs/security/new-third-party-test-compares-amazon-guardduty-to-network-intrusion-detection-systems/

[8] https://maturitymodel.security.aws.dev/en/whitepapers-faq/whitepapers/

[9] https://d1.awsstatic.com/whitepapers/aws_security_incident_response.pdf

[10] https://www.dashsdk.com/resource/best-practices-for-amazon-guardduty/

[11] https://aws.amazon.com/solutions/case-studies/volkswagen-group-guardduty/

[12] https://aws.amazon.com/blogs/security/automatically-block-suspicious-dns-activity-with-amazon-guardduty-and-route-53-resolver-dns-firewall/

[13] https://aws.amazon.com/blogs/security/how-get-started-security-response-automation-aws/

[14] https://d1.awsstatic.com/events/reinvent/2019/REPEAT_1_Automating_threat_detection_and_response_in_AWS_SEC301-R1.pdf

[15] https://github.com/aws-samples/amazon-guardduty-automated-response-sample