**Auditing Your Security with AWS Trusted Advisor**

**Introduction**

I embarked on a hands-on project as part of a free lab provided by AWS. This project aimed to offer practical experience and insights into AWS resources and best practices. Through a series of tasks and challenges, I delved into AWS Trusted Advisor, Amazon EC2 Security Groups, and the implementation of Multi-factor Authentication (MFA). These hands-on activities not only enriched my understanding of AWS but also allowed me to put theory into practice, improving my skills and knowledge in cloud management and security.

Amazon Web Services (AWS) is a global leader in cloud computing, offering a multitude of powerful and versatile services. However, the power and versatility of AWS come with a corresponding need for diligent security measures. AWS Trusted Advisor is a fundamental tool in your arsenal for achieving this goal.

AWS Trusted Advisor is an intelligent service that assesses your AWS environment, analyzing configurations and usage patterns. It identifies areas where security might be compromised, cost optimization could be achieved, performance could be enhanced, and fault tolerance could be increased.

**Objectives**

* Use Trusted Advisor to perform a basic audit of your AWS resources
* Modify Amazon Elastic Compute Cloud (Amazon EC2) Security Groups to meet best practices
* Configure Multi-factor Authentication (MFA) (Optional, requiring installation of software on a mobile device)

**Task description**

**Task 1: *Check recommended actions with Trusted Advisor***

1. Accessed the AWS Management Console and used the search bar to look for "Trusted Advisor."
2. Selected the Trusted Advisor service from the list of search results.
3. On the Trusted Advisor Recommendations page, I reviewed the Checks summary and the list of Recommended Actions, which are based on pre-defined checks for the AWS account. These recommended actions are categorized into three types:(Red) Critical – Action recommended (Orange) Investigation recommended (Green) No issues or concerns found
4. To gain further insights, I expanded each AWS Trusted Advisor check and examined the details. If there were items that were not green, I reviewed the criteria for the status and considered the provided Recommended Actions.

**Task 2: *Modify security groups with unrestricted ports***

1. In the Trusted Advisor navigation pane on the left side of the page, I chose "Security" under the "Recommendations" menu.
2. Under the "Checks" section, I expanded "Security Groups - Specific Ports Unrestricted.
3. "In the "Security Groups" table at the bottom of this section, I identified the Security Group(s) marked as "Red" in the Status column.
4. One of the items listed indicated that there was an unrestricted protocol/port (tcp/port 21). I received information from the security staff that this port was not currently needed and should be removed from the rules.
5. To locate the item that contained "tcp" in the Protocol column and "21" in the From Port column, I looked through the list. If I couldn't find the item with "tcp/port 21," I refreshed the "Security Groups - Specific Ports Unrestricted" item using the provided button.
6. Once I found the item, I chose the "WordPress Security Group" link to open the Security Groups page.
7. On the Security Groups page, I selected the "Inbound rules" tab.
8. Then, I clicked "Edit inbound rules" and followed these sub-steps:

For the rule related to port 21, I selected "Delete" to the right of the rule.

Finally, I selected "Save rules" to apply the changes.

The output



**Task 3**: ***Modify Security Groups to Restrict Access***

1. I identified the "tcp/port 3306" security group that was flagged as "Red" by Trusted Advisor. This group was allowing unrestricted access to an Amazon RDS MySQL database on port 3306/tcp.
2. I chose the "MySQL Security Group" link to open the Security Groups page.
3. On the Security Groups page, I selected the "Inbound rules" tab.
4. Then, I clicked "Edit inbound rules" and noticed that the rule was permitting incoming traffic to port 3306 from "0.0.0.0/0," which meant traffic was allowed from any computer on the Internet. This is considered poor security practice for a database.
5. To improve security, I removed the rule for port 3306 by choosing "Delete" to the right of the rule.
6. Next, I chose "Add rule" and followed these additional sub-steps:

For "Type," I selected "MySQL/Aurora," and I noticed that the "Protocol" and "Port range" fields were automatically populated with "TCP 3306."

For "Source," I chose "Custom," and in the search field, I entered "sg" and then selected the "Web Security Group."

Output



**Task 4: *Configure multi-factor authentication (MFA)***

I setup my MFA virtual Device (Google Authenticator) then I

1. I opened a new private browser window and pasted the "Login URL" to access the AWS Management Console login page.
2. On the AWS Management Console login page, I entered "user-1" for the IAM user name. I copied the "Administrator Password" value from the Resources pane and pasted it into the Password field. I clicked "Sign in" to proceed.
3. The Multi-factor Authentication (MFA) page opened, and I entered the 6-digit code displayed in my MFA app for the "MFA Code."
4. After entering the MFA code, I clicked "Submit."
5. If I successfully logged in to the AWS Management Console, it indicated that the MFA virtual device had been integrated with IAM for "user-1."Following a successful login, I closed the browser's private window and returned to the original AWS Management Console page to continue the lab.

Task 5: Exclude Security Groups if Unrestricted Access is Required

**Challenges I encountered**

1. Complex Security Group Configurations: Modifying Amazon EC2 Security Groups to align with best practices presented a challenge due to the complexity of the configurations in my environment. I had to carefully assess and plan the necessary rule changes to maintain security while ensuring that legitimate access was not disrupted. Managing numerous rules and verifying their correctness was time-consuming and required meticulous attention to detail.
2. Resource Dependency and Impact: Making changes to security configurations and implementing Multi-factor Authentication (MFA) can have a ripple effect on various resources and users within the AWS environment. I had to consider the interdependencies between different resources and the potential impact of security changes on existing services. Ensuring that these changes didn't disrupt critical operations or lead to unintended consequences required careful planning and testing.

**Conclusion**

In conclusion, I used AWS Trusted Advisor to perform a basic audit of my AWS resources. I also took the initiative to modify Amazon EC2 Security Groups to meet best practices, and I configured Multi-factor Authentication (MFA). These steps have significantly improved the security and management of my AWS resources.