Disabling inactive user accounts in AWS

Your organization relies on the Identity and Access Management (IAM) service of AWS to keep track of all the users, groups, roles, and policies that provide access to data and resources. Over time, there are often personnel changes within the organization as users change roles or leave the company. When this happens, the user accounts may not get updated with the correct permissions or get deleted from IAM if the user is no longer an employee. You know that stale user accounts, whether they are for humans or applications, are a huge security risk for any organization. You would like to create a semi-automated process that is repeatable and extensible for deleting inactive users.

Required data

AWS IAM Service

How to use Splunk software for this use case

After you identify an AWS user account as a problem and decide it shouldn’t be there, this playbook will run and disable it. The playbook starts with a filter that checks the allowlist, which is a custom list containing AWS usernames that should not be disabled. Next, the playbook prompts an analyst with the names of the user accounts to be disabled. If the analyst confirms the action, the final block deletes the appropriate login profiles on the AWS console and disables the associated access keys.

If you have not already configured the AWS IAM connector on Splunk SOAR and identified a list of inactive users, you will need to do so first using a separate playbook.

To use the playbook:

1. In the AWS Console, navigate to the IAM service, choose Policies on the sidebar, and edit the policy used by the AWS IAM app on Splunk SOAR to include the following permissions:
   - iam:CreateLoginProfile
   - iam:DeleteLoginProfile
   - iam:UpdateAccessKey
2. In Splunk SOAR, navigate to the Playbooks listing page and select Custom Lists on the top bar. Press +List to create a new custom list called “aws_inactive_user_allowlist”. For each trusted user account you don’t want to disable, create a new row with the username in the first column. You can leave the list empty at first while establishing a baseline.

Next steps

After this playbook has been in use for a trial period, you might want to add more automated steps. For instance, a
Splunk Enterprise query could reveal other accounts or resources associated with users that have left the organization or changed teams. To make this playbook more proactive, you could also connect it to a more robust process for enforcing least privilege, such as a Jira ticket that is created to exhaustively enforce least privilege any time an organizational change occurs.

The content in this article comes from a previously published blog, one of the thousands of Splunk resources available to help users succeed. In addition, these resources might help you understand and implement this guidance:

• Tech Talk: Splunk SOAR playbook – Finding and disabling inactive users on AWS
• GitHub: SOAR playbooks

The information provided in Splunk Lantern is intended for informational and educational purposes only. All information is provided in good faith, however, Splunk disclaims any and all representations and warranties, express and implied, regarding the information provided, including without limitation any warranties and representations regarding the completeness, adequacy or accuracy of the information. You agree to take full responsibility for the results arising from the use of the information provided.