Exploit Elastic Container Service (ECS) Task Definitions(version 1.0)

**Cloud Service Label: IaaS, PaaS**

Description

Adversaries may attempt to exploit the task definitions that are run via Elastic Container Services on AWS. This tactic can be used to exfiltrate information or escalate privileges. ECS is AWS’s container management service which is capable of running, stopping, and managing Docker containers on a cluster.

Task definitions are used to define one or more containers that are to be run and are required to run Docker containers in ECS. A task definition can specify parameters such as which Docker image you want to use, the IAM role to be used, logging configurations, etc.. These task definitions can be exploited to allow for malicious task definitions to be sent.

Examples

|  |  |
| --- | --- |
| **Name** | **Description** |
| Rhino Security Labs Blog Post (Pacu Tool) | This blog post outlines an attack where an adversary starts with a low-level role with access to ECS and then finds a task role that has permissions that are elevated to what they need. A task definition is edited to be malicious and run a command to pull a shell script from a server being hosted by the adversary. A shell script payload to exfiltrate credentials is created and then using the AWS CLI is used to run a command that is used to run the malicious task definition, this is done using run-task API. The adversary can then receive exfiltrated credentials and use them to continue attacks. |

Mitigations

|  |  |  |
| --- | --- | --- |
| **Mitigation** | | **Description** |
| Least Privilege | | Ensure that the Task Roles attached to ECS task definitions are following the principle of least privilege. |
|  | *AWS* | To implement least privilege in an AWS environment IAM policies will be used. This gives the ability to allow users to perform list, read, write, permissions management, or tagging actions. AWS suggests utilizing *last accessed information* and A*WS CloudTrail event history* to get a better understanding of privileges that might be needed or reduced based on a specific role. Full details can be found at **https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#grant-least-privilege.** |
|  | *Azure* | To implement least privilege in an Azure environment Azure Active Directory roles will be used. Azure outlines different tasks and the least privileged role that are suggested to be associated with the task. Those details can be found at: **https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/roles-delegate-by-task.** To learn how to assign specific roles it can be done via the Azure Active Directory Portal. Instructions on how to assign roles can be found here: **https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-manage-roles-portal.** |
|  | *GCP* | To implement least privilege in GCP it is recommended to use predefined roles (which allow for granular access permissions) instead of primitive roles (roles/owner, roles/editor, and roles/viewer). Full details on the difference between types of roles can be found here: **https://cloud.google.com/iam/docs/understanding-roles.** To assign these roles IAM service accounts are used and complete details can be found at: **https://cloud.google.com/iam/docs/using-iam-securely#least\_privilege.** |
| Task Definition Edit Privileges | | If it is necessary to have a task definition run a role that requires an elevated level of permission, ensure that that task definition cannot be altered by everyone. To properly configure the roles given for task definitions, IAM roles will be used. They can specify which users have permissions to read, write, and execute the task definitions. To create the IAM roles for the task the IAM console will be used. On the console a role will be created for an AWS service (Elastic Container Service). Then the permissions to be given will be applied via Tags. Once the role is created it can be attached to the users needing the permissions. More details can be found here: https://docs.aws.amazon.com/AmazonECS/latest/userguide/task-iam-roles.html. |

Detection

This can be difficult to detect due to the fact that once the attack is completed the malicious task definition created is deregistered and the environment is in the same state as it was before the attack occurred.

References

1. https://github.com/RhinoSecurityLabs/pacu/tree/master/modules/ecs\_\_enum\_task\_def. Accessed June 30, 2020.
2. https://rhinosecuritylabs.com/aws/weaponizing-ecs-task-definitions-steal-credentials-running-containers/. Accessed June 30, 2020.