Roles Create by Multicloud Defense

Roles Created by Multicloud Defense

When you onboard a cloud service account to Multicloud Defense Controller with the provided script, user roles are created within the parameters of the cloud service provider to ensure that communication between the services are protected. Depending on the cloud service provider, different roles and permissions are created.

The following roles are created when you onboard an account.

AWS IAM Roles

This document explains the details of the IAM roles created by the CloudFormation template used in the previous section.

The CloudFormation template creates the following three IAM roles and one CloudWatch Event rule:

- **Multicloud DefenseControllerRole** - Used by the Multicloud Defense to connect to your AWS cloud account.
- **Multicloud DefenseFirewallRole** - Used by the Multicloud Defense instances running in your cloud account to access S3, SecretsManager, KMS.
- **Multicloud DefenseCloudWatchEventRole** - Used by the CloudWatch Event Rule to transfer inventory changes to the Multicloud Defense.
- **Multicloud DefenseCloudWatchEventRule** - A rule created on CloudWatch Events to transfer inventory changes to the Multicloud Defense. The rule assumes the Multicloud DefenseCloudWatchEventRole defined above provides permissions to transfer CloudWatch Events.

MCDControllerRole

Cross-account IAM role that allows Multicloud Defense to access your cloud account and take necessary actions, for example, Create EC2 instances, create load balancers, and change Route53 entries. The service principal is the Multicloud Defense-controller-account with an external id applied. Here is the IAM policy applied to the role (e.g controller role name used in this example is ***Multicloud Defense-controller-role***):

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aacm:ListCertificates",
        "apigateway:Get",
        "ec2:*",
        "elasticloadbalancing:*",
        "events:DeleteRule",
        "events:ListTargetsByRule",
        "events:PutRule",
        "events:PutTargets",
        "events:RemoveTargets",
        "globalaccelerator:*",
        "iam:ListPolicies",
        "iam:ListRoles",
        "iam:ListRoleTags",
        "logs:*",
        "route53resolver:*",
        "servicequotas:GetServiceQuota",
        "s3:ListAllMyBuckets",
        "s3:ListBucket"
        "wafv2:Get*"
        "wafv2:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "iam:GetRole",
        "iam:ListRolePolicies",
        "iam:GetRolePolicy"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:iam::<ciscomcd-account>:role/ciscomcd-controller-role"
      ]
    },
    {
      "Effect": "Allow",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::<S3Bucket>/*"
    },
    {
      "Action": [
        "iam:GetRole",
        "iam:ListRolePolicies",
        "iam:GetRolePolicy",
        "iam:PassRole"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:iam::<customer- account>:role/ciscomcd_firewall_role"
    },
    {
      "Action": "iam:CreateServiceLinkedRole",
      "Effect": "Allow",
      "Resource": "arn:aws:iam::*:role/aws-service-role/*"
    }
  ]
}
```

Service Principal:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::<ciscomcd-account>:root"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "ciscomcd-external-id"
        }
      }
    }
  ]
}
```

MCDGatewayRole

Role that is assigned to the Multicloud Defense Gateway (Firewall) EC2 instances. The role gives the Gateway instance capabilities to access secretsmanager where the private keys for the application are stored, ability to decrypt keys using AWS KMS if the keys are stored in KMS, and save objects like PCAPs and technical support data onto a S3 bucket. The service principal of this role is ec2.amazonaws.com. Here is the IAM policy applied to the role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::*/*"
    },
    {
      "Action": [
        "kms:Decrypt"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Tip You can download and edit the CloudFormation template to make the policy more restrictive e.g. restricting decrypt to use a specific key, or PutObject to a defined/specific S3 bucket.

MCDInventoryRole

This is the role used for dynamic inventory purposes and provides the capability for the CloudTrail events to be transferred to the Controller's AWS account. It does the following:

- Put events on the event bus in the AWS account where the Multicloud Defense Controller exists.
- Send events matching the rule to the Multicloud Defense Controller's webhook server directly from the customer's AWS account.

The Service Principal for this role is **events.amazonaws.com**. Here is the policy applied to the role:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": "events:PutEvents",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:events:*:<ciscomcd-account>:event-bus/default"
      ]
    }
  ]
}
```

InventoryMonitorRule

Rule that is added to the MCDInventoryRole to put all CloudTrail inventory changes to EC2 and API gateways to be copied to the event bus on the AWS account where the Multicloud Defense Controller runs. The rule is required to match on specific event patterns that occur in the customer's AWS account. Once a match occurs, the rule states that the matched event should be sent to the webhook server (API based destination) of the controller. This rule is executed using the Multicloud DefenseMCDInventoryRole created in the previous section.

Custom Event Pattern:

```
{
  "detail-type": [
    "AWS API Call via CloudTrail",
    "EC2 Instance State-change Notification"
  ],
  "source": [
    "aws.ec2",
    "aws.elasticloadbalancing",
    "aws.apigateway"
  ]
}
```

Target:

```
Event Bus in another AWS Account (mcd-account) using the MCDInventoryRole
```

Azure IAM Roles

This document explains the details of the IAM roles created by the CloudFormation template used in the previous section.

The CloudFormation template creates the following role:

- **Custom Role** - The custom role gives the application permissions to read inventory information and create resources (e.g., VMs, load balancers, etc.) The custom role can be created in multiple ways.

GCP IAM Roles

This document explains the details of the service accountes created by the CloudFormation template used in the previous section.

The CloudFormation template creates the following accounts:

- **ciscomcd-controller service account** - This account is used by the Multicloud Defense Controller to access your GCP project to create resources (Multicloud Defense Gateway), load balancers for gateways, and read information about the VPCs, subnets, security grroup tags, and more.
- **ciscomcd-firewall service account** - This account is assigned to the Multicloud Defense Gateway (compute VM instances). The account provides access to the secret manager (private keys for TLS decryption) and storage. Also, the gateways many need permissions to send logs from Multicloud Defense Gatewayto the GCP logging instance (if configured by the user).