AWS Organizations

Overview

This document is intended to provide an Axonius user with the information that you need to configure the AWS adapter to query your AWS organization for all account IDs. Axonius then uses those account IDs to assume roles in each of your organization member accounts. This guide is not intended to be an exhaustive guide to AWS IAM or AWS Organizations, but a basic set of instructions to configure the AWS adapter. Please consult the AWS documentation for specifics on the IAM and Organizations services.

This guide will assume that you have 2 accounts as follows:

- **Organization Master Account**: This account is commonly referred to as the *root*, *management*, *master* or *master-payer* account for an AWS organization. Typically, this account will not contain resources outside of the IAM service. Its account number will be 111111111111 in the examples in this guide and will be referred to as the root account.
- **Organization Member Account #1**: This account is a subordinate or child member of the organization and typically contains the AWS resources (IAM, EC2, S3, etc.) that you would like to discover using Axonius. The account number will be 222222222222 in the examples in this guide. For the purposes of this document, this is also the account in which the Axonius system is deployed on an EC2 instance although this is not required in practice.

Axonius Adapter Configuration

The Axonius AWS adapter should be configured to use an AWS IAM user (with an Access Key ID and Secret Key) or an EC2-attached instance role. Please consult the Axonius documentation for a complete explanation of each method. In addition to these configuration items, you need to create an advanced configuration file in valid JSON format.

The advanced configuration file is a JSON-formatted file that contains information that Axonius requires to correctly query the organization and assume roles in the member accounts. The advanced configuration file should have the following contents:

```
{
  "fetch_roles_from_organization":
    {
      "organization_role_for_discovery": "arn:aws:iam::111111111111:role/Axonius-Adapter",
      "role_name": "Axonius-Adapter",
      "role_path": "",
      "external_id": "",
      "region": "us-east-1"
    },
  "skip_ec2_verification": true
}
```

The advanced configuration file should be loaded into the **Advanced Configuration file** section of the AWS adapter configuration dialog. An explanation of the advanced configuration file is in the Axonius documentation and a further description is included here.

Advanced Configuration Fields

This section describes each of the fields in the advanced configuration shown above.

Role for Organization Discovery

The configuration assumes the initial role defined at `organization_role_for_discovery`. This role is used to query the root organization account for a list of all organization member account numbers. The `111111111111` account number should be replaced with the account number of your organization root account. This role should exist in the organization root account.

Common Role Name

The `role_name` is the name of a role that must be present in all member accounts, and it is the role that will be used for the normal device and user discovery by Axonius. This role should have all the normal permissions for the AWS adapter and the services and resources that you would like to discover using Axonius. A sample policy is included in this guide in the section titled *Axonius Adapter Policy*.

NOTE Please note that the role referenced here, *Axonius-Adapter* must be created in advance and it must have the appropriate IAM policy attached for this feature to work. If the roles do not exist in all your organization accounts and if that role is not configured with an IAM policy that allows for asset discovery, the adapter will not function correctly and will present errors. Please refer to the Axonius Documentation for more information.

Role Path

If your IAM strategy uses special paths for IAM roles, that path should be entered here. In most AWS deployments, this field will be left empty.

External ID

If you use an external ID as an additional authentication factor during role inheritance, enter that external ID here between the double quotes. **This external ID must be the same in all member accounts**.

Region

We use this field to make our initial connection to the AWS APIs. This field can be left empty and, if so, we will assume `us-east-1` for the initial connection. In most cases, you can leave this field as it is presented above.

Skip EC2 Verification

If you have no EC2 instances in the root organization account, you must set `skip_ec2_verification` to `true`. This simply tells the AWS adapter to not look for EC2 instances in the account to verify a successful connection to AWS.

Axonius-Adapter Attached Role Policy

All the organization member accounts should have the same *Axonius Adapter Policy* as detailed below. This policy allows the *Axonius-Adapter* role in the organization member account to perform a device and/or user discovery. The policy that is included here should be edited to remove any unneeded permissions for services that are not important to you or are not configured for discovery in the adapter settings.

Next Steps

To continue the configuration, you must now decide to configure Axonius to authenticate to AWS with an IAM user that has an Access Key and Secret Key pair or using an EC2 Instance Attached Role. Both methods are discussed in the following sections. Please skip directly to the method that you prefer.

User with Keys

User Setup Overview

If you choose to configure the AWS adapter with an IAM user that has an access key and secret key, these instructions will help you to correctly configure that user and the roles in the organization member accounts so that the organizations feature will work with relative ease. This document is not intended to serve as a tutorial for AWS IAM or AWS Organizations and some level of familiarity with these services is required to execute these instructions.

Organization Root Account Setup

This section of the document will help you to set up the organization root account IAM user and IAM role, along with the requisite Trust Relationships and IAM policies.

User Setup

In the organization root account, you should configure an IAM user that has security credentials for programmatic access to AWS. We will refer to this user as *axonius_organization_user* for the purposes of this guide. This user should have a single IAM policy attached to it that allows it to assume the *Axonius-Adapter* role. The *Axonius-Adapter* role is referenced in the *role_name* section of the *Axonius Adapter Configuration* section.

Assume Role Policy

The policy presented here is only a reference. You should configure it according to your specific security requirements as needed. This policy does nothing more than allow the *axonius_organization_user* to assume the *Axonius-Adapter* role in all AWS accounts.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AxoniusAssumeOrgDiscoveryRole",
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::*:role/Axonius-Adapter"
    }
  ]
}
```

Role Setup

In the organization root account, you should configure an IAM role for this user to assume.

Axonius-Adapter Role and Permissions

This role is the role referenced in the *Advanced Configuration File* and it needs a Trust Relationship defined to allow the *axonius_organization_user* to assume the role. It also requires 2 permission grants as detailed in the following example IAM policies.

Important This role, Trust Relationship and Adapter Discovery Policy must be created in each of the organization member accounts. The List Accounts Policy is only required in the root organization account.

Axonius-Adapter - Trust Relationship

The trust relationship for the Axonius-Adapter role allows for the user that you created above to assume this role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:user/axonius_organization_user"
      },
      "Action": "sts:AssumeRole",
      "Condition": {}
    }
  ]
}
```

Axonius-Adapter - List Accounts Policy

This IAM policy allows the *Axonius-Adapter* role to list all account IDs in the organization.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AxoniusListAccounts",
            "Effect": "Allow",
            "Action": "organizations:ListAccounts",
            "Resource": "*"
        }
    ]
}
```

Axonius-Adapter - Adapter Discovery Policy for Users

The adapter discovery policy allows the Axonius system to perform asset discovery. You can refer to the *Axonius Adapter Policy* section of this guide for a sample policy.

EC2-Instance Attached Role

Overview for EC2 Role

If you choose to configure the AWS adapter with an EC2-instance attached role, these instructions will help you to correctly configure the instance, instance role and member account roles such that the organizations feature will work with relative ease. This document is not intended to serve as a tutorial for AWS EC2, AWS IAM or AWS Organizations and some level of familiarity with these services is required to execute these instructions.

This document assumes that you have already deployed the EC2 instance in AWS according to the instructions in the Axonius Documentation.

EC2 Instance Attached Role

Using an EC2 instance attached role offers flexibility and reduces the reliance on a dedicated AWS IAM user account and the need to manage a set of Access and Secret keys. The following sections of this guide will explain how to configure a role, called *Axonius-Adapter* to use as an EC2 instance attached role.

This role should be created in the account that houses the Axonius system, deployed on EC2.

Axonius-Adapter Role and Permissions for EC2 Role

This role is the role referenced in the *Advanced Configuration File* and it needs a Trust Relationship defined to allow the AWS EC2 service to assume the role. It also requires a single permission grant as detailed in the following example IAM policies.

Important This role, Trust Relationship and Adapter Discovery Policy must be created in each of the organization member accounts. The List Accounts Policy is only required in the root organization account.

Axonius-Adapter - Trust Relationship for EC2 Role

The trust relationship for the Axonius-Adapter role allows for the EC2 service to assume this role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EC2AssumeRole",
      "Effect": "Allow",
      "Principal": {
        "Service": "ec2.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Axonius-Adapter - Assume Role Policy for EC2 Role

This IAM policy allows the *Axonius-Adapter* role to assume a role in each of the organization member accounts. Conveniently, these roles in the organization member accounts are all named the same, *Axonius-Adapter*. This is the only permission required for the EC2 instance attached role.

```
{
    "Statement": [
        {
            "Action": [
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:iam::*:role/Axonius-Adapter",
            "Sid": "AxoniusAdapterOrganizationAssumeRole"
        }
    ],
    "Version": "2012-10-17"
}
```

Organization Account Configuration for EC2 Role

Each account in the organization must have a role named *Axonius-Adapter*. This role must also have a Trust Relationship and an Axonius adapter discovery policy. Additionally, a separate List Accounts policy must be placed in the organization root account. Examples of the Trust Relationship and these policies is included in the following sections of this guide.

Axonius-Adapter - Trust Relationship for EC2 Role in Member Accounts

The trust relationship for the Axonius-Adapter role allows for the role that you created above to assume this role in an organization member account. Replace the placeholder account number below, '222222222222', with the account number that contains the Axonius EC2 instance.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::222222222222:role/Axonius-Adapter"
      },
      "Action": "sts:AssumeRole",
    }
  ]
}
```

Axonius-Adapter - List Accounts Policy for EC2 Role

This IAM policy allows the *Axonius-Adapter* role to list all account IDs in the organization. This policy is required only in the organization root account.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AxoniusListAccounts",
            "Effect": "Allow",
            "Action": "organizations:ListAccounts",
            "Resource": "*"
        }
    ]
}
```

Axonius-Adapter - Adapter Discovery Policy for EC2 Role

The adapter discovery policy allows the Axonius system to perform asset discovery. You can refer to the *Axonius Adapter Policy* section of this guide for a sample policy.

Axonius Adapter Policy

This policy allows the *Axonius-Adapter* role to perform a device and/or user discovery. The policy that is included here should be edited to remove any unneeded permissions for services that are not important to you or are not configured for discovery in the adapter settings.

```
{
    "Statement": [
        {
            "Action": [
                "apigateway:GET",
                "acm:DescribeCertificate",
                "acm:ListCertificates",
                "cloudfront:GetDistribution",
                "cloudfront:ListDistributions",
                "cloudtrail:DescribeTrails",
                "cloudtrail:GetEventSelectors",
                "cloudtrail:GetTrailStatus",
                "cloudwatch:DescribeAlarmsForMetric",
                "config:DescribeConfigurationRecorders",
                "config:DescribeConfigurationRecorderStatus",
                "dynamodb:DescribeGlobalTable",
                "dynamodb:DescribeGlobalTableSettings",
                "dynamodb:DescribeTable",
                "dynamodb:ListGlobalTables",
                "dynamodb:ListTables",
                "ec2:DescribeAddresses",
                "ec2:DescribeFlowLogs",
                "ec2:DescribeImages",
                "ec2:DescribeInstances",
                "ec2:DescribeInternetGateways",
                "ec2:DescribeNatGateways",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeTags",
                "ec2:DescribeVolumes",
                "ec2:DescribeVpcPeeringConnections",
                "ec2:DescribeVpcs",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:DescribeServices",
                "ecs:DescribeTasks",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
                "ecs:ListServices",
                "ecs:ListTagsForResource",
```

```
                "ecs:ListTasks",
                "eks:DescribeCluster",
                "eks:ListClusters",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeLoadBalancerPolicies",
                "elasticloadbalancing:DescribeSSLPolicies",
                "elasticloadbalancing:DescribeTargetGroups",
                "elasticloadbalancing:DescribeTargetHealth",
                "es:DescribeElasticsearchDomain*",
                "es:ListDomainNames",
                "iam:GenerateCredentialReport",
                "iam:GenerateServiceLastAccessedDetails",
                "iam:GetAccountSummary",
                "iam:GetAccountPasswordPolicy",
                "iam:GetAccessKeyLastUsed",
                "iam:GetCredentialReport",
                "iam:GetLoginProfile",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:GetRolePolicy",
                "iam:GetUser",
                "iam:GetUserPolicy",
                "iam:ListEntitiesForPolicy",
                "iam:ListPolicies",
                "iam:ListAccessKeys",
                "iam:ListAccountAliases",
                "iam:ListAttachedGroupPolicies",
                "iam:ListAttachedRolePolicies",
                "iam:ListAttachedUserPolicies",
                "iam:ListGroupsForUser",
                "iam:ListInstanceProfilesForRole",
                "iam:ListMFADevices",
                "iam:ListRolePolicies",
                "iam:ListRoles",
                "iam:ListUserPolicies",
                "iam:ListUsers",
                "iam:ListVirtualMFADevices",
                "kms:GenerateDataKey",
                "kms:Decrypt",
                "kms:ListKeys",
                "lambda:GetPolicy",
                "lambda:ListFunctions",
                "logs:DescribeMetricFilters",
                "organizations:DescribeAccount",
                "organizations:DescribeOrganization",
                "organizations:ListAccounts",
                "organizations:DescribeEffectivePolicy",
                "organizations:DescribePolicy",
                "organizations:ListPoliciesForTarget",
                "organizations:ListTagsForResource",
                "rds:DescribeDBInstances",
                "route53:ListHostedZones",
                "route53:ListResourceRecordSets",
                "s3:GetAccountPublicAccessBlock",
                "s3:GetBucketAcl",
                "s3:GetBucketLocation",
                "s3:GetBucketLogging",
                "s3:GetBucketPolicy",
                "s3:GetBucketPolicyStatus",
                "s3:GetBucketTagging",
                "s3:GetBucketPublicAccessBlock",
                "s3:GetObject",
                "s3:ListBucket",
                "s3:ListAllMyBuckets",
```

```
                "s3:PutObject",
                "s3:PutObjectTagging",
                "servicediscovery:ListNamespaces",
                "sns:ListSubscriptionsByTopic",
                "ssm:DescribeAvailablePatches",
                "ssm:DescribeInstanceInformation",
                "ssm:DescribeInstancePatches",
                "ssm:DescribePatchGroups",
                "ssm:GetInventorySchema",
                "ssm:ListInventoryEntries",
                "ssm:ListResourceComplianceSummaries",
                "ssm:ListTagsForResource",
                "sts:GetCallerIdentity",
                "sts:AssumeRole",
                "waf:GetWebACL",
                "waf:ListWebACLs",
                "waf-regional:GetWebACL",
                "waf-regional:GetWebACLForResource",
                "waf-regional:ListWebACLs",
                "wafv2:GetWebACL",
                "wafv2:GetWebACLForResource",
                "wafv2:ListWebACLs",
                "workspaces:DescribeTags",
                "workspaces:DescribeWorkspaceDirectories",
                "workspaces:DescribeWorkspaces",
                "workspaces:DescribeWorkspacesConnectionStatus"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Sid": "AxoniusFullLeastAccess"
        }
    ],
    "Version": "2012-10-17"
}
```