Discovery Using Cloud APIs (version 1.1)

**Cloud Service Label:**

Description

An adversary may attempt to enumerate the cloud services running on a system after gaining access via credentials or by using publicly accessible API’s to brute force access to cloud resources. Many different services exist throughout the various cloud providers and can include continuous integration and continuous delivery (CI/CD), Lambda Functions, Azure AD, storage etc. Adversaries may attempt to discover information about the services enabled throughout the environment.

Examples

|  |  |
| --- | --- |
| **Name** | **Description** |
| Pacu | Pacu, an open source AWS exploitation framework, supports several methods for discovering cloud services. s3\_bucket\_finder for instance will attempt to guess S3 bucket names and scan for them in every AWS region. If configuration settings are misconfigured, an adversary can even access your S3 bucket with no additional knowledge or credentials. PACU will soon have Azure functionality according to the authors. |
| WeirdAAL | This is an open source resource to interact with AWS services. It is a compilation of discovery scripts that can be utilized by offensive and defensive security experts. |
| NimboStratus | This is an open source tool for fingerprinting and exploiting Amazon cloud infrastructures. These tools are a PoC developed using the [boto](https://github.com/boto/boto) library for accessing Amazon's API. |
| ROADtools | This is an open source framework to interact with Azure AD. There are two libraries this consists of, ROADlib and ROADrecon. ROADlib is a library that can be used to authenticate with Azure AD or to build tools that integrate with a database containing ROADrecon data. ROADrecon Uses an automatically generated metadata model to create an SQLAlchemy backed database on disk. It also uses asynchronous HTTP calls in Python to dump all available information in the Azure AD graph to this database. It gives the ability to provide plugins to query the database and output it to a useful format. |

Mitigations

|  |  |  |
| --- | --- | --- |
| **Mitigation** | | **Description** |
|  | | This type of attack technique cannot be easily mitigated with preventive controls since it is based on the abuse of system features. |
| Least Privilege | | All access given to users in the cloud environment should be assigned by the necessary privileges needed for team members to complete their job responsibilities. Ensure that temporary access tokens are issued rather than permanent credentials, especially when access is being granted to entities outside of the internal security boundary . |
|  | *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.** |

Detection

|  |  |
| --- | --- |
| **Detection** | **Description** |
| Create Log Metric Filters and Alarms for AWS | To create a metric filter and alarm:   1. Create a metric filter that checks for IAM policy changes and the *<cloudtrail\_log\_group\_name>* 2. Create an SNS topic 3. Create an SNS subscription to the above topic 4. Create an alarm associated with the filter and SNS topic created in steps 1 and 2 respectively |
| Monitor Activity in AWS Account | Various services in AWS offer logging features that allow for detection capabilities. These include CloudFront, CloudTrail, CloudWatch, Config, and S3. |
| Monitor for Suspicious Activity in Azure | Azure AD can generate anomaly reports than can be run on a daily basis. Azure AD Identity Protection show current risks in its dashboard and provides daily email summary notifications. Policies can also be configured to alert to specific issues. |
| Create Log Metric Filters and Alarms for CloudTrail | To create a metric filter and alarm:   1. Create a filter that checks for CloudTrail changes and the specific *<cloudtrail\_log\_group\_name>* 2. Create an SNS topic that the alarm will notify 3. Create an SNS subscription to the above topic 4. Create an alarm associated with the filter from step 1 and SNS topic in step 2 |
| Create Activity Log Alerts in Azure | To create log activity alerts for deletion in the Azure Console:   1. Navigate to *Monitor’ / ‘Alerts* 2. Select *Manage alert rules* 3. Click on the Alert *Name* where Condition contains *operationName equals Microsoft.Network/networkSecurityGroups/securityRules/delete* 4. Hover a mouse over *Condition* to ensure it is set to *Whenever the Administrative Activity Log “Delete Security Rule (networkSecurityGroups/securityRules)” has “any” level with “any” status and event is initiated by “any*” |
| Create, View, and Manage Activity Alerts in Azure Monitor | To create a log alert in the Azure portal:   1. Select **Monitor -> Alerts** 2. Select **New alert rule** of the **Alerts** window 3. Provide information in **Define alert condition** 4. Provide details in **Define alert details** 5. Specify action group for new alert rule under **Action group**, or create a new action group with + **New group** 6. Select **Yes** for the **Enable rule upon creation** option 7. Select **Create alert rule**   To view and manage alerts:   1. Select **Monitor -> Alerts -> Manage alert rules** 2. Select the rule you want to modify and double-click to edit the rule options 3. Click **Save** |
| Azure Resource Manager Templates | Azure Resource Manager templates in the format of JSON files that can be used to configure metric alerts in Azure Monitor. These templates can be used for simple static and dynamic threshold metric alerts, availability tests, and monitoring multiple resources. |
| Enable S3 Bucket Logging | To enable CloudTrail S3 bucket logging:   1. Navigate to CloudTrail console at Go to the Amazon CloudTrail console 2. Click Trails in the API activity history pane on the left 3. Sign into AWS Management Console and open the S3 console Sign in to the AWS Management Console and open the S3 console 4. Click on a bucket under *All Buckets* 5. Click on *Properties* 6. Under *Bucket:\_<bucket\_name>\_* click *Logging* 7. Ensure *Enabled* is checked |
| Enable Azure Storage Logging | This is used to track how requests made to Azure Storage were authorized. Enabling logs provides visibility into whether a request was anonymous, made with an OAuth2.0 token, a shared key, or shared access signature (SAS). Full Azure Storage analytics logging details can be found at [**https://docs.microsoft.com/en-us/azure/storage/common/storage-analytics-logging?tabs=dotnet**](https://docs.microsoft.com/en-us/azure/storage/common/storage-analytics-logging?tabs=dotnet)**.** |
| Enable CloudTrail across all regions in AWS | To enable CloudTrail across all regions:   1. Sign into the AWS Management Console and open the CloudTrail console 2. Click on *Trails* 3. Set necessary Trails to All option in the I column 4. Click on a trail via the link *Name* column 5. Set *Logging* to *ON* 6. Set *Apply trail to all regions* to *Yes* |
| Configure log profile to capture activity logs for all regions in Azure | To set up activity logs for all regions:   1. Navigate to Azure console 2. Go to *Activity log* 3. Select *Export* 4. Select *Subscription* 5. Check *Select all* in *Regions* 6. Select *Save* |
| Monitoring for Regional Activity | Tools like Splunk or even CloudSploit have the ability to alert based on region and ingest CloudTrail logs. In CloudSploit, a plugin called EC2 Max Count can be configured to alert after a certain threshold of EC2 instances is met. Real-Time Events service is another feature of CloudSploit that allows alerts for activity in unused regions. |

References

1. [Rhino Security Labs. (2019, August 22). Pacu. Retrieved October 17, 2019.](https://github.com/RhinoSecurityLabs/pacu)
2. <https://github.com/carnal0wnage/weirdAAL/wiki/_modules_sts>. Accessed Feb 22,2020
3. <http://andresriancho.github.io/nimbostratus/>. Accessed Feb 22,2020