Cloud Service Enumeration (version 1.1)

**Cloud Service Label: IaaS, PaaS**

Description

An adversary may attempt to enumerate the cloud security services running on a system after gaining initial access. These services can differ depending on if it's platform-as-a-service (PaaS), or infrastructure-as-a-service (IaaS). Most security aspects of Cloud services are easily viewed and configured from the cloud service dashboard/portal.  Even without the ability to change security settings, the ability to read the security configurations of network firewalls, monitoring agents, etc., enable a significantly greater opportunity to defeat these defenses.

Examples

|  |  |
| --- | --- |
| **Name** | **Description** |
| AWS Service Enumeration | Matching a collected AWS credential with its associated services. **aws\_service\_enum.py** on the NotSoSecure GitHub page helps identify various AWS services based on the found access or secret key like S3 buckets, service region, Route53 domains, IP addresses, etc. |
| Azure Service Enumeration | Matching a collected Managed Identity access token with its associated services. Providing an access key to **azure\_service\_enum.py** on the NotSoSecure GitHub page will enumerate information related to VM ID’s, hardware and storage profiles, operation system types, etc. |
| GCP Service Enumeration | Matching a collected OAuth Access Token with its associated services. The list of services can be accessed through the REST API interface, but this process is automated through **gcp\_service\_enum.py** on the NotSoSecure GitHub page, where information related to e-mails, applications, storage buckets, can be found. |
| Aws\_pwn | The aws\_pwn GitHub repo has a script named *dump\_account\_data.sh,* which calls account-based read, list, get, and describe functions and saves the data to a given location. |
| CloudSploit | Once an attacker has gained initial access to a cloud environment, whether it is through credentials or exploitation of a vulnerable service, an open-source tool like CloudSploit can be used to detect security risks and potential misconfigurations in various cloud platforms like AWS, Azure, GCP or Oracle Cloud Infrastructure. |
| Barq | Open-source GitHub post-exploitation framework for AWS, created by Voulnet. Features include performing further attacks after intitial access like launching Metasploit and Empire payloads, as well as enumerating information related to EC2 instances, secrets, and metadata. |
| Pacu | Post-exploitation tool created by Rhino Security Labs that enables plug-in functionality for enumeration, privilege escalation, data exfiltration, service exploitation, and log manipulation within AWS. |

Mitigations

|  |  |  |
| --- | --- | --- |
| **Mitigation** | | **Description** |
| Multi-factor Authentication | | Use multi-factor authentication for user and privileged accounts. Do not manage Cloud portals from machines that perform user email and web browsing tasks. All users should be required to utilize two factor authentication. |
|  | *AWS* | This can be enforced by first creating a policy that would prohibit actions except those that allow a user to change their password or manage 2FA, then attaching a policy to a group that includes all user accounts where they can be allowed all access if they sign in with 2FA. Once these actions are completed it should be tested to verify the access is given correctly. To see full details on how to complete this view AWS documentation at: **https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial\_users-self-manage-mfa-and-creds.html.** |
|  | *Azure* | This can be done by creating a MFA registration policy. It can than be assigned to all users (with the ability to exclude some if need be, but is not recommended). Make sure once the policy is created and added to users that it is then being enforced, once enforced it should be tested for verification. To see full details on how to complete this view Azure documentation at: **https://docs.microsoft.com/en-us/azure/active-directory/identity-protection/howto-identity-protection-configure-mfa-policy.** |
|  | *GCP* | This can be done by first enabling it on the current account being used by admin to assign the roles, then enable two factor on an instance by instance or project by project basis, then assigning the requirements based on IAM roles and applying it to all users. To see full details on how to complete this view Azure documentation at: **https://cloud.google.com/compute/docs/oslogin/setup-two-factor-authentication.** |
| Privileged Account Management | | Do not allow subscription-level administrator accounts to be used for day-to-day operations that may expose them to potential adversaries on unprivileged systems. This can be done by first limiting the access that these accounts have outside of the administrative rights, but then also monitoring (details in detection) for events that might show a compromised account. |
|  | AWS | To manage the access that privileged accounts have on the AWS cloud system to only allow administrators to perform administrative tasks on such accounts can be accomplished utilizing limited IAM administrator accounts. To configure this the administrator would have two accounts; one would have administrative rights and no basic access while the other account has basic access with no administrative rights. To limit the administrative account the IAM limited administrator would be used. This is done by creating a policy that gives a user admin rights, but disallows the other actions using the AWS command line interface. This is outlined at: [**https://aws.amazon.com/blogs/security/how-to-create-a-limited-iam-administrator-by-using-managed-policies/**](https://aws.amazon.com/blogs/security/how-to-create-a-limited-iam-administrator-by-using-managed-policies/)**.** |
|  | Azure | To manage the access that privilege accounts have on the Azure cloud system to only allow administrators to perform administrative tasks on such accounts can be accomplished utilizing limited IAM administrator accounts. To configure this the administrator would have two accounts; one would have administrative rights and no basic access while the other account has basic access with no administrative rights. To limit the administrative account the specific administrative needs can be picked from a number of options available (Azure DevOps Administrator, Billing Administrator, Cloud Application Administrator, etc.) These different options can be edited to fit the needs and limit the basic access. This is outlined at: [**https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-assign-admin-roles**](https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-assign-admin-roles)**.** |
|  | GCP | To manage the access that privilege accounts have on the Azure cloud system to only allow administrators to perform administrative tasks on such accounts can be accomplished utilizing limited IAM administrator accounts. To configure this the administrator would have two accounts; one would have administrative rights and no basic access while the other account has basic access with no administrative rights. To limit the administrative account pre-defined administrator accounts can be used (mobile admin, Google voice admin, help desk admin, etc.). These accounts can be used with their pre-defined settings, or modified depending on specific use cases. These can limit access to basic functionality needed. This is outlined at: **https://support.google.com/a/answer/2405986?hl=en.** |
| 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.** |
| Temporary Access Tokens | | Rotate access keys often to shorten abuse potential |
| Block Network Discovery Tools | | Prevent tools like nmap, traceroute, ping from non-whitelisted accounts |

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 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* |

References

1. <https://cseweb.ucsd.edu/~savage/papers/CCS09.pdf>. Accessed July 2, 2020.
2. <https://github.com/NotSoSecure/cloud-service-enum/>. Accessed July 6, 2020.
3. <https://github.com/dagrz/aws_pwn>. Accessed July 16. 2020.
4. https://github.com/cloudsploit/scans. Accessed July 22, 2020.
5. https://github.com/Voulnet/barq. Accessed July 23, 2020.
6. https://github.com/RhinoSecurityLabs/pacu. Accessed July 24, 2020.