Collecting Data from Cloud Logs (version 1.0)

**Cloud Service Label: IaaS, PaaS**

Description

Adversaries may be able to use the cloud’s extensive logging capabilities, and the information written to them, as a way to gain information/access to a system. Ultimately, this can lead to the adversary gaining access to user data of a sensitive nature. Logging is essential for many aspects of day to day operations but logging of all information is not always necessary once a system is in production. Some types of log files that can be utilized are server log files, debugging logs, plugin logs, etc.

The Cloud depends on copious logging to make things more transparent to customers who no longer have access to underlying hardware and software layers. The cloud also enables these logs to be distributed to several different locations such as storage accounts, event queues, SIEMS, on premise repositories, etc. Cloud logs can be very informative, telling an adversary what assets are available and how they are protected. Sensitive data may also be exposed in these logs. In Azure for example, Key Vault secrets that would normally be very difficult to obtain may be logged in Logic function logs in the clear by default. The availability and integrity of logs is often stressed by security professionals. The confidentiality of these logs is often not given priority.

Examples

|  |  |
| --- | --- |
| **Name** | **Description** |
| CVE-2020-7599 | Adversaries may exploit a vulnerability in com.gradle.plugin-publish versions prior to 0.11.0 where an author publishes a plugin where the build log is public and the –info flag is used. Under these circumstances the build log will contain an AWS pre-signed URL that would allow an adversary to access the pre-signed URL, of a plugin that was recently uploaded, and replace it with their own plugin. |
| CVE-2019-4284 | IBM Cloud Private versions 2.1.0, 3.1.0, 3.1.1, and 3.1.2 contain a vulnerability that allows a local privileged user to view OpenID Connect tokens that could be printed to log files. These tokens could be utilized to log into the system as other users. |
| CVE-2019-1003062 | Jenkins AWS CloudWatch Logs Publisher Plugin stores unencrypted credentials in its global configuration file, which can be viewed by users who have access to the master file system. |
| CVE-2019-4143 | The Private Key Management Service in IBM Cloud Private versions 3.1.1 and 3.1.2 allows for a user to potentially obtain sensitive information from the KMS plugin log. |

Mitigations

|  |  |
| --- | --- |
| **Mitigation** | **Description** |
| Log Configurations | Make the decision on what is necessary to be logged and what is not. |
| Log File Permissions | Make sure that log files have permissions in place to avoid unauthorized read/write access. |
| AWS CIS Benchmark Standards | Refer to the logging section of the AWS CIS Benchmark Standards. These include ensuring CloudTrail is enabled in all regions, enabling log file validation, removing public access to S3 buckets, encrypting CloudTrail logs at rest using KMS CMKs, and enabling VPC flow logging in all VPCs, and integrating CloudTrail trails with CloudWatch Logs. |
| AWS CloudWatch Logs Documentation | IAM Identities (identity-based policies) can be attached to specific writing permissions policies. CloudWatch Log policies use AWS condition keys to express conditions. |
| Azure CIS Benchmark Standards | Refer to the logging section of the Azure CIS Benchmark Standards. These include ensuring that activity logs for storage containers are not publicly accessible and that they are encrypted with BYOK (Bring Your Own Key) |

Detection

|  |  |
| --- | --- |
| **Detection** | **Description** |
| 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, 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. |

References

1. [CWE. (Feb 20, 2020). CWE-532: Insertion of Sensitive Information into Log File. Retrieved June 9, 2020.](https://cwe.mitre.org/data/definitions/532.html)
2. [CVE. (Jan 21, 2020). CVE-2020-7599. Retrieved June 9, 2020.](https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-7599)
3. <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-4284>. Accessed August 6, 2020.
4. <https://www.youtube.com/watch?v=PgujSug1ZbI>. Accessed June 22, 2020.
5. <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-1003062>. Accessed July 17. 2020.
6. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-4143. Accessed August 6, 2020.
7. <https://docs.microsoft.com/en-us/azure/azure-monitor/platform/alerts-activity-log>. Accessed August 6, 2020.
8. <https://docs.microsoft.com/en-us/azure/azure-monitor/platform/alerts-metric-create-templates#template-for-a-metric-alert-that-monitors-multiple-resources>. Accessed August 6, 2020.