ManageEngine
Log360

5 errores comunes en la configuración de AWS que llevan a ciberataques

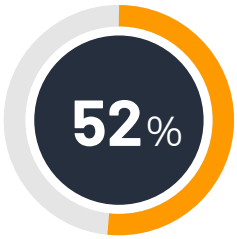
www.manageengine.com/latam/log360

Tabla de contenidos

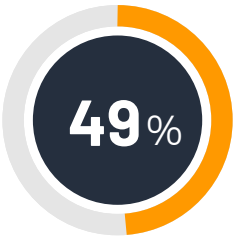
Introducción

La tecnología de nube ha estado en el mercado desde hace casi una década. Ahora, dado el aumento en la adopción del trabajo a distancia, 59 por ciento del uso de la nube por parte de las empresas supera sus planes anteriores. Aunque es conveniente para las operaciones empresariales, este auge en la adopción de la nube y el trabajo a distancia también es un imán para que los ciberdelincuentes lancen ataques, y muchas empresas están mal preparadas para enfrentar las amenazas de seguridad en la nube. Los atacantes lo ven todo; vigilan a las empresas constantemente y buscan vulnerabilidades y oportunidades para hackear.

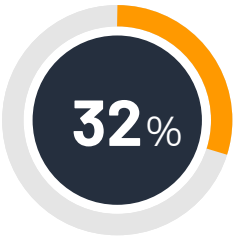
Cuando se trata de la nube, los hackers suelen aprovecharse de los errores y comportamientos humanos más que de los fallos técnicos de seguridad. Según una encuesta reciente, los **errores de configuración de la nube son la principal causa de las violaciones de datos en la nube**. Además, las estadísticas revelan que



de los encuestados dijeron que el **desconocimiento** de la seguridad en la nube y de las políticas causó una violación de datos.



de los encuestados afirmaron que la **falta de controles adecuados** y vigilancia son responsables de la pérdida de datos.



piensan que la negligencia en el **comportamiento de los intrusos** dio lugar a incidentes de seguridad.

Estas cifras revelan una clara necesidad de gestionar los errores de configuración de la nube de forma más eficaz y eficiente para evitar incidentes adversos. Indican la necesidad de implementar una herramienta que pueda proporcionar una mejor visibilidad de la infraestructura de la nube, detectar y remediar automáticamente los errores de configuración y enviar notificaciones oportunas en caso de cambios de configuración inapropiados y peligrosos.

En este e-book, hablaremos de los errores de configuración más comunes en Amazon Web Services (AWS) que debería evitar para frenar los incidentes y las infracciones de seguridad.

Configurar AWS Cloud Trail:
Requisitos previos para la auditoría

Para obtener una visibilidad completa en su entorno de AWS Cloud y para reforzar las configuraciones de seguridad, primero debe **habilitar la recopilación de logs de CloudTrail**. CloudTrail es un servicio de AWS que proporciona el historial de eventos para toda la actividad de la cuenta de AWS. Genera datos de log para todas las llamadas a la API realizadas dentro de AWS, incluida la consola de gestión de AWS, los kits de desarrollo de software (SDK), las herramientas de línea de comandos y otros servicios de AWS. Habilitar el registro de CloudTrail para todas las regiones ayuda a prevenir posibles brechas en el monitoreo, y también le permite cumplir con los mandatos normativos y llevar a cabo investigaciones forenses después de los incidentes.

Los datos de log generados se almacenan en un bucket de Amazon S3. Sin embargo, si los ciberatacantes lograran ingresar en su red en la nube, lo primero que harían sería desactivar su CloudTrail y tratar de comprometer los archivos de log. Por esta razón, es vital tomar las siguientes precauciones para que sus archivos de log de CloudTrail sean a prueba de ataques:

- **Garantizar la integridad de los archivos de log de CloudTrail:** Habilite la validación de la integridad de los archivos de log de CloudTrail para poder rastrear cualquier cambio realizado en los datos de los archivos de log después de que se almacenen en el bucket de S3.
- **Evitar los intentos de acceso no autorizados:** Habilite el registro de acceso para el bucket de CloudTrail S3 para que pueda rastrear todas las solicitudes de acceso y detectar los intentos de acceso no autorizados.
- **Duplicar la seguridad para acceder a los archivos de log:** Active la opción de autenticación multifactor (MFA) para eliminar los buckets de CloudTrail S3, de modo que incluso si una cuenta se ve comprometida, pueda garantizar la seguridad de los archivos de log.

Bonus tips:
Which compliance requirements mandate you to manage unused user keys?

- ISO 27001 - A.9.2.4 -
Management of secret authentication information of users.

- HIPAA 164.308(a)(5)(ii)(D) -
Procedures for creating, changing, and safeguarding passwords.

How to detect and fix this issue

Auditing AWS accounts is one of the best ways to detect unused access keys. Follow the steps below in your AWS management console to detect and remove the unused access keys.

- Log in to the AWS console. Select **Users** under the IAM service option from the left-hand side pane.
- Click the username you want to audit.
- In the consequent panel, click the **Security Credentials** tab.
- Investigate the **Last Used** column, and ensure that it doesn't read N/A. If it says N/A, it means that the access key has never been used by the selected user and is hence an unused key.

You'll have to repeat this procedure for every suspected user account, and check if the access key is unused or used. Alternatively, you can also find the unused access keys by downloading the credentials report and executing specific commands in the command-line interface (CLI). After getting the list of unused access keys, you can go ahead and delete it from the same console.

Mistake 4

Providing unrestricted access to Redshift cluster

A Redshift data warehouse is a collection of computing resources called nodes, which are organized into a group called a cluster. Each cluster runs an Amazon Redshift engine and contains one or more databases.

When you first provision an Amazon Redshift cluster, nobody has access to it by default. To grant other users access to this cluster, you need to associate it with a security group and define rules for allowing access.

Note:

If you are on the EC2-VPC platform, you can either associate an existing Virtual Private Cloud (VPC) security group or define a new one. If you're on the EC2-Classic platform, you need to define the security group and associate it with a cluster.

If you do not configure the security groups associated with the cluster properly, these Redshift clusters will be made publicly accessible. Anyone on the internet can then establish a connection to your database, increasing the security risk of brute-force attacks, SQL injections, or DoS attacks.

Bonus tip:

To secure the Redshift cluster traffic further, enable encryption for all the traffic on the wire. To do this, you must have the ***require_ssl*** parameter enabled.

How to fix it

Do not configure the security group open to a broad range of IP addresses. Provide inbound access to specific IP ranges, and also expose only the ports that are needed.

Mistake 5

Overly permissive access to cloud resources

We would never allow any direct connections to our network devices through the internet without a firewall. However, when it comes to the cloud, sometimes administrators unintentionally miss out on configuring inbound access rules; this provides direct access of sensitive resources through the internet. Below are some examples of overly permissive access provided to cloud resources and components that could lead to a cloud security disaster.

- **Kubernetes cluster being exposed to the internet:**

  Over the last couple of years, the usage of Kubernetes has increased rapidly, as many cloud companies have adopted it as the default way to orchestrate and scale their container-based workloads. A recent analysis shows that etcd services, an open source distributed key-value store used to hold and manage the critical information that distributed systems need to keep running, have shown up on the internet without proper authentication.

According to this research, a simple search on Shodan revealed nearly 2,284 etcd servers on the internet, highlighting that this is a problem to look out for while setting up your clusters. Ensure that the etcd port 2379 for Kubernetes cluster is not exposed to the internet.

How to fix it

Constantly audit and review the VPC security groups and network access control lists (NACLs) that guard the inbound and outbound traffic to resources. Track changes to these groups and get alerted when the above protocols and ports are configured for unrestricted access.

Bonus tips:

Apart from removing the unused user access keys, ensure that you take the below security steps as well:

- **It's not just the etcd:** Depending on your cluster's configuration, other services might also pose vulnerabilities that lead to a security incident. Take for instance the Kubelet API, used by Kubernetes to manage containers on individual nodes or in some clusters; if this API is available unauthenticated, it could lead to a Kubelet exploit. Hackers could leverage this opportunity to execute code in your containers, and even take over your entire cluster, so you must always have an authentication gateway for Kubelet API.

- **A compromised container is equal to a compromised cluster:** As you know, a Kubernetes cluster hosts a set of application containers, so it's important to ensure the security of each and every application running on the cluster to avoid a full-blown cluster compromise.
  If role-based access control (RBAC) isn't configured for these clusters, attackers who gain access to a single container in a cluster can easily escalate their privilege and gain full control of it. Therefore, it's always mandatory to configure RBAC for these clusters.
  If you really want to follow the default behavior wherein a token that provides access to the Kubernetes API is mounted onto each container, ensure that you've restricted the rights the user has on the other cluster resources.

- **Unrestricted access to well-known ports:** Ensure that legacy ports and protocols are enabled on the cloud host without any restrictions. This will lead to unrestricted and easy access of these hosts by the malicious actor over the internet. Ensure access to required entities for:

  - **Common Internet File System (CIFS)** to avoid unauthorized data access.

  - **File Transfer Protocol (FTP),** which accesses through port 20/21 to avoid unauthorized data access and accidental data breach.

  - **Internet Control Message Protocol (ICMP)** to avoid unauthorized data access, rogue scanning of network vulnerabilities, or DoS attacks against the cloud infrastructure.

- **Port 27017 (MongoDB), port 1433 (MSSQL), port 3306 (MySQL), port 1521 (Oracle DB), and port 5432 (PostgreSQL)** to avoid accidental data breach, and unauthorized data access and leaks.

- **Remote Desktop Protocol (RDP)** accessed through **port 3389** to avoid unauthorized resource access and security breaches.

- **Remote procedure call (RPC)** accessed through **port 135** and **SMTP through port 25** to avoid data leaks.

- **Secure Shell (SSH)** through **port 22** and **Telnet** through **port 23** to avoid data leaks.

About Log360

The need of the hour is a solution that monitors, audits, and helps you manage your AWS cloud infrastructure. Introducing Log360, a comprehensive security and information event management (SIEM) solution that provides you:

- A central console that collects, analyzes, and monitors the log data from CloudTrail and S3 bucket to give complete visibility into activities happening in your AWS environment.
- Exhaustive information on user activities in your AWS.
- Real-time notifications and detailed reports on the changes happening to VPC security groups, subnet changes, and more.
- In-depth reports on the configuration changes to critical resources such as ELB, RDS, EC2, and more to spot unauthorized changes instantly and stop data leaks.
- Details on the changes to files stored on the S3 bucket to ensure integrity of the files stored on these resources.
- Exhaustive S3 and ELB traffic details so you can gain visibility into who is accessing what resources in your AWS environment.

$ Get Quote Download Schedule a demo