Managing Security Vulnerabilities within Google Cloud

This document provides recommendations on handling security vulnerabilities identified for the DIT Google managed projects currently implemented under the umd.edu organization. The purpose of this document is to provide guidance on how to prevent or remediate the most commonly found security risks within google projects

What is a security vulnerability

A security vulnerability, in the context of Google, refers to a weakness or flaw within the system's code, configuration, or processes that can be exploited by attackers to compromise security. These vulnerabilities can lead to unauthorized access, data breaches, or disruption of services.

Vulnerabilities of all sizes can result in data leaks, and eventually, data breaches—both of which pose significant data security risks for organizations. What is a data leak? A data leak occurs when data is accidentally leaked from within an organization, as opposed to a data breach, which is the result of data being stolen. Data leakage is usually the result of a mistake. For example: sending a document with sensitive or confidential information to the wrong email recipient, saving the data to a public cloud file share, or having data on an unlocked device in a public place for others to see.

Security vulnerability examples

There are a number of Security Vulnerabilities, but some common examples are:

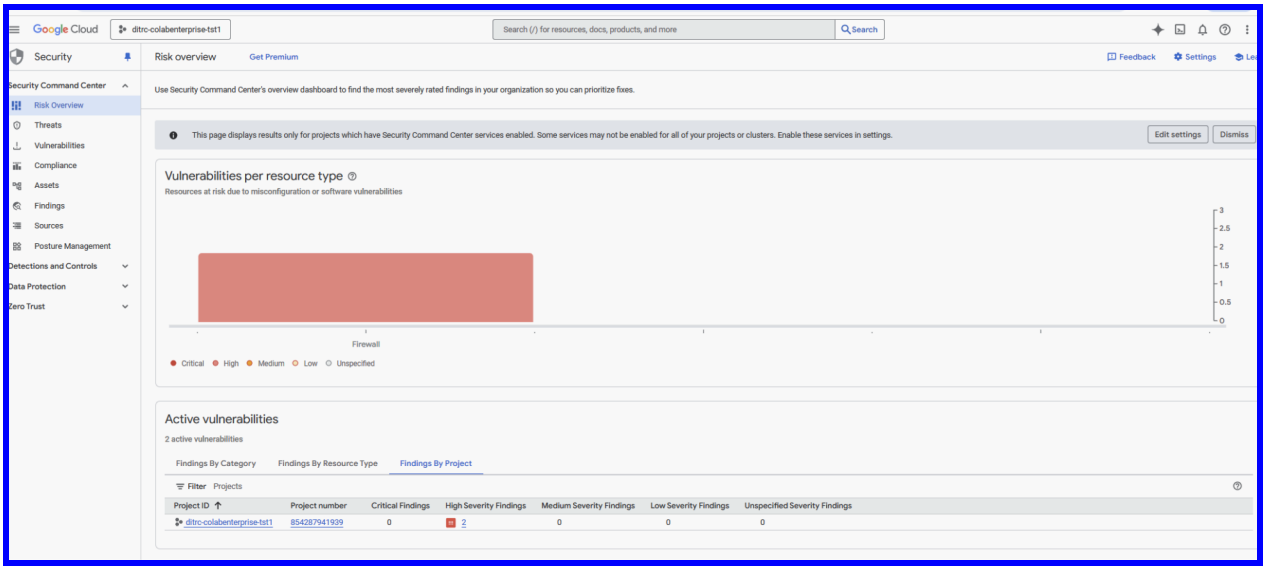
- **Security Misconfiguration:** Any component of a security system that can be leveraged by attackers due to a configuration error can be considered a “Security Misconfiguration.”
- **Broken Authentication**: When authentication credentials are compromised, user sessions and identities can be hijacked by malicious actors to pose as the original user.
- **SQL Injection**: As one of the most prevalent security vulnerabilities, SQL injections attempt to gain access to database content via malicious code

injection. A successful SQL injection can allow attackers to steal sensitive data, spoof identities, and participate in a collection of other harmful activities.

- **Cross-Site Scripting**: Much like an SQL Injection, a Cross-site scripting (XSS) attack also injects malicious code into a website. However, a Cross-site scripting attack targets website users, rather than the actual website itself, which puts sensitive user information at risk of theft.
- **Cross-Site Request Forgery**: A Cross-Site Request Forgery (CSRF) attack aims to trick an authenticated user into performing an action that they do not intend to do. This, paired with social engineering, can deceive users into accidentally providing a malicious actor with personal data.

SELF SERVICE SECURITY VULNERABILITY CHECKS

To identify vulnerabilities for UMD Google Cloud managed projects, the project's point of contact can monitor the project using the Security Command Center service which is available through the Google console.



The most frequently found vulnerabilities are

- **OPEN RDP PORT**
- **OPEN SSH PORT**
- **OPEN FIREWALL**
- **PUBLIC IP ADDRESS**
- **NON-ORG IAM MEMBER**
- **PUBLIC BUCKET ACL**
- **SSL NOT ENFORCED**
- **DATAPROC IMAGE OUTDATED**

These vulnerabilities usually occur when configuring cloud services like compute engine or cloud storage. These vulnerabilities are **high-risk** and could seriously compromise the security of your project system if left unaddressed. We strongly recommend that you remediate these vulnerabilities when identified.

Below are resolutions provided by Google for these vulnerabilities.

1. Vulnerability: Open RDP Port

- **Description:** The firewall rules currently allow any IP address to connect to RDP ports (TCP 3389, UDP 3389), which exposes your RDP services to potential attackers.
- **Recommendation**

➢ Go to the firewall rules page.
➢ Click EDIT.
➢ Delete 0.0.0.0/0.
➢ Add specific IP addresses or IP ranges you want to allow to connect to your instance.
➢ Add specific protocols and ports you want to open on your instance.
➢ Click SAVE

2. Vulnerability: Open SSH Port

- **Description:** Firewall rules that allow any IP address to connect to SSH ports may expose your SSH services to attackers. The SSH service ports are: SCTP: 22 and TCP: 22
- **Recommendation**

❖ Go to the firewall rules page.
❖ Click EDIT.
❖ Delete 0.0.0.0/0.
❖ Add specific IP addresses or IP ranges you want to allow to connect to your instance.
❖ Add specific protocols and ports you want to open on your instance.
❖ Click SAVE

3. Vulnerability: Open Firewall

- **Description:** This firewall rule allows all IP addresses. Firewall rules that allow connections on all ports or from all IP addresses, like 0.0.0.0/0, can unnecessarily expose resources to attacks from unintended sources. These rules should be removed or scoped explicitly to the intended source IP ranges
- **Recommendation**

  To restrict the firewall rules:

❖ Go to the Firewall rule page.
❖ Edit the rule to restrict the range of IPs that are allowed.

4. Vulnerability: Public IP Address

- **Description:** To reduce the attack surface, avoid assigning public IP addresses to your VMs. Stopped instances may still be flagged with a Public IP finding, e.g. if the network interfaces are configured to assign an ephemeral public IP on start. Ensure the network configurations for stopped instances do not include external access.
- **Recommendation**

❖ Go to the Compute instance details page.
❖ Click EDIT.
❖ For each interface under Network interfaces, set External IP to None.
❖ Click Done and then click Save.

5. Vulnerability: Non-Org IAM Member

- **Description:** A user outside of your organization or project has IAM permissions on a project or organization. This is because a **Gmail** email is being used to access the project.
- **Recommendation**

❖ Go to the IAM page.
❖ For each of the following principals:

Principal user	**Conflicting Roles**
mary.jane.doe@gmail.com	roles/billing.projectManager roles/compute.admin roles/owner

❖ Click Remove.

6. Vulnerability: Public Bucket ACL

- **Description:** This bucket is public and can be accessed by anyone on the internet. **allUsers** represents anyone on the Internet, and **allAuthenticatedUsers** represents anyone who is authenticated with a Google account; neither is constrained to users within your organization.
- **Recommendation**

❖ Go to the bucket's permissions page.
❖ Remove allUsers and allAuthenticatedUsers from the bucket's principals.

Optional solution.

- ❖ Go to the bucket's permissions page.
- ❖ Click on the PREVENT PUBLIC ACCESS link.

7. Vulnerability: SSL not enforced

- **Description:** To avoid leaking sensitive data in transit through unencrypted communications, all incoming connections to your SQL database instance should use SSL.
- **Recommendation**
- ❖ Go to the SQL Instance overview page.
- ❖ Verify that the instance is running by looking for the next to the instance name, otherwise start the instance by clicking on the START button.
- ❖ The instance must be running in order to perform the remaining remediation steps, but need not remain running afterwards.
- ❖ Go to the SQL Instance.
- ❖ On the Connections tab, click either Allow only SSL connections or Require trusted client certificates. For more information, see Enforce SSL/TLS encryption

It is recommended to use the client certificate, for more information see Create a new client certificate

Vulnerability: Dataproc image outdated

- **Description:** Dataproc images with subminor image version numbers prior to 1.4.77, 1.5.53, or 2.0.27 are impacted by security vulnerabilities in the open-source Apache "Log4j 2" utility (CVE-2021-44228 and CVE-2021-45046). It is strongly recommended to recreate clusters with an unaffected version.
- **Recommendation**
- ❖ Go to the Dataproc documentation on cluster recreation for instructions on recreating clusters with an updated image version.

Note these are recommendations not required to stay functioning within the environment. However, if you do not accept the recommended resolution, you should find an option to remediate the vulnerability.

DIT RESEARCH COMPUTING MONITORS GOOGLE CLOUD ENVIRONMENT

Periodically, the Division of Information Technology(DIT) performs security checks within the google environment at the umd.edu organization level.

When that occurs, the project point of contact (POC) will receive a notification via email that a security vulnerability has been detected for your project. An example of a sample email is seen below

Sample Email Notification

Subject: Urgent: Security Vulnerabilities Identified in Your Project '**sampleproject-25**'

Dear user,

Through our security monitoring system, we have identified critical vulnerabilities within your project **'sampleproject-25'** in the Google Security Command Center. These vulnerabilities pose a significant security risk and require immediate remediation.

Below are the identified issues along with their recommended resolutions:

Vulnerability: Open RDP Port

- **Description:** The firewall rules currently allow any IP address to connect to RDP ports (TCP 3389, UDP 3389), which exposes your RDP services to potential attackers.
- **Recommendation**

➢ Go to the firewall rules page.
➢ Click EDIT.
➢ Delete 0.0.0.0/0.
➢ Add specific IP addresses or IP ranges you want to allow to connect to your instance.
➢ Add specific protocols and ports you want to open on your instance.

➢ Click SAVE

2. Vulnerability: Open SSH Port

- **Description:** Firewall rules that allow any IP address to connect to SSH ports may expose your SSH services to attackers. The SSH service ports are: SCTP: 22 and TCP: 22
- **Recommendation**

❖ Go to the firewall rules page.
❖ Click EDIT.
❖ Delete 0.0.0.0/0.
❖ Add specific IP addresses or IP ranges you want to allow to connect to your instance.
❖ Add specific protocols and ports you want to open on your instance.
❖ Click SAVE

If this project is no longer in use, please reply to this email that this project can be deleted from the environment and no other actions are required.

These vulnerabilities are high-risk and could seriously compromise the security of our system if left unaddressed. We strongly recommend that you remediate these vulnerabilities immediately and no later than within 30 days to mitigate the risk of exploitation.

If you require any assistance in implementing these changes, please reach out to DIT research computing via itsupport@umd.edu by submitting a question within an incident ticket. Once changes have been completed, please confirm by replying to this email.

Best regards,

Research Computing