Impact on writes with BigQuery column-level security

ature is covered by the Pre-GA Offerings Terms (/terms/service-terms#1) of the Google Cloud Platform Terms
e. Pre-GA features may have limited support, and changes to pre-GA features may not be compatible with oth
sions. For more information, see the launch stage descriptions (/products#product-launch-stages).

This page explains the impact to writes when you use BigQuery column-level security to restrict access to data at the column level. For general information about column-level security, see Introduction to BigQuery column-level security (/bigquery/docs/column-level-security-intro).

Column-level security requires a user to have read permission for columns that are protected by policy tags. Some write operations need to read column data before actually writing into a column. For those operations, BigQuery checks the user's read permission to ensure the user has access to the column. For example, if a user is updating data that includes writing to a protected column, the user must have read permission for the protected column. If the user is inserting a new data row that includes writing to a protected column, the user doesn't need read access for the protected column. But, the user who writes such a row won't be able to read the newly written data unless the user has read permission for the protected columns.

The following sections provide details about different types of write operations. The examples in this topic use `customers` tables with the following schema:

Field name	Type	Mode	Policy tag
`user_id`	STRING	REQUIRED	`policy-tag-1`
`credit_score`	INTEGER	NULLABLE	`policy-tag-2`
`ssn`	STRING	NULLABLE	`policy-tag-3`

Using BigQuery data manipulation language (DML)

For an `INSERT` statement, BigQuery does not check Fine-Grained Reader permission on the policy tags on either the scanned columns or the updated columns. This is because an `INSERT` does not require reading any of the column data. But, even if you successfully insert values into columns where you don't have read permission, once inserted, the values are protected as expected.

For `DELETE`, `UPDATE`, and `MERGE` statements, BigQuery checks Fine-Grained Reader permission on both the scanned columns and updated columns. Note that `DELETE`, even though it is deleting data, scans the table and can be conditional on column values.

Loading data

When loading data (for example, from Cloud Storage or local files) to a table, BigQuery checks Fine-Grained Reader permission on the columns of the destination table. If you don't have access to the destination table columns, BigQuery denies access to load the data.

Streaming is an exception. If you are performing a streaming operation and don't have Fine-Grained Reader permission on the destination table columns, BigQuery will allow you to load the data.

Copying data

For a copy operation, BigQuery checks whether the user has Fine-Grained Reader permission to the source table. BigQuery does not check whether the user has Fine-Grained Reader permission to the columns in the destination table. Note however once the copy is complete, the user won't be able to read the data just written unless the user does have Fine-Grained Reader permission to the destination table.

DML examples

`INSERT`

Example:

```
T INTO customers VALUES('alice', 85, '123-456-7890');
```

	Source columns	Update columns
Policy tags checked for Fine-Grained Reader?	N/A	No

Columns checked	N/A	`user_id`
		`credit_score`
		`ssn`

UPDATE

Example:

```
E customers SET credit_score = 0
RE user_id LIKE 'alice%' AND credit_score < 30
```

	Source columns	Update columns
Policy tags checked for Fine-Grained Reader?	Yes	Yes
Columns checked	`user_id` `credit_score`	`credit_score`

DELETE

Example:

```
E customers WHERE credit_score = 0
```

	Source columns	Update columns
Policy tags checked for Fine-Grained Reader?	Yes	Yes

	Source columns	Update columns

Columns checked	`credit_score`	`user_id`
		`credit_score`
		`ssn`

Load examples

Loading from a local file or Cloud Storage

Example:

```
--source_format=CSV samples.customers \
ustomers_data.csv \
ustomers_schema.json
```

	Source columns	Update columns
Policy tags checked for Fine-Grained Reader?	N/A	Yes

Columns checked	N/A	`user_id`
		`credit_score`
		`ssn`

Streaming

No policy tags are checked when streaming, for either legacy or vortex streaming.

Copy examples

Appending data to an existing table

Example:

```
samples.customers samples.customers_dest
```

	Source columns	Update columns
Policy tags checked for Fine-Grained Reader?	Yes	No

Columns checked	`customers.user_id`
	`customers.credit_score`
	`customers.ssn`
		`customers_dest.user_id`
		`customers_dest.credit_score`
		`customers_dest.ssn`

Saving query results to a destination table

Example:

```
 --use_legacy_sql=false \
tination_table samples.customers_dest \
end_table "SELECT * FROM samples.customers LIMIT 10;"
```

	Source columns	Update columns
Policy tags checked for Fine-Grained Reader?	Yes	Yes

Columns checked	`customers.user_id`
	`customers.credit_score`
	`customers.ssn`
		`customers_dest.user_id`
		`customers_dest.credit_score`
		`customers_dest.ssn`

Last updated 2020-08-18 UTC.