

Apply Filters to SQL Queries

Project description

A potential security issues that involve login attempts and employee machines has been discovered. Using SQL filters I examined the organization's data in the employees and log_in_attempts tables.

Retrieve after hours failed login attempts

I recently discovered a potential security incident that occurred after business hours. To investigate this, I query the log_in_attempts table and review after hours login activity. Using filters in SQL to create a query that identifies all failed login attempts that occurred after 18:00. The time of the login attempt is found in the login_time column. The success column contains a value of 0 when a login attempt failed; it can be either a value of 0 or FALSE in the query to identify failed login attempts. The failed login attempts are 19.

```
Your MariaDB connection id is 41
Server version: 10.3.38-MariaDB-0+deb10u1 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [organization]> clear
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_time > '18:00' AND success = FALSE;
+-----+-----+-----+-----+-----+-----+-----+
| event_id | username | login_date | login_time | country | ip_address | success |
+-----+-----+-----+-----+-----+-----+-----+
| 2 | apatel | 2022-05-10 | 20:27:27 | CAN | 192.168.205.12 | 0 |
| 18 | pwashing | 2022-05-11 | 19:28:50 | US | 192.168.66.142 | 0 |
| 20 | tshah | 2022-05-12 | 18:56:36 | MEXICO | 192.168.109.50 | 0 |
| 28 | aestrada | 2022-05-09 | 19:28:12 | MEXICO | 192.168.27.57 | 0 |
| 34 | drosas | 2022-05-11 | 21:02:04 | US | 192.168.45.93 | 0 |
| 42 | cgriffin | 2022-05-09 | 23:04:05 | US | 192.168.4.157 | 0 |
| 52 | cjackson | 2022-05-10 | 22:07:07 | CAN | 192.168.58.57 | 0 |
| 69 | wjaffrey | 2022-05-11 | 19:55:15 | USA | 192.168.100.17 | 0 |
| 82 | abernard | 2022-05-12 | 23:38:46 | MEX | 192.168.234.49 | 0 |
| 87 | apatel | 2022-05-08 | 22:38:31 | CANADA | 192.168.132.153 | 0 |
| 96 | ivelasco | 2022-05-09 | 22:36:36 | CAN | 192.168.84.194 | 0 |
| 104 | asundara | 2022-05-11 | 18:38:07 | US | 192.168.96.200 | 0 |
| 107 | bisles | 2022-05-12 | 20:25:57 | USA | 192.168.116.187 | 0 |
| 111 | aestrada | 2022-05-10 | 22:00:26 | MEXICO | 192.168.76.27 | 0 |
| 127 | abellmas | 2022-05-09 | 21:20:51 | CANADA | 192.168.70.122 | 0 |
| 131 | bisles | 2022-05-09 | 20:03:55 | US | 192.168.113.171 | 0 |
| 155 | cgriffin | 2022-05-12 | 22:18:42 | USA | 192.168.236.176 | 0 |
| 160 | jclark | 2022-05-10 | 20:49:00 | CANADA | 192.168.214.49 | 0 |
| 199 | yappiah | 2022-05-11 | 19:34:48 | MEXICO | 192.168.44.232 | 0 |
+-----+-----+-----+-----+-----+-----+-----+
19 rows in set (0.063 sec)

MariaDB [organization]> |
```

Retrieve login attempts on specific dates

A suspicious event occurred on 2022-05-09. I investigated this event, by reviewing all login attempts which occurred on this day and the day before. Using filters in SQL to create a query that identifies all login attempts that occurred on 2022-05-09 or 2022-05-08. The date of the login attempt is found in the login_date column. The login attempts on specific dates are 75.

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE login_date = '2022-05-08' OR login_date = '2022-05-09';
```

event_id	username	login_date	login_time	country	ip_address	success
1	jrafael	2022-05-09	04:56:27	CAN	192.168.243.140	1
3	dkot	2022-05-09	06:47:41	USA	192.168.151.162	1
4	dkot	2022-05-08	02:00:39	USA	192.168.178.71	0
8	bisles	2022-05-08	01:30:17	US	192.168.119.173	0
12	dkot	2022-05-08	09:11:34	USA	192.168.100.158	1
15	lyamamot	2022-05-09	17:17:26	USA	192.168.183.51	0
24	arusso	2022-05-09	06:49:39	MEXICO	192.168.171.192	1
25	sbaelish	2022-05-09	07:04:02	US	192.168.33.137	1
26	apatel	2022-05-08	17:27:00	CANADA	192.168.123.105	1
28	aestrada	2022-05-09	19:28:12	MEXICO	192.168.27.57	0
30	yappiah	2022-05-09	03:22:22	MEX	192.168.124.48	1
32	acook	2022-05-09	02:52:02	CANADA	192.168.142.239	0
36	asundara	2022-05-08	09:00:42	US	192.168.78.151	1
38	sbaelish	2022-05-09	14:40:01	USA	192.168.60.42	1
39	yappiah	2022-05-09	07:56:40	MEXICO	192.168.57.115	1
42	cgriffin	2022-05-09	23:04:05	US	192.168.4.157	0
43	mcouliba	2022-05-08	02:35:34	CANADA	192.168.16.208	0
44	daquino	2022-05-08	07:02:35	CANADA	192.168.168.144	0
47	dkot	2022-05-08	05:06:45	US	192.168.233.24	1
49	asundara	2022-05-08	14:00:01	US	192.168.173.213	0
53	nmason	2022-05-08	11:51:38	CAN	192.168.133.188	1
56	acook	2022-05-08	04:56:30	CAN	192.168.209.130	1
58	ivelasco	2022-05-09	17:20:54	CAN	192.168.57.162	0
61	dtanaka	2022-05-09	09:45:18	USA	192.168.98.221	1
65	aalonso	2022-05-09	23:42:12	MEX	192.168.52.37	1
66	aestrada	2022-05-08	21:58:32	MEX	192.168.67.223	1
67	abernard	2022-05-09	11:53:41	MEX	192.168.118.29	1
68	mrah	2022-05-08	17:16:13	US	192.168.42.248	1
70	tmitchel	2022-05-09	10:55:17	MEXICO	192.168.87.199	1
71	mcouliba	2022-05-09	06:57:42	CAN	192.168.55.169	0

Retrieve login attempts outside of Mexico

There's been suspicious activity with login attempts, but the team has determined that this activity didn't originate in Mexico. Now, I checked login attempts that occurred outside of Mexico. Using filters in SQL to create a query that identifies all login attempts that occurred outside of Mexico. When referring to Mexico, the country column contains values of both MEX and MEXICO, and it is necessary to use the LIKE keyword with % to make sure the query reflects this. The login attempts outside of Mexico were 144.

```
MariaDB [organization]> SELECT * FROM log_in_attempts WHERE NOT country LIKE 'MEX%';
```

event_id	username	login_date	login_time	country	ip_address	success
1	jrafael	2022-05-09	04:56:27	CAN	192.168.243.140	1
2	apatel	2022-05-10	20:27:27	CAN	192.168.205.12	0
3	dkot	2022-05-09	06:47:41	USA	192.168.151.162	1
4	dkot	2022-05-08	02:00:39	USA	192.168.178.71	0
5	jrafael	2022-05-11	03:05:59	CANADA	192.168.86.232	0
7	eraab	2022-05-11	01:45:14	CAN	192.168.170.243	1
8	bisles	2022-05-08	01:30:17	US	192.168.119.173	0
10	jrafael	2022-05-12	09:33:19	CANADA	192.168.228.221	0
11	sgilmore	2022-05-11	10:16:29	CANADA	192.168.140.81	0
12	dkot	2022-05-08	09:11:34	USA	192.168.100.158	1
13	mrah	2022-05-11	09:29:34	USA	192.168.246.135	1
14	sbaelish	2022-05-10	10:20:18	US	192.168.16.99	1
15	lyamamot	2022-05-09	17:17:26	USA	192.168.183.51	0
16	mcouliba	2022-05-11	06:44:22	CAN	192.168.172.189	1
17	pwashing	2022-05-11	02:33:02	USA	192.168.81.89	1
18	pwashing	2022-05-11	19:28:50	US	192.168.66.142	0
19	jhill	2022-05-12	13:09:04	US	192.168.142.245	1
21	iuduike	2022-05-11	17:50:00	US	192.168.131.147	1
25	sbaelish	2022-05-09	07:04:02	US	192.168.33.137	1
26	apatel	2022-05-08	17:27:00	CANADA	192.168.123.105	1
29	bisles	2022-05-11	01:21:22	US	192.168.85.186	0
31	acook	2022-05-12	17:36:45	CANADA	192.168.58.232	0
32	acook	2022-05-09	02:52:02	CANADA	192.168.142.239	0
33	zbernal	2022-05-11	02:52:10	US	192.168.72.59	1
34	drosas	2022-05-11	21:02:04	US	192.168.45.93	0
36	asundara	2022-05-08	09:00:42	US	192.168.78.151	1
37	eraab	2022-05-10	06:03:41	CANADA	192.168.152.148	0
38	sbaelish	2022-05-09	14:40:01	USA	192.168.60.42	1
41	apatel	2022-05-10	17:39:42	CANADA	192.168.46.207	0
42	cgriffin	2022-05-09	23:04:05	US	192.168.4.157	0

Retrieve employees in Marketing

The team provided security updates on specific employee machines in the Marketing department. I was responsible for getting information on these employee machines and needed to query the employees table. Using filters in SQL I created a query that identified all employees in the Marketing department for all offices in the East building. The department of the employee is found in the department column, which contains values that include Marketing. The office is found in the office column. Some examples of values in this column are East-170, East-320, and North-434. I used the LIKE keyword with % to filter for the East building. The Marketing employees in the East building machines whom needed security updates were 7.

```
MariaDB [organization]> SELECT * FROM employees WHERE department = 'Marketing' AND office LIKE 'East%';
+-----+-----+-----+-----+-----+
| employee_id | device_id | username | department | office |
+-----+-----+-----+-----+-----+
| 1000 | a320b137c219 | elarson | Marketing | East-170 |
| 1052 | a192b174c940 | jdarosa | Marketing | East-195 |
| 1075 | x573y883z772 | fbautist | Marketing | East-267 |
| 1088 | k8651965m233 | rgosh | Marketing | East-157 |
| 1103 | NULL | randers | Marketing | East-460 |
| 1156 | a184b775c707 | dellery | Marketing | East-417 |
| 1163 | h679i515j339 | cwilliam | Marketing | East-216 |
+-----+-----+-----+-----+-----+
7 rows in set (0.001 sec)
```

Retrieve employees in Finance or Sales

The team needed to implement a different security update on machines for employees in the Sales and Finance departments. Using filters in SQL I created a query that identified all employees in the Sales or Finance departments. The department of the employee is found in the department column, which contains values that include Sales and Finance. The employee machines' from Sales and Finance whom need to implement a different security update were 71.

```
MariaDB [organization]> SELECT * FROM employees WHERE department = 'Finance' OR department = 'Sales';
```

employee_id	device_id	username	department	office
1003	d394e816f943	sgilmore	Finance	South-153
1007	h174i497j413	wjaffrey	Finance	North-406
1008	i858j583k571	abernard	Finance	South-170
1009	NULL	lrodrigu	Sales	South-134
1010	k242l212m542	jlansky	Finance	South-109
1011	l748m120n401	drosas	Sales	South-292
1015	p611q262r945	jsoto	Finance	North-271
1017	r550s824t230	jclark	Finance	North-188
1018	s310t540u653	abellmas	Finance	North-403
1022	w237x430y567	arusso	Finance	West-465
1024	y976z753a267	iuduike	Sales	South-215
1025	z381a365b233	jhill	Sales	North-115
1029	d336e475f676	ivelasco	Finance	East-156
1035	j236k303l245	bisles	Sales	South-171
1039	n253o917p623	cjackson	Sales	East-378
1041	p929q222r778	cgriffin	Sales	North-208
1044	s429t157u159	tbarnes	Finance	West-415
1045	t567u844v434	pwashing	Finance	East-115
1046	u429v921w138	daquino	Finance	West-280
1047	v109w587x644	cward	Finance	West-373
1048	w167x592y375	tmitchel	Finance	South-288
1049	NULL	jreckley	Finance	Central-295
1050	y132z930a114	csimmons	Finance	North-468
1057	f370g535h632	msscott	Sales	South-270
1062	k367l639m697	redwards	Finance	North-180
1063	l686m140n569	lpope	Sales	East-226
1066	o678p794q957	ttyrell	Sales	Central-444
1069	NULL	jpark	Finance	East-110
1071	t244u829v723	zdutchma	Sales	West-348
1072	u905v920w694	esmith	Sales	East-421

Retrieve all employees not in IT

The team provided one more update to employee machines. The employees who are in the Information Technology department already had this update, but employees in all other departments need it. Using filters in SQL I created a query which identified all employees not in the IT department. The department of the employee is found in the department column, which contains values that include Information Technology. The employees machine's out of the Information Technology department whom needed the update were 161.

```
MariaDB [organization]> SELECT * FROM employees WHERE department != 'Information Technology';
```

employee_id	device_id	username	department	office
1000	a320b137c219	elarson	Marketing	East-170
1001	b239c825d303	bmoreno	Marketing	Central-276
1002	c116d593e558	tshah	Human Resources	North-434
1003	d394e816f943	sgilmore	Finance	South-153
1004	e218f877g788	eraab	Human Resources	South-127
1005	f551g340h864	gesparza	Human Resources	South-366
1007	h174i497j413	wjaffrey	Finance	North-406
1008	i858j583k571	abernard	Finance	South-170
1009	NULL	lrodriqu	Sales	South-134
1010	k242l212m542	jlansky	Finance	South-109
1011	l748m120n401	drosas	Sales	South-292
1015	p611q262r945	jsoto	Finance	North-271
1016	q793r736s288	sbaelish	Human Resources	North-229
1017	r550s824t230	jclark	Finance	North-188
1018	s310t540u653	abellmas	Finance	North-403
1020	u899v381w363	arutley	Marketing	South-351
1022	w237x430y567	arusso	Finance	West-465
1024	y976z753a267	iuduike	Sales	South-215
1025	z381a365b233	jhill	Sales	North-115
1026	a998b568c863	apatel	Human Resources	West-320
1027	b806c503d354	mrah	Marketing	West-246
1028	c603d749e374	aestrada	Human Resources	West-121
1029	d336e475f676	ivelasco	Finance	East-156
1030	e391f189g913	mabadi	Marketing	West-375
1031	f419g188h578	dkot	Marketing	West-408
1034	i679j565k940	bsand	Human Resources	East-484
1035	j236k303l245	bisles	Sales	South-171
1036	k550l533m205	rjensen	Marketing	Central-239
1038	m873n636o225	btang	Human Resources	Central-260
1039	n253o917p623	cjackson	Sales	East-378

Summary

I applied filters to SQL queries to get specific information on login attempts and employee machines. I used two different tables, log_in_attempts and employees. I used the AND, OR, and NOT operators to filter for the specific information needed for each task. I also used LIKE and the percentage sign (%) wildcard to filter for patterns.