

graph_redhat

```
1 CREATE TABLE test.t_tmp_craw_redhat(id varchar, vul_msg jsonb)
2
3
4 CREATE TABLE test.t_tmp_node_vul_REDHAT AS
5 WITH wt_cwe AS
6 (
7     SELECT tmp2.cve_id, jsonb_agg(tmp2.weakness_tmp) AS weakness
8     FROM
9     (
10         SELECT tmp1.cve_id, jsonb_build_object('type', NULL, 'cweId', trim(trim(UNNEST(string_to_array(tmp1.cve_id, ','), ''))) AS cve_id
11         FROM
12         (
13             SELECT UNNEST(string_to_array(cr.vul_msg->> 'cwe','->')) cwes, cr.vul_msg ->> 'name' AS cve_id
14             FROM test.t_tmp_craw_redhat cr
15         )tmp1
16     )tmp2
17     GROUP BY tmp2.cve_id
18 ),
19 wt_cvss_msg AS
20 (
21     SELECT ('{"cvssV2_0":{"version":"2.0", "baseScore":"" || COALESCE( (t.vul_msg -> 'cvss' ->> 'cvss_base_score')
22             "vectorString":"" || COALESCE( (t.vul_msg -> 'cvss' ->> 'cvss_scoring_vector'), '') || "
23             "baseSeverity":null, "attackVector":null,
24             "confidentialityImpact":null,
25             "integrityImpact":null, "availabilityImpact":null, "userInteraction":null, "attackComplexity":null,
26             },
27             "cvssV3" || COALESCE( REPLACE(t.cvss3_version, '.', '_'), '3_0') || ":{\"version\":\" || COALESCE( t.cvss3
28             "baseScore":"" || COALESCE( (t.vul_msg -> 'cvss3' ->> 'cvss3_base_score'), '') || "
29             "vectorString":"" || COALESCE( (t.vul_msg -> 'cvss3' ->> 'cvss3_scoring_vector'), '') || "
30             "baseSeverity":null,
31             "attackVector":null,
32             "confidentialityImpact":null,
33             "integrityImpact":null, "availabilityImpact":null, "userInteraction":null, "attackComplexity":null,
34             }
35         }')::jsonb AS severity,
36     -- t.vul_msg -> 'cvss' AS cvss2,
37     -- t.vul_msg -> 'cvss3' AS cvss3
38     t.vul_msg ->> 'name' AS cve_id
39     FROM
40     (
41         SELECT cr.vul_msg, split_part( split_part( cr.vul_msg -> 'cvss3' ->> 'cvss3_scoring_vector', '/', 1),
42         FROM test.t_tmp_craw_redhat cr
43     )t
44 )
45 SELECT cr.vul_msg ->> 'name' AS id,
46        '[' AS aliases,
47        jsonb_build_object('discovery', NULL, 'identifier', null) AS SOURCE,
48        jsonb_build_object('title', cr.vul_msg ->> 'statement', 'details', cr.vul_msg ->> 'details') AS description,
49        COALESCE(wc.weakness, '{}') AS weakness,
```

```

50     COALESCE(cm.severity, '{}') AS severity,
51     jsonb_build_object('published', cr.vul_msg ->> 'public_date', 'lastModified', NULL, 'datePublic', NULL)
52     ('{"impact_info":{"impacts":null, "impactScore":null},
53      "exploit_info":{"exploitable":null, "exploits":null, "exploit_url":null, "exploitabilityScore":null
54      "PoC_info":{"PoC_available":null, "PoC":null, "PoC_url":null},
55      "patch_info":{"patch_available":null, "patch_url":null},
56      "report_status":null}'):::jsonb ||
57     jsonb_build_object('solution_info', cr.vul_msg -> 'mitigation' ->> 'value')
58     AS vul_status
59 FROM test.t_tmp_craw_redhat cr
60 LEFT JOIN wt_cwe wc
61     ON (cr.vul_msg ->> 'name') = wc.cve_id
62 LEFT JOIN wt_cvss_msg cm
63     ON (cr.vul_msg ->> 'name') = cm.cve_id;
64
65 --vul node
66 DELETE FROM test.dws_graph_node_vul WHERE vul_source = 'REDHAT';
67 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
68 INSERT INTO test.dws_graph_node_vul
69 SELECT nextval('cve_graph_seq') AS seq, tmp.*
70 FROM
71 (
72 SELECT DISTINCT id , aliases , "source", description ,weakness, severity, time_info, vul_status, 'REDHAT' AS
73 )tmp ;
74
75 SELECT *FROM test.t_tmp_node_vul_REDHAT LIMIT 10;
76
77 --affected component
78 DROP TABLE test.t_tmp_graph_node_component_redhat ;
79 CREATE TABLE test.t_tmp_graph_node_component_redhat AS
80 WITH wt_upstream_fix AS
81 (
82     SELECT tmp2.MODULE, tmp2.cve_id, '[' || string_agg(DISTINCT tmp2.version_num, ',') || ']' AS upstream_fix_
83     FROM
84     (
85         SELECT split_part(tmp1.mig, ' ', tmp1.blank_count + 1) AS version_num,
86             trim(replace( tmp1.mig, split_part(tmp1.mig, ' ', tmp1.blank_count + 1), '')) AS module,
87             tmp1.*
88     FROM
89     (
90         SELECT ( length(t.mig) - length(replace(t.mig,' ','')) ) AS blank_count, t.*
91     FROM
92     (
93         SELECT trim( UNNEST(string_to_array(
94             trim( UNNEST( string_to_array( cr.vul_msg ->> 'upstream_fix' , ',') )), 'and') )) AS
95             cr.vul_msg ->> 'name' AS cve_id
96         FROM test.t_tmp_craw_redhat cr
97         )t
98         WHERE t.mig IS NOT NULL
99     )tmp1
100 )tmp2
101 GROUP BY tmp2.MODULE, tmp2.cve_id
102 ),
103 wt_package_state AS
104 (
105     SELECT jsonb_build_object('release', tmp1.product_name, 'fixed_version', uf.upstream_fix_version, 'urgency'
106         tmp1.product_name, tmp1.package_name, tmp1.cve_id
107     FROM (

```

```

108     SELECT DISTINCT package_state ->> 'fix_state' AS fix_state, --AS urgency
109         package_state ->> 'package_name' AS package_name, -- AS MODULE
110         package_state ->> 'product_name' AS product_name, -- AS product
111         t.vul_msg ->> 'name' AS cve_id
112     FROM test.t_tmp_craw_redhat t,
113         jsonb_array_elements(t.vul_msg -> 'package_state') package_state
114 )tmp1
115 LEFT JOIN wt_upstream_fix uf
116     ON tmp1.package_name = uf.MODULE
117     AND tmp1.cve_id = uf.cve_id
118 ),
119 wt_affected_release AS
120 (
121     SELECT 'RedHat' AS vendor,
122         COALESCE(affected_release ->> 'package', '') AS package_with_version, --AS version
123         COALESCE (affected_release ->> 'product_name', '*') AS product_name, --AS release
124         regexp_replace(affected_release ->> 'package', '^(.*)-[0-9]+:([0-9]+\.[0-9]+\.[0-9]+).*$', '\1') AS
125         regexp_replace(affected_release ->> 'package', '^(.*)-[0-9]+:([0-9]+\.[0-9]+\.[0-9]+).*$', '\2') AS
126         '*' AS ecosystem,
127         t.vul_msg ->> 'name' AS cve_id
128     FROM test.t_tmp_craw_redhat t,
129         jsonb_array_elements(t.vul_msg -> 'affected_release') affected_release
130 )
131 SELECT ar.cve_id, ar.vendor, ar.ecosystem, ar.product_name, ar.module_name, COALESCE(ps.fixed_versions, '[]',
132     jsonb_build_object('release', ar.product_name, 'version', COALESCE(uf.upstream_fix_version, ar.package_w
133 FROM (SELECT *FROM wt_affected_release WHERE module_name IS NOT NULL) ar --过滤掉没有package的数据
134 LEFT JOIN wt_upstream_fix uf
135     ON ar.module_name = uf.MODULE
136     AND ar.cve_id = uf.cve_id
137 LEFT JOIN wt_package_state ps
138     ON ar.product_name = ps.product_name
139     AND ar.module_name = ps.package_name
140     AND ar.cve_id = ps.cve_id ;
141
142
143 DELETE FROM test.dws_graph_node_affected_component WHERE vul_source = 'REDHAT';
144 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
145 INSERT INTO test.dws_graph_node_affected_component
146     SELECT nextval('cve_graph_seq') AS seq, tmp.*
147 FROM (
148     SELECT DISTINCT product_name, vendor, module_name, ecosystem, 'REDHAT' AS vul_source FROM test.t_tmp_
149 )tmp;
150
151 SELECT *FROM test.t_tmp_graph_node_component_redhat LIMIT 100;
152
153 --affected component
154 --seq int , vul_id , component_name , vendor , package_name , ecosystem , repo_url ,
155 --platform , collectionUrl , defaultStatus , affected_versions , unaffected_versions , vul_source
156 DELETE FROM test.dws_graph_relationships_affected_components WHERE vul_source = 'REDHAT';
157 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
158 INSERT INTO test.dws_graph_relationships_affected_components
159     SELECT nextval('cve_graph_seq') AS seq, tmp.*
160 FROM (
161     SELECT DISTINCT cve_id, product_name, vendor, module_name, ecosystem, '' repo_url, '' AS platform,
162         '' AS collectionurl, '' AS defaultstatus, affected_versions, COALESCE(fixed_versions, '{}'), 'REDHA
163 )tmp;
164
165 --reference

```

```

166 CREATE TABLE test.t_tmp_graph_node_refs_REDHAT AS
167 SELECT t.vul_msg ->> 'name' AS cve_id, UNNEST(string_to_array( refs ->> 0 , e'\n'))AS ref_url
168 FROM test.t_tmp_craw_redhat t,
169 jsonb_array_elements(t.vul_msg -> 'references') refs
170
171 --node seq int, url varchar, vul_source varchar
172 DELETE FROM test.dws_graph_node_refs WHERE vul_source = 'REDHAT';
173 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
174 INSERT INTO test.dws_graph_node_refs
175 SELECT nextval('cve_graph_seq') AS seq, tmp.*
176 FROM (
177 SELECT DISTINCT ref_url, 'REDHAT' AS source FROM test.t_tmp_graph_node_refs_REDHAT
178 )tmp;
179
180 --relationship seq int , vul_id varchar, url varchar, tags varchar, description varchar, vul_source varchar
181 DELETE FROM test.dws_graph_relationships_refs WHERE vul_source = 'REDHAT';
182 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
183 INSERT INTO test.dws_graph_relationships_refs
184 SELECT nextval('cve_graph_seq') AS seq, tmp.*
185 FROM (
186 SELECT DISTINCT cve_id AS id, Ref_url , '' AS tags, '' AS ref_desc , 'REDHAT' AS vul_source FROM test.
187 )tmp;
188
189
190
191
192 --cwe
193 CREATE TABLE test.t_tmp_graph_node_cwe_REDHAT AS
194 SELECT tmp1.cve_id, trim(trim(UNNEST(string_to_array(tmp1.cwes, '|')), '('), ')') AS cwe_id
195 FROM
196 (
197 SELECT UNNEST(string_to_array(cr.vul_msg->> 'cwe','->')) cwes, cr.vul_msg ->> 'name' AS cve_id
198 FROM test.t_tmp_craw_redhat cr
199 )tmp1
200
201 --cwe node
202 --seq , id , vul_source
203 DELETE FROM test.dws_graph_node_cwe WHERE vul_source = 'REDHAT';
204 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
205 INSERT INTO test.dws_graph_node_cwe
206 SELECT nextval('cve_graph_seq') AS seq, tmp.*
207 FROM (
208 SELECT DISTINCT cwe_id, 'REDHAT' FROM test.t_tmp_graph_node_cwe_REDHAT
209 )tmp;
210
211 --relationships cwe
212 -- seq int , vul_id varchar, cwe_id varchar, cwe_type varchar, cwe_desc varchar, vul_source varchar
213 DELETE FROM test.dws_graph_relationships_cwe WHERE vul_source = 'REDHAT';
214 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
215 INSERT INTO test.dws_graph_relationships_cwe
216 SELECT nextval('cve_graph_seq') AS seq, tmp.*
217 FROM (
218 SELECT DISTINCT cve_id AS id, cwe_id , '' AS cwe_type, '' AS cwe_desc , 'REDHAT' AS vul_source FROM te
219 )tmp;

```