

graph_NVD

```
2 CREATE SEQUENCE cve_graph_seq
3     START WITH 1
4     INCREMENT BY 1;
5
6 ##### node: vuln_nvd 构建
7 DROP TABLE test.t_tmp_nvd_impact_exploit_score;
8 CREATE TABLE test.t_tmp_nvd_impact_exploit_score AS
9 WITH wt_cvss_score AS
10 (
11 SELECT dns.nvd_id , dns.cve_msg -> 'metrics' -> 'cvssMetricV2' -> 0 -> 'impactScore' AS impact_score,
12        dns.cve_msg -> 'metrics' -> 'cvssMetricV2' -> 0 -> 'exploitabilityScore' AS exploitabilityScore,
13        jsonb_build_object('source',dns.cve_msg -> 'metrics' -> 'cvssMetricV2' -> 0 -> 'source' ,
14        'type',dns.cve_msg -> 'metrics' -> 'cvssMetricV2' -> 0 -> 'type' ) AS score_source_type
15 FROM warehouse.dwd_nvd_source_f dns
16 --WHERE dns.nvd_id = 'CVE-1999-0095'
17 UNION all
18 SELECT dns.nvd_id , dns.cve_msg -> 'metrics' -> 'cvssMetricV30' -> 0 -> 'impactScore' ,
19        dns.cve_msg -> 'metrics' -> 'cvssMetricV30' -> 0 -> 'exploitabilityScore' ,
20        jsonb_build_object('source',dns.cve_msg -> 'metrics' -> 'cvssMetricV30' -> 0 -> 'source' ,
21        'type',dns.cve_msg -> 'metrics' -> 'cvssMetricV30' -> 0 -> 'type' )
22 FROM warehouse.dwd_nvd_source_f dns
23 --WHERE dns.nvd_id = 'CVE-1999-0095'
24 UNION ALL
25 SELECT dns.nvd_id , dns.cve_msg -> 'metrics' -> 'cvssMetricV31' -> 0 -> 'impactScore' ,
26        dns.cve_msg -> 'metrics' -> 'cvssMetricV31' -> 0 -> 'exploitabilityScore' ,
27        jsonb_build_object('source',dns.cve_msg -> 'metrics' -> 'cvssMetricV31' -> 0 -> 'source' ,
28        'type',dns.cve_msg -> 'metrics' -> 'cvssMetricV31' -> 0 -> 'type' )
29 FROM warehouse.dwd_nvd_source_f dns
30 --WHERE dns.nvd_id = 'CVE-1999-0095'
31 ),
32 wt_impact AS
33 (
34     SELECT im_tmp.nvd_id, jsonb_agg(im_tmp.impact_msg) AS impact_msg_array
35     FROM (SELECT wcs.nvd_id, jsonb_build_object( wcs.impact_score, jsonb_agg(DISTINCT score_source_type)) AS i
36           FROM wt_cvss_score wcs
37           WHERE wcs.impact_score IS NOT NULL
38           GROUP BY wcs.nvd_id, wcs.impact_score
39           )im_tmp
40     GROUP BY im_tmp.nvd_id
41 ),
42 wt_exploit_Score AS
43 (
44     SELECT ex_tmp.nvd_id, jsonb_agg(ex_tmp.exploit_msg) AS exploit_msg_array
45     FROM (
46         SELECT wcs.nvd_id, jsonb_build_object( wcs.exploitabilityScore, jsonb_agg(DISTINCT score_source_type) )
47         FROM wt_cvss_score wcs
48         WHERE wcs.exploitabilityScore IS NOT NULL
49         GROUP BY wcs.nvd_id, wcs.exploitabilityScore
```

```

50     )ex_tmp
51     GROUP BY ex_tmp.nvd_id
52 )
53 SELECT t.nvd_id , wi.impact_msg_array, we.exploit_msg_array
54 FROM warehouse.dwd_nvd_source_f t
55 LEFT JOIN wt_impact wi
56     ON t.nvd_id = wi.nvd_id
57 LEFT JOIN wt_exploit_Score we
58     ON t.nvd_id = we.nvd_id;
59
60 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
61 DROP TABLE IF EXISTS test.t_tmp_grahp_node_vuln_nvd ;
62 CREATE TABLE test.t_tmp_grahp_node_vuln_nvd AS
63 WITH wt_weakness AS
64 (
65     SELECT wn.nvd_id, jsonb_agg(wn.weakness_msg) AS weakness_msg_array
66 FROM
67     (
68         SELECT dns.nvd_id, jsonb_build_object('type', weaknesses.value ->> 'type' ,
69                                             'cweId', weaknesses.value -> 'description' -> 0 ->> 'value' ,
70                                             'description', NULL) AS weakness_msg
71         FROM warehouse.dwd_nvd_source_f dns
72         LEFT JOIN jsonb_array_elements(dns.cve_msg -> 'weaknesses') weaknesses ON 1=1
73     )wn
74     GROUP BY wn.nvd_id
75 )
76 ,nvd_vul_tmp AS
77 (
78 SELECT
79     dns.cve_msg ->> 'id' AS id,
80     -- null AS name, 在创建node的时候,给name属性值为空
81     '[]' AS aliases,
82     dns.cve_msg ->> 'sourceIdentifier' AS source_Identifier,
83     dns.cve_msg -> 'descriptions' -> 0 ->> 'value' AS descriptions_details,
84     jsonb_build_object('cvssV4_0', dns.cve_msg -> 'metrics' -> 'cvssMetricV40' -> 0 -> 'cvssData' )
85     AS cvssMetricV40,
86     jsonb_build_object('cvssV3_1', dns.cve_msg -> 'metrics' -> 'cvssMetricV31' -> 0 -> 'cvssData' )
87     AS cvssMetricV31,
88     jsonb_build_object('cvssV3_0', dns.cve_msg -> 'metrics' -> 'cvssMetricV30' -> 0 -> 'cvssData' )
89     AS cvssMetricV30,
90     jsonb_build_object('cvssV2_0', dns.cve_msg -> 'metrics' -> 'cvssMetricV2' -> 0 -> 'cvssData'
91                         || ('{"baseSeverity": "' || (dns.cve_msg -> 'metrics' -> 'cvssMetricV2' -> 0 ->>
92                         || (',"userInteraction":"' || (dns.cve_msg -> 'metrics' -> 'cvssMetricV2' -> 0 ->
93     AS cvssMetricV2,
94     dns.cve_msg ->> 'lastModified' AS timeinfo_lastModified,
95     dns.cve_msg ->> 'published' AS timeinfo_published,
96     dns.cve_msg ->> 'vulnStatus' AS vulnStatus,
97     -- dns.cve_msg ->> 'metrics' AS metrics,
98     -- dns.cve_msg ->> 'configurations' AS configurations,
99     'https://services.nvd.nist.gov/rest/json/cves/2.0?cveId=' || (dns.cve_msg ->> 'id')::varchar AS api_url
100 FROM warehouse.dwd_nvd_source_f dns
101 )
102 SELECT nextval('cve_graph_seq') AS seq, t.id, t.aliases, jsonb_build_object('discovery', NULL, 'identifier', t.
103     jsonb_build_object('title', NULL, 'details', descriptions_details) AS description_msg,
104     ww.weakness_msg_array,
105     ( CASE WHEN (cvssMetricV40 -> 'cvssV4_0') = 'null' THEN '{}' ELSE cvssMetricV40 END
106     || CASE WHEN (cvssMetricV31 -> 'cvssV3_1') = 'null' THEN '{}' ELSE cvssMetricV31 END
107     || CASE WHEN (cvssMetricV30 -> 'cvssV3_0') = 'null' THEN '{}' ELSE cvssMetricV30 END

```

```

108      || CASE WHEN (cvssMetricV2 -> 'cvssV2_0') = 'null' THEN '{}' ELSE cvssMetricV2 END)::jsonb AS severity,
109      jsonb_build_object('published', t.timeinfo_published, 'lastModified', t.timeinfo_lastModified, 'datePubli
110      jsonb_build_object('impact_info', jsonb_build_object('impacts', NULL, 'impactScore', ies.impact_msg_array)
111                      'solution_info', NULL ,
112                      'exploit_info', jsonb_build_object('exploitable', NULL, 'exploits', NULL, 'exploit_url'
113                      'PoC_info', jsonb_build_object('PoC_available', NULL, 'PoC', NULL, 'PoC_url', NULL),
114                      'patch_info', jsonb_build_object('patch_available', NULL, 'patch_url', null),
115                      'report_status', null
116      ) AS vul_status, t.vulnStatus, t.api_url
117 FROM nvd_vul_tmp t
118 LEFT JOIN test.t_tmp_nvd_impact_exploit_score ies
119      ON t.id = ies.nvd_id
120 LEFT JOIN wt_weakness ww
121      ON t.id = ww.nvd_id ;
122
123
124 DELETE FROM test.dws_graph_node_vul WHERE vul_source = 'NVD';
125 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
126 INSERT INTO test.dws_graph_node_vul
127 SELECT nextval('cve_graph_seq') AS seq, tmp.*
128 FROM
129 (
130 SELECT DISTINCT id , aliases , source_msg, description_msg ,weakness_msg_array, severity, time_info, vul_st
131 )tmp ;
132 bash gen_graph_data.sh "NVD" "node" "vul"
133 bash neo4j_vul_node_load.sh "NVD" "0"
134
135
136
137 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
138
139 DROP TABLE IF EXISTS test.t_tmp_cve_graph_cwe_nodes ;
140 CREATE TABLE test.t_tmp_cve_graph_cwe_nodes AS
141 SELECT nextval('cve_graph_seq') AS seq,
142      tmp.*
143 FROM
144 (
145 SELECT DISTINCT dns.nvd_id,
146      weaknesses.value ->> 'type' AS cwe_type,
147      weaknesses.value ->> 'source' AS cwe_source,
148      weaknesses.value -> 'description' -> 0 ->> 'value' AS cwe_id
149 FROM warehouse.dwd_nvd_source_f dns ,
150      jsonb_array_elements(dns.cve_msg -> 'weaknesses') weaknesses
151 --      ,jsonb_array_elements(weaknesses.value -> 'description') des
152 )tmp ;
153
154 SELECT *FROM test.t_tmp_cve_graph_cwe_nodes LIMIT 10
155
156 DELETE FROM test.dws_graph_node_cwe WHERE vul_source = 'NVD';
157 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
158 INSERT INTO test.dws_graph_node_cwe
159 SELECT nextval('cve_graph_seq') AS seq, tmp.*
160 FROM (
161 SELECT DISTINCT cwe_id, 'NVD' FROM test.t_tmp_cve_graph_cwe_nodes
162 )tmp;
163 bash gen_graph_data.sh "NVD" "node" "cwe"
164 bash neo4j_cwe_node_load.sh "NVD" "0"
165

```

```

166
167 DELETE FROM test.dws_graph_relationships_cwe WHERE vul_source = 'NVD';
168 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
169 INSERT INTO test.dws_graph_relationships_cwe
170     SELECT nextval('cve_graph_seq') AS seq, tmp.*
171     FROM (
172         SELECT DISTINCT nvd_id, cwe_id, cwe_type, cwe_source, 'NVD' AS vul_source FROM test.t_tmp_cve_graph_
173     )tmp;
174 bash gen_graph_data.sh "NVD" "relationships" "cwe"
175 bash neo4j_relationships_cwe_load.sh "NVD" "0"
176
177 --\copy (SELECT DISTINCT cwe_id FROM test.t_tmp_cve_graph_cwe_nodes) to 'cwe_nodes.csv' with (delimiter ',', F
178 --
179 --LOAD CSV WITH HEADERS FROM 'file:///cwe_nodes.csv' AS row
180 --CREATE (n:cwe {id: row.cwe_id})
181 --
182 --LOAD CSV WITH HEADERS FROM 'file:///cwe_nodes.csv' AS row
183 --MERGE (cwe:CWE {id: row.cwe_id})
184 --SET cwe.description = row.description,
185 --     cwe.severity = row.severity
186 --RETURN cwe;
187 --
188 --
189
190 --\copy (SELECT DISTINCT nvd_id AS r_start, cwe_id AS r_end, cwe_type, cwe_source FROM test.t_tmp_cve_graph_c
191 --\copy (SELECT DISTINCT nvd_id AS r_start, cwe_id AS r_end, cwe_type, cwe_source FROM test.t_tmp_cve_graph_c
192 --\copy (SELECT DISTINCT nvd_id AS r_start, cwe_id AS r_end, cwe_type, cwe_source FROM test.t_tmp_cve_graph_c
193 --
194
195 --LOAD CSV WITH HEADERS FROM 'file:///cwe_ref.csv' AS row
196 --     MATCH (cve:Vuln_NVD {id: row.r_start})
197 --     MATCH (cwe:CWE {id: row.r_end})
198 --     MERGE (cve)-[:HAS_CWE{cweId: row.r_end, type: row.cwe_type, description: row.cwe_source}]->(cwe)
199
200
201
202
203
204
205
206
207
208 --https://lists.apache.org/thread.html/r0276683d8e1e07153fc8642618830ac0ade85b9ae0dc7b07f63bb8fc%40%3Ccvns.httpd
209 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
210 DROP TABLE test.t_tmp_cve_graph_ref_nodes ;
211 CREATE TABLE test.t_tmp_cve_graph_ref_nodes AS
212 SELECT nextval('cve_graph_seq') AS seq,
213         tmp.*
214 FROM
215 (
216 SELECT DISTINCT dns.nvd_id,
217         refs.value ->> 'url' AS ref_url,
218         refs.value ->> 'source' AS ref_source,
219         refs.value ->> 'tags' AS ref_tags
220 FROM warehouse.dwd_nvd_source_f dns,
221         jsonb_array_elements(dns.cve_msg -> 'references') refs
222 )tmp ;
223

```

```

224 SELECT t.*
225 FROM test.t_tmp_cve_graph_ref_nodes t LIMIT 10
226
227 DELETE FROM test.dws_graph_node_refs WHERE vul_source = 'NVD';
228 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
229 INSERT INTO test.dws_graph_node_refs
230     SELECT nextval('cve_graph_seq') AS seq, tmp.*
231     FROM (
232         SELECT DISTINCT ref_url, 'NVD' AS source FROM test.t_tmp_cve_graph_ref_nodes
233     )tmp;
234 bash gen_graph_data.sh "NVD" "node" "refs"
235 bash neo4j_refs_node_load.sh "NVD" "0"
236
237
238 DELETE FROM test.dws_graph_relationships_refs WHERE vul_source = 'NVD';
239 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
240 INSERT INTO test.dws_graph_relationships_refs
241     SELECT nextval('cve_graph_seq') AS seq, tmp.*
242     FROM (
243         SELECT DISTINCT nvd_id, ref_url, ref_tags, ref_source, 'NVD' AS vul_source FROM test.t_tmp_cve_graph
244     )tmp;
245 bash gen_graph_data.sh "NVD" "relationships" "refs"
246 bash neo4j_relationships_refs_load.sh "NVD" "0"
247
248 --\copy (SELECT DISTINCT ref_url FROM test.t_tmp_cve_graph_ref_nodes) to 'ref_nodes.csv' with (delimiter ',',
249 --LOAD CSV WITH HEADERS FROM 'file:///ref_nodes.csv' AS row
250 --MERGE (ref:references {url: row.ref_url});
251 --
252 --create index for (n:references) on (n.url);
253 --
254
255 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
256 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
257 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
258 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
259 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
260 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
261 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
262 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
263 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
264 --\copy (SELECT DISTINCT nvd_id AS r_start, ref_url AS r_end, ref_source, ref_tags FROM test.t_tmp_cve_graph
265 --
266
267 --LOAD CSV WITH HEADERS FROM 'file:///ref_10w_9.csv' AS row
268 --    MATCH (cve:Vuln_NVD {id: row.r_start})
269 --    MATCH (ref:references {url: row.r_end})
270 --    MERGE (cve)-[:HAS_REFERENCE{description: row.ref_source,tags: COALESCE(row.ref_tags, '')}]->(ref
271
272
273
274
275
276 DROP TABLE test.t_tmp_grahp_node_affected_msg;
277 CREATE TABLE test.t_tmp_grahp_node_affected_msg AS
278 WITH wt_cpe_match AS
279 (
280     SELECT replace(cve_cpeMatch.value ->> 'criteria', E'\\:', '|+') AS cpe_value,
281     cve_cpeMatch.value AS cpe_match_val,

```

```

282         dnsf.nvd_id ,
283         (cve_cpeMatch.value ->> 'criteria')::varchar AS cpe_source_val
284         -- ,ORDINALITY AS cve_nodes_index
285     FROM warehouse.dwd_nvd_source_f dnsf ,
286     jsonb_array_elements(dnsf.cve_msg -> 'configurations') cve_conf, --WITH ORDINALITY cve_conf,
287     jsonb_array_elements(cve_conf.value -> 'nodes') cve_nodes ,
288     jsonb_array_elements(cve_nodes.value -> 'cpeMatch') cve_cpeMatch
289 )
290 ,
291 wt_ecosystem_map AS (
292     select 'node.js' as key ,    'npm' as value union all
293     select 'python' as key ,    'PyPI' as value union all
294     select 'ruby' as key ,    'RubyGems' as value union all
295     select 'java' as key ,    'Maven' as value union all
296     select 'go' as key ,    'Go' as value union all
297     select 'nuget' as key ,    'NuGet' as value union all
298     select 'php' as key ,    'Packagist' as value union all
299     select 'rust' as key ,    'crates.io' as value union all
300     select 'android' as key ,    'Android' as value union all
301     select 'debian' as key ,    'Debian' as value union all
302     select 'linux' as key ,    'Linux' as value union all
303     select 'alpine' as key ,    'Alpine' as value union all
304     select 'git' as key ,    'GIT' as value union all
305     select 'oss-fuzz' as key ,    'OSS-Fuzz'
306 ),
307 wt_cpe_parse AS
308 (
309     SELECT cm.cpe_value,
310            split_part(cm.cpe_value, ':', 3) AS part,
311            REPLACE(split_part(cm.cpe_value, ':', 4), '|+', ':') AS vendor,
312            REPLACE(split_part(cm.cpe_value, ':', 4), '|+', '\\:') AS vendor_source,
313            replace(split_part(cm.cpe_value, ':', 5), '|+', ':') AS product,
314            replace(split_part(cm.cpe_value, ':', 5), '|+', '\\:') AS product_source,
315     --    split_part(cm.cpe_value, ':', 5) AS product,
316            split_part(cm.cpe_value, ':', 6) AS version,
317            split_part(cm.cpe_value, ':', 7) AS update,
318            split_part(cm.cpe_value, ':', 8) AS edition,
319            split_part(cm.cpe_value, ':', 9) AS language,
320            split_part(cm.cpe_value, ':', 10) AS sw_edition,
321            split_part(cm.cpe_value, ':', 11) AS target_sw,
322            split_part(cm.cpe_value, ':', 12) AS target_hw,
323            split_part(cm.cpe_value, ':', 13) AS other,
324            cm.nvd_id ,
325            cm.cpe_source_val,
326            cm.cpe_match_val
327            -- ,cm.cve_nodes_index
328     FROM wt_cpe_match cm
329 )
330 ,
331 wt_cpe_process AS
332 (
333     SELECT cp1.cpe_match_val,
334            (CASE WHEN cp1.versionStartExcluding IS NOT NULL THEN '(' || cp1.versionStartIncluding
335                 WHEN cp1.versionStartIncluding IS NOT NULL THEN '[' || cp1.versionStartIncluding
336                 ELSE '({0' END )
337            || (CASE WHEN cp1.versionEndExcluding IS NOT NULL THEN ', ' || cp1.versionEndExcluding || ')'
338                 WHEN cp1.versionEndIncluding IS NOT NULL THEN ', ' || cp1.versionEndIncluding || ']' ELSE ',-
339            cp1.package_name,

```

```

340         cp1.part || ':' || cp1.vendor_source || ':' || cp1.package_name AS unique_package,
341         cp1.version_op,
342         cp1.matchCriteriaId,
343         cp1.nvd_id,
344         cp1.vulnerable,
345         cp1.vendor,
346         cp1.cpe_source_val,
347         -- , cp1.cve_nodes_index
348         cp1.ecosystem
349 FROM
350 (
351     SELECT CASE WHEN cp.target_sw <> '*' THEN cp.product_source || '-' || cp.target_sw ELSE cp.product END
352            CASE WHEN cp.target_sw <> '*' AND em.KEY IS NOT NULL THEN em.value ELSE '*' END AS ecosystem,
353            cp.VERSION || (CASE WHEN cp.VERSION <> '*' AND cp.UPDATE <> '*' THEN '.' || cp.UPDATE ELSE '' END
354            || (CASE WHEN cp.VERSION <> '*' AND cp.edition <> '*' THEN '.' || cp.edition ELSE ''
355            cp.cpe_match_val -> 'vulnerable' AS vulnerable,
356            cp.cpe_match_val -> 'versionStartExcluding' AS versionStartExcluding,
357            cp.cpe_match_val -> 'versionStartIncluding' AS versionStartIncluding,
358            cp.cpe_match_val -> 'versionEndExcluding' AS versionEndExcluding,
359            cp.cpe_match_val -> 'versionEndIncluding' AS versionEndIncluding,
360            cp.cpe_match_val -> 'matchCriteriaId' AS matchCriteriaId,
361            cp.nvd_id,
362            cp.cpe_match_val,
363            cp.vendor
364            ,cp.cpe_source_val
365            ,cp.part, cp.vendor_source
366            -- ,cp.cve_nodes_index
367            FROM wt_cpe_parse cp
368            LEFT JOIN wt_ecosystem_map em
369            ON cp.target_sw = em.key
370    )cp1
371 )
372 , wt_cve_affect_unnest AS
373 (
374     SELECT tmp.nvd_id, tmp.unique_package, tmp.package_name, tmp.vendor , tmp.vulnerable, tmp.ecosystem,
375            array_to_string(array_agg(tmp.version_events), ',') AS cve_affect
376            ,sum(tmp.version_flag) AS version_flag_sum
377            , '{' || array_to_string( array_agg(tmp.cpe_source_val_list) , ',') || '}' AS cpe_source_val_list
378 --      json_build_object('version_events', tmp.version_events , 'vulnerable', tmp.vulnerable) AS json_affe
379 --      tmp.cve_nodes_index
380 --      SELECT      tmp.nvd_id, tmp.unique_package, tmp.package_name, tmp.vendor , tmp.cve_nodes_index
381 FROM
382 (
383     SELECT cp.nvd_id , cp.unique_package, cp.package_name, cp.vendor, '{' || array_to_string(array_agg(DI
384            ,array_to_string(array_agg(cp.cpe_source_val),'') AS cpe_source_val_list, 1 AS version_flag, cp
385 FROM wt_cpe_process cp
386 WHERE cp.version_ranges = '(0,-)'-- <> '(0,-)'
387 GROUP BY cp.nvd_id , cp.unique_package, cp.package_name, cp.vendor, cp.vulnerable, cp.ecosystem
388 UNION ALL
389 SELECT cp2.nvd_id , cp2.unique_package, cp2.package_name, cp2.vendor, array_to_string(array_agg(DISTI
390            ,array_to_string(array_agg(cp2.cpe_source_val),'') AS cpe_source_val_list, 2 AS version_flag, cp
391 FROM wt_cpe_process cp2
392 WHERE cp2.version_ranges <> '(0,-)'
393 GROUP BY cp2.nvd_id , cp2.unique_package, cp2.package_name, cp2.vendor, cp2.vulnerable, cp2.ecosystem
394 )tmp
395 -- WHERE tmp.nvd_id = 'CVE-2019-12767'
396 GROUP BY tmp.nvd_id, tmp.unique_package, tmp.package_name, tmp.vendor , tmp.vulnerable, tmp.ecosystem
397 -- HAVING count(DISTINCT tmp.vulnerable) > 1

```

```

398 )
399 SELECT t1.nvd_id, t1.unique_package, t1.vendor, t1.package_name, t1.ecosystem, jsonb_build_object('true', t1.cv
400     t1.cpe_source_val_list, split_part(t1.unique_package, ':', 1) AS cpe_part , t1.version_flag_sum, CASE W
401 FROM (SELECT *FROM wt_cve_affect_unnest cau WHERE cau.vulnerable = 'true')t1
402 LEFT JOIN (SELECT *FROM wt_cve_affect_unnest cau2 WHERE cau2.vulnerable = 'false')t2
403     ON t1.nvd_id = t2.nvd_id
404     AND t1.unique_package = t2.unique_package
405 UNION ALL
406 SELECT t1.nvd_id, t1.unique_package, t1.vendor, t1.package_name, t1.ecosystem, jsonb_build_object('true', null
407     t1.cpe_source_val_list, split_part(t1.unique_package, ':', 1) AS cpe_part, NULL AS version_flag, '{}' A
408 FROM (SELECT *FROM wt_cve_affect_unnest cau WHERE cau.vulnerable = 'false')t1;
409
410
411 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
412 CREATE TABLE test.t_tmp_node_component AS
413 SELECT nextval('cve_graph_seq') AS seq, nvd_id , package_name AS component_name, vendor, package_name, ecosyste
414     NULL AS collectionURL, NULL AS defaultStatus,
415     split_part(unique_package, ':', 1) AS platform, jsonb_build_object('version_range', t.package_events ->> '
416     jsonb_build_object('version_range', t.package_events ->> 'false', 'version_list', t.packageversions ->>
417 FROM test.t_tmp_grahp_node_affected_msg t ;
418
419
420 SELECT *FROM test.t_tmp_node_component LIMIT 10
421
422 DELETE FROM test.dws_graph_node_affected_component WHERE vul_source = 'NVD';
423 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
424 INSERT INTO test.dws_graph_node_affected_component
425     SELECT nextval('cve_graph_seq') AS seq, tmp.*
426 FROM (
427     SELECT DISTINCT component_name , vendor, package_name, ecosystem, 'NVD' AS vul_source FROM test.t_tmp
428 )tmp;
429 bash gen_graph_data.sh "NVD" "node" "affected_component"
430 bash neo4j_affected_component_node_load.sh "NVD" "0"
431
432
433
434 DELETE FROM test.dws_graph_relationships_affected_components WHERE vul_source = 'NVD';
435 ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
436 INSERT INTO test.dws_graph_relationships_affected_components
437     SELECT nextval('cve_graph_seq') AS seq, tmp.*
438 FROM (
439     SELECT DISTINCT nvd_id, component_name, vendor, package_name, ecosystem, '' repo_url, '' AS platform,
440         '' AS collectionurl, '' AS defaultstatus, affect_msg, unaffected_msg, 'NVD' AS vul_source FROM t
441 )tmp;
442 bash gen_graph_data.sh "NVD" "relationships" "affected_components"
443 bash neo4j_relationships_affected_components.sh "NVD" "0"
444
445 --\copy (SELECT DISTINCT package_name AS component_name, vendor, package_name, ecosystem FROM test.t_tmp_node_c
446 --LOAD CSV WITH HEADERS FROM 'file:///affected_component_nodes.csv' AS row
447 --merge (n:affected_components {component_name: row.component_name, ecosystem: row.ecosystem, package_name:row.
448
449 --SELECT *FROM test.t_tmp_node_component LIMIT 100
450
451
452 --\copy (SELECT DISTINCT vendor FROM test.t_tmp_grahp_node_affected_msg) to 'vendor_nodes.csv' with (delimiter
453
454
455 --SELECT *FROM test.t_tmp_grahp_node_affected_msg LIMIT 10

```



```

456 --
457 --
458 --
459 --create index for (n:Package) on (n.unique_name);
460 --
461 --
462 --LOAD CSV WITH HEADERS FROM 'file:///vendor_nodes.csv' AS row
463 --CREATE (n:Vendor {name: row.vendor})
464 --create index for (n:Vendor) on (n.name);
465
466
467 --SELECT *FROM test.t_tmp_node_component LIMIT 10
468 --
469 --\copy (SELECT * FROM test.t_tmp_node_component WHERE seq <= 100000) to 'r_affected_10w_1.csv' with (delimi
470 ----\copy (SELECT DISTINCT unique_package AS r_start, vendor AS r_end FROM test.t_tmp_grahp_node_affected_msg
471 --
472 --LOAD CSV WITH HEADERS FROM 'file:///r_affected_10w_11.csv' AS row
473 --      MATCH (n_nvd:VuIn_NVD {id: row.nvd_id})
474 --      MATCH (np:affected_components {component_name: row.component_name, ecosystem: row.ecosystem, pac
475 --      MERGE (n_nvd)-[:AFFECTS {repo_url: COALESCE(row.repo_url,''), platform: row.platform, collecti
476

```