graph_osv

```
1
2  --node vul:
3  DROP TABLE IF EXISTS test.t_tmp_node_vul_osv;
4  ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
5  CREATE TABLE test.t_tmp_node_vul_osv AS
6  WITH wt_severity AS
7  (
8  SELECT tmp.ecosys || '_' || tmp.osv_id AS id , jsonb_agg(tmp.severity_val) AS severity
9  FROM (
10   SELECT
11 --          CASE  severity.value ->> 'type'  WHEN 'CVSS_V3'AS severity_type,
12 --         severity.value ->> 'type' AS severity_type,
13          jsonb_build_object(
14                          CASE split_part(severity.value ->> 'score', '/',1)
15                             WHEN 'CVSS:3.1' THEN 'cvssV3_1'
16                             WHEN 'CVSS:3.0' THEN 'cvssV3_0'
17                             WHEN 'CVSS:4.0' THEN 'cvssV4_0'
18                             ELSE 'cvssV2_0' END , jsonb_build_object('vectorString', severity.value ->> 'sc
19                                                                      'version', split_part( split_part(sev
20                                                                      'baseScore', NULL,
21                                                                      'attackVector', NULL,
22                                                                      'baseSeverity', NULL,
23                                                                      'confidentialityImpact', NULL,
24                                                                      'integrityImpact', NULL,
25                                                                      'availabilityImpact', NULL,
26                                                                      'userInteraction', NULL,
27                                                                      'attackComplexity', NULL,
28                                                                      'privilegesRequired',NULL,
29                                                                      'scope', null
30                                                                     ) ) AS severity_val,
31          oos.osv_id ,
32          oos.ecosys
33     FROM warehouse.ods_osv_source_df oos,
34     jsonb_array_elements(oos.osv_msg -> 'severity') severity )tmp
35  GROUP BY tmp.ecosys || '_' || tmp.osv_id
36  ) ,
37  wt_osv_credit AS
38 (
39     SELECT cre_ot.id, jsonb_build_object('identifier', array_agg( cre_ot.source_msg), 'discovery', null) AS sou
40     FROM
41     (
42         SELECT
43            DISTINCT  credit->>'name' AS source_msg, t.ecosys || '_' ||  t.osv_id AS id --, credit -> 'contact'
44         FROM
45             warehouse.ods_osv_source_df t,
46             jsonb_array_elements(t.osv_msg->'credits') AS credit
47         WHERE
```

```
            credit->>'type' = 'FINDER' OR
            NOT credit ? 'type'
    )cre_ot
    GROUP BY cre_ot.id
)
SELECT nextval('cve_graph_seq') AS seq, tmp.*, ws.severity, COALESCE (oc.source_msg_agg,'{}') AS source
FROM
(
    SELECT oos.ecosys || '_' || oos.osv_id AS id,
           COALESCE(oos.osv_msg ->> 'aliases','') AS aliases,
           jsonb_build_object('title', COALESCE(oos.osv_msg ->> 'summary',''), 'details',  oos.osv_msg ->> 'det
           jsonb_build_object('type', NULL, 'cweId', coalesce(COALESCE(oos.osv_msg -> 'database_specific' ->> '
           jsonb_build_object( 'published', oos.osv_msg ->> 'published',
                               'lastModified', oos.osv_msg ->> 'modified', 'datePublic', null) AS time_info,
--         oos.ecosys AS ecosys,
--         oos.osv_msg ->> 'schema_version' AS schema_version,
--         COALESCE (oos.osv_msg ->> 'database_specific','{}') AS database_specific ,
--         osv_msg AS msg
          ('{"impact_info":{"impacts":null, "impactScore":null},
            "solution_info":[],
            "exploit_info":{"exploitable":null, "exploits":null, "exploit_url":null, "exploitabilityScore":null
            "PoC_info":{"PoC_available":null, "PoC":null, "PoC_url":null},
            "patch_info":{"patch_available":null, "patch_url":null},
            "report_status":' || CASE WHEN (oos.osv_msg -> 'database_specific' ->> 'github_reviewed') = 'true'
    FROM warehouse.ods_osv_source_df oos
)tmp
LEFT JOIN wt_severity ws
      ON ws.id = tmp.id
LEFT JOIN wt_osv_credit oc
      ON oc.id = tmp.id ;

DELETE FROM test.dws_graph_node_vul WHERE vul_source = 'OSV';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_node_vul
SELECT nextval('cve_graph_seq') AS seq, tmp.*
FROM
(
SELECT  DISTINCT  id ,  aliases , source, description ,weakness, severity, time_info,  vul_status, 'OSV' AS vul
)tmp ;




--
----node:: SecurityNotice  create
--SELECT oos.osv_msg ->> 'id' AS osv_id, related.value ->> 0 AS related
--FROM warehouse.ods_osv_source_df oos,
--    jsonb_array_elements(oos.osv_msg -> 'related') related

--node CWE merge
DROP TABLE IF EXISTS test.t_tmp_node_cwe;
CREATE TABLE test.t_tmp_node_cwe AS
SELECT oos.ecosys || '_' || oos.osv_id AS id , cwe_ids.value ->> 0 AS cwe_id
FROM warehouse.ods_osv_source_df oos,
    jsonb_array_elements(oos.osv_msg -> 'database_specific' -> 'cwe_ids') cwe_ids;


DELETE FROM test.dws_graph_cwe_node WHERE vul_source = 'OSV';
```

```
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_cwe_node
SELECT nextval('cve_graph_seq') AS seq, tmp.*
FROM
(
SELECT  DISTINCT cwe_id , 'OSV' FROM  test.t_tmp_node_cwe
)tmp ;

\copy (SELECT DISTINCT cwe_id FROM test.t_tmp_node_cwe) to 'node_cwe_osv.csv' with  (delimiter ',', FORCE_QUOTE



DELETE FROM test.dws_graph_Relationships_cwe WHERE vul_source = 'OSV';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_Relationships_cwe
SELECT nextval('cve_graph_seq') AS seq, tmp.*
FROM
(
SELECT  DISTINCT t.id AS vul_id, t.cwe_id , '' AS cwe_type, '' AS cwe_desc, 'OSV' AS vul_source FROM  test.t_tm
)tmp ;

SELECT *FROM test.dws_graph_Relationships_cwe LIMIT 100

SELECT * FROM test.dws_graph_Relationships_cwe




LOAD CSV WITH HEADERS FROM 'file:///relationships_cwe_${vul_source}_${num}.csv' AS row
             MATCH (cve:Vuln_OSV {id: row.vul_id})
             MATCH (cwe:CWE {id: row.cwe_id})
             MERGE (cve)-[:HAS_CWE{cweId: row.cwe_id, type: row.cwe_type, description: row.cwe_desc}]->(cwe);

--node Reference merge

DROP TABLE IF EXISTS test.t_tmp_node_ref;
CREATE TABLE test.t_tmp_node_ref AS
SELECT oos.ecosys || '_' || oos.osv_id AS id, refs.value ->> 'type' AS ref_type, refs.value ->> 'url' AS ref_ur
FROM warehouse.ods_osv_source_df oos,
    jsonb_array_elements(oos.osv_msg -> 'references') refs  ;



[{"repo": "https://github.com/kanboard/kanboard", "type": "GIT", "events": [{"introduced": "0"}, {"fixed": "074

DELETE FROM test.dws_graph_cwe_node WHERE vul_source = 'OSV';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_refs_node
SELECT nextval('cve_graph_seq') AS seq, tmp.*
FROM
(
SELECT  DISTINCT ref_url , 'OSV' FROM  test.t_tmp_node_ref
)tmp ;

SELECT * FROM test.dws_graph_refs_node LIMIT 10

```

```
LOAD CSV WITH HEADERS FROM 'file:///node_.csv' AS row
MERGE (ref:references {url: row.url});

DELETE FROM test.dws_graph_relationships_refs WHERE vul_source = 'OSV';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_relationships_refs
SELECT nextval('cve_graph_seq') AS seq, tmp.*
FROM
(
SELECT  DISTINCT id AS vul_id , ref_url AS url, ref_type AS tag, '' AS description, 'OSV' AS vul_source FROM  t
)tmp ;

SELECT *FROM test.dws_graph_relationships_refs LIMIT 10


LOAD CSV WITH HEADERS FROM 'file:///ref_10w_9.csv' AS row
             MATCH (cve:Vuln_NVD {id: row.r_start})
             MATCH (ref:references {url: row.r_end})
             MERGE (cve)-[:HAS_REFERENCE{description: row.ref_source,tags: COALESCE(row.ref_tags, '')}]->(ref);


--node component:
    --if COALESCE(package_ecosys, package_name)

DROP TABLE IF EXISTS test.t_tmp_grahp_node_component_osv;
CREATE TABLE test.t_tmp_grahp_node_component_osv AS
WITH wt_osv_tmp as
(
SELECT oos.ecosys || '_' || oos.osv_id AS id,
           affecteds.value -> 'package' ->> 'ecosystem' AS package_ecosys, --vendor: *
           split_part(affecteds.value -> 'package' ->> 'ecosystem', ':', 1) AS  package_ecosys_group,
           affecteds.value -> 'package' ->> 'name' AS module_package, --AS component_name
           affecteds.value -> 'package' ->> 'purl' AS repo_purl,
           affecteds ->> 'ranges' AS affected_ranges,
           affecteds ->> 'versions' AS affected_versions, --,
           affecteds_ranges ->> 'type' AS affected_range_type,
           affecteds_ranges ->> 'repo' AS affected_range_repo,
--         affecteds_ranges.value ->> 'database_specific' AS affected_range_database_specific,
           affecteds_ranges ->> 'events' AS affected_range_events,
           oos.ecosys
    FROM warehouse.ods_osv_source_df oos,
        jsonb_array_elements(oos.osv_msg -> 'affected') affecteds ,
        jsonb_array_elements(affecteds.value -> 'ranges') affecteds_ranges
--   WHERE oos.ecosys || '_' || oos.osv_id = 'GIT_CVE-2020-36138'
)
,wt_osv_tmp1 AS
(
   SELECT wot.id, wot.package_ecosys_group AS package_ecosys,
          CASE WHEN wot.package_ecosys_group = 'Maven' THEN split_part(wot.module_package, ':', 1)
               WHEN wot.package_ecosys_group = 'Packagist' THEN split_part(wot.module_package, '/', 1)
               WHEN wot.package_ecosys_group = 'SwiftURL' THEN split_part(wot.module_package, '/', 2)
               ELSE '-'
          END AS vendor,
          '-' AS component,
          CASE WHEN wot.package_ecosys_group = 'Maven' THEN split_part(wot.module_package, ':', 2)
               WHEN wot.package_ecosys_group = 'Packagist' THEN split_part(wot.module_package, '/', 2)
               WHEN wot.package_ecosys_group = 'SwiftURL' THEN split_part(wot.module_package, '/', 3)
               --GO 语言的 module_package 对应的是 go 官网的id ，可以通过id获取到package 的名称
```

```
               ELSE wot.module_package
          END AS package,
          wot.module_package,
          wot.ecosys,
          wot.repo_purl, wot.affected_ranges, wot.affected_versions,  wot.affected_range_type,
          wot.affected_range_repo,
          split_part(wot.affected_range_repo, '/', 3) ~ '^git\.[a-zA-Z0-9_-]+\.org$' AS git_xxx_org_flag,
          wot.affected_range_events
   FROM wt_osv_tmp wot
)
SELECT tmp.*, COALESCE(em.target_name, tmp.ecosystem_tmp) AS ecosystem
FROM
(
SELECT  ot.id,  coalesce(ot.package_ecosys, '*')  AS ecosystem_tmp, --AS platform
        ot.affected_range_type,
        ot.vendor  AS vendor,
        ot.component,
       ot.package,
       ot.repo_purl  AS repo_url,
       '' AS defaultstatus, '' AS collectionURL, jsonb_build_object('version_range', ot.affected_range_events,
       jsonb_build_object('version_range', '[]', 'version_list', '[]') AS unaffected_versions
FROM wt_osv_tmp1 ot
WHERE ot.affected_range_type <> 'GIT'
UNION ALL
SELECT  ot.id,  'GIT'  AS ecosystem_tmp, --AS platform
        ot.affected_range_type,
       CASE WHEN ot.git_xxx_org_flag = TRUE  THEN substring( split_part(ot.affected_range_repo, '/', 3) FROM '^
            WHEN regexp_replace(split_part(ot.affected_range_repo,'/',5), '\.git$','') = '' THEN '-'
            ELSE  split_part(ot.affected_range_repo,'/',4)  END   AS vendor,
       ot.component,
       CASE WHEN ot.git_xxx_org_flag = TRUE  THEN regexp_replace( split_part(ot.affected_range_repo, '/', 4) ,
            WHEN  regexp_replace(split_part(ot.affected_range_repo,'/',5), '\.git$','') = '' THEN  ot.module_pa
            ELSE  regexp_replace(split_part(ot.affected_range_repo,'/',5), '\.git$','') END  AS package,
       ot.affected_range_repo  AS repo_url,
       '' AS defaultstatus, '' AS collectionURL, jsonb_build_object('version_range', ot.affected_range_events,
       jsonb_build_object('version_range', '[]', 'version_list', '[]') AS unaffected_versions
FROM wt_osv_tmp1 ot
WHERE ot.affected_range_type = 'GIT'
)tmp
LEFT JOIN test.dim_vul_ecosystem_map em
       ON em.vul_source = 'OSV'
       AND em.source_name = tmp.ecosystem_tmp;


UPDATE  test.t_tmp_grahp_node_component_osv t
SET ecosystem = 'GIT', package = 'curl',  vendor = 'curl', repo_url = 'https://github.com/curl/curl.git'
WHERE split_part(id, '_', 1)  <> ecosystem
AND t.ecosystem = '*';









SELECT *FROM test.t_tmp_grahp_node_component_osv t WHERE t.package = ''

```

```
SELECT DISTINCT ecosystem FROM test.t_tmp_grahp_node_component_osv t LIMIT 100

DELETE FROM test.dws_graph_node_affected_component WHERE vul_source = 'OSV';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_node_affected_component
SELECT nextval('cve_graph_seq') AS seq, tmp.*
FROM
(
SELECT  DISTINCT  component , vendor, package , ecosystem, 'OSV' AS vul_source FROM  test.t_tmp_grahp_node_comp
)tmp ;

SELECT * FROM test.dws_graph_node_affected_component  WHERE vul_source = 'OSV'  AND package_name IS NULL LIMIT



DELETE FROM test.dws_graph_relationships_affected_components WHERE vul_source = 'OSV';
ALTER SEQUENCE cve_graph_seq RESTART START WITH 1;
INSERT INTO test.dws_graph_relationships_affected_components
SELECT nextval('cve_graph_seq') AS seq, tmp.*
FROM
(
SELECT  DISTINCT id AS vul_id,   component , vendor, package , ecosystem,  repo_url AS repo_url,
        '' AS platform, COALESCE(collectionurl,'') AS collectionUrl, COALESCE(defaultstatus,'') AS defaultstatu
      'OSV' AS vul_source FROM  test.t_tmp_grahp_node_component_osv
)tmp ;



bash gen_graph_data.sh "OSV" "node" "affected_component"
bash neo4j_affected_component_node_load.sh "OSV" "0"

bash gen_graph_data.sh "OSV" "relationships" "affected_components"
bash neo4j_relationships_affected_components.sh "OSV" "2"

SELECT *FROM test.t_tmp_grahp_node_component_osv LIMIT 10


(seq int , vul_id varchar, component varchar, vendor varchar, package_name varchar, ecosystem varchar, repo_url
INSERT INTO test.dws_graph_relationships_affected_components

SELECT *FROM test.dws_graph_relationships_affected_components WHERE defaultstatus IS NULL

Alpine_CVE-2006-20001
id: Alpine_CVE-2006-20001

SELECT *FROM test.dws_graph_relationships_affected_components t
WHERE t.vul_id = 'Alpine_CVE-2006-20001'


SELECT *FROM test.dws_graph_relationships_affected_components WHERE vul_id = 'Alpine_CVE-2006-20001'


SELECT *FROM test.dws_graph_node_vul dgvn WHERE dgvn.id  = 'Alpine_CVE-2006-20001'
```