1. **Steps to Install Nexpose Community eddition.**

**Step-1:** Download the Nexpose Community Edition, you will receive a mail containing Product Key:  
 Click here to download: <https://www.rapid7.com/info/nexpose-community/>

**Step-2:** As mentioned below you can see I have transfer the downloaded file(Rapid7Setup-Linux64.bin) to a instance & changed the permission to executable by using command:

**chmod +x <nexpose-fileName.bin>**

**Text

Description automatically generated**

**Step-3:** Run this command to execute: **./Rapid7Setup-Linux64.bin**

**Text

Description automatically generated**

**Step-4: After that f**ollow the instructions along the snips as shown below:

Graphical user interface, text

Description automatically generated

Text

Description automatically generated

When asked for Database port number, hit **Enter** to continue (default port is 5432). If needed enter the port number as per your requirements.

Text

Description automatically generated

Enter the information in displayed fields (“First name”, “Last name”, “Company”) as per your requirements, as shown in below snip:

Text

Description automatically generated

As mentioned in below snip enter a username and password (Note: these credentials are for the Nexpose UI)

Text

Description automatically generated

Text

Description automatically generated

As mention in above snip enter “y” if you want to initialize and start the nexpose after installation and enter “n” to do it manually later.

Text

Description automatically generated

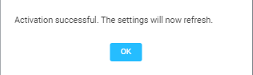
Now, the installation is finished.

**Note:** open the ports mentioned in above snip (i.e., 3780), which is Nexpose GUI default port.

**Step-5:** After Successful Installation, open the site in browser (eg:https://<server\_ip>:3780) and enter the Product Key which you have received on your mail in Step-1.

Graphical user interface, text, application

Description automatically generated



Graphical user interface, application, Teams

Description automatically generated

1. **Sending logs to Splunk**

To Get the data in Splunk, we need to export Nexpose Data to an external Database “postgress”. Hence, Install Postgress on another Instance. Do not install Postgress where Nexpose is installed because Nexpose itself uses postgress for its purpose.

After Installing Postgress run the below commands to create a database and give right permission:

*CREATE DATABASE nexposeCE;*

*create user nexposeCE with encrypted password <password>';*

*ALTER USER nexposeCE WITH SUPERUSER;*

*ALTER DATABASE nexposece OWNER TO nexposece;*

*GRANT ALL PRIVILEGES ON DATABASE nexposece TO nexposece;*

*GRANT CONNECT ON DATABASE nexposece TO nexposece;*

Useful Postgree Commands:

*Psql -> to enter postgress*

*\l - > to list database*

*\h -> help with sql cmds*

*\q -> To quit*

*\dt --> to list tables*

*\du -> list of roles*

**Note:**  We need to add some additional configuration in Postgress so that the communication link with Nexpose can be established.

*Edit the* ***pb\_hba.conf*** *to add the host as shown below:*

*Graphical user interface

Description automatically generated with low confidence*

*And after that edit the* ***postgresql.conf*** *as shown below:*

*A screenshot of a computer

Description automatically generated with medium confidence*

Now Install Splunk DB Connect on Splunk Heavy Forwarder and create connection between postgress as shown below:

**Graphical user interface, text, application

Description automatically generated**

After that create inputs:

Graphical user interface, application

Description automatically generated

As Shown in the above images, three different inputs are created for Asset and vulnerability data and the queries we have used are mentioned below, you can modify or create your own query as per your requirement.

**NexposeAsset:**

**A screenshot of a computer

Description automatically generated**

*SELECT da.last\_assessed\_for\_vulnerabilities,*

*dsa.asset\_id as asset\_id, dsa.site\_id as site\_id,*

*ds.name as site\_name, da.mac\_address, da.ip\_address as ip, da.host\_name as hostname,da.os\_description as description,*

*da.os\_vendor as vendor\_product, da.host\_type,da.os\_architecture as family,da.os\_version as version,da.os\_system as os, fa.vulnerabilities, fa.critical\_vulnerabilities, fa.severe\_vulnerabilities, fa.moderate\_vulnerabilities, fa.malware\_kits, fa.exploits, fa.vulnerability\_instances, fa.risk\_score as riskscore, fa.pci\_status,das.protocols,das.services,dagc.asset\_group\_accounts,*

*dasoft.installed\_software*

*from dim\_site\_asset dsa*

*LEFT JOIN (select da.asset\_id,da.os\_architecture, da.os\_version, da.os\_system, da.os\_description, da.ip\_address, da.mac\_address, da.host\_name, da.os\_vendor, da.host\_type, da.last\_assessed\_for\_vulnerabilities FROM dim\_asset da) da USING (asset\_id)*

*LEFT JOIN (select ds.site\_id, ds.name FROM dim\_site ds) ds using (site\_id)*

*LEFT JOIN (select fa.asset\_id, fa.vulnerabilities, fa.critical\_vulnerabilities, fa.severe\_vulnerabilities, fa.moderate\_vulnerabilities, fa.malware\_kits, fa.exploits, fa.vulnerability\_instances, fa.risk\_score, fa.pci\_status FROM fact\_asset fa) fa USING (asset\_id)*

*LEFT JOIN (select das.asset\_id, (string\_agg(DISTINCT das.name, ';')) as services, (string\_agg(DISTINCT das.protocol, ';')) as protocols FROM dim\_asset\_service das GROUP BY das.asset\_id) das using (asset\_id)*

*LEFT JOIN (select dagc.asset\_id, (string\_agg(DISTINCT dagc.name,';')) as asset\_group\_accounts FROM dim\_asset\_group\_account dagc GROUP BY dagc.asset\_id) dagc USING (asset\_id)*

*LEFT JOIN (select dasoft.asset\_id, (string\_agg(DISTINCT dasoft.name,';')) as installed\_software FROM dim\_asset\_software dasoft GROUP BY dasoft.asset\_id) dasoft USING (asset\_id)*

**NexposeVulnerability**:

SELECT timestamp,asset\_id, site\_id,favf.vulnerability\_instances,favf.vulnerability\_id as signature\_id,

favf.risk\_score,dv.title as signature ,dv.severity,dv.cvss\_score as cvss,dv.cvss\_vector,

dv.date\_added,dv.date\_published,dvc.categories as category,dve.skill\_level,

dvr.other\_references,da.ip\_address as dest,da.mac\_address as mac,da.host\_name,fv.first\_discovered,fv.most\_recently\_discovered,solution\_summary,solution\_count,solution\_types

FROM dim\_site\_asset

RIGHT JOIN (SELECT favf.date AS timestamp,favf.asset\_id ,favf.vulnerability\_instances ,

favf.vulnerability\_id ,favf.risk\_score

FROM fact\_asset\_vulnerability\_finding as favf) as favf USING(asset\_id)

LEFT JOIN (SELECT fv.vulnerability\_id,fv.first\_discovered,fv.most\_recently\_discovered

FROM fact\_vulnerability as fv) as fv USING(vulnerability\_id)

LEFT JOIN (SELECT ds.site\_id,ds.name, ds.scan\_template,ds.scan\_engine

FROM dim\_site as ds) as ds USING(site\_id)

LEFT JOIN (SELECT dv.vulnerability\_id ,dv.title ,dv.severity, dv.cvss\_score, dv.cvss\_vector,

dv.date\_added, dv.date\_published

FROM dim\_vulnerability as dv) as dv USING(vulnerability\_id)

LEFT JOIN (SELECT dvc.vulnerability\_id,(string\_agg(DISTINCT dvc.category\_name,';')) AS categories

FROM dim\_vulnerability\_category dvc

GROUP BY dvc.vulnerability\_id) dvc

USING (vulnerability\_id)

LEFT JOIN (SELECT dve.vulnerability\_id, dve.skill\_level AS skill\_level

FROM dim\_vulnerability\_exploit dve) dve

USING (vulnerability\_id)

LEFT JOIN (SELECT dvr.vulnerability\_id,

(string\_agg(DISTINCT dvr.source || ';' || dvr.reference,';')) AS other\_references

FROM dim\_vulnerability\_reference dvr

GROUP BY dvr.vulnerability\_id) dvr

USING (vulnerability\_id)

LEFT JOIN (SELECT da.asset\_id, da.ip\_address,da.mac\_address, da.host\_name

FROM dim\_asset da) da

USING (asset\_id)

LEFT JOIN (SELECT vulnerability\_id,(array\_agg(summary))[1] AS solution\_summary,

COUNT(solution\_id) AS solution\_count,

string\_agg(distinct(solution\_type),

'|') AS solution\_types

FROM dim\_vulnerability\_solution

JOIN (SELECT solution\_id,solution\_type,summary

FROM dim\_solution) dsol USING (solution\_id) GROUP BY vulnerability\_id) dsv

USING (vulnerability\_id)

*A screenshot of a computer

Description automatically generated*

***NexposeScan:***

***A screenshot of a computer

Description automatically generated***

*SELECT scan\_id,site\_id,started,finished, status, type,ds.scan\_template, ds.scan\_engine*

*FROM dim\_scan*

*LEFT JOIN (SELECT ds.site\_id,ds.name,ds.scan\_template,ds.scan\_engine*

*FROM dim\_site ds)ds USING(site\_id)*