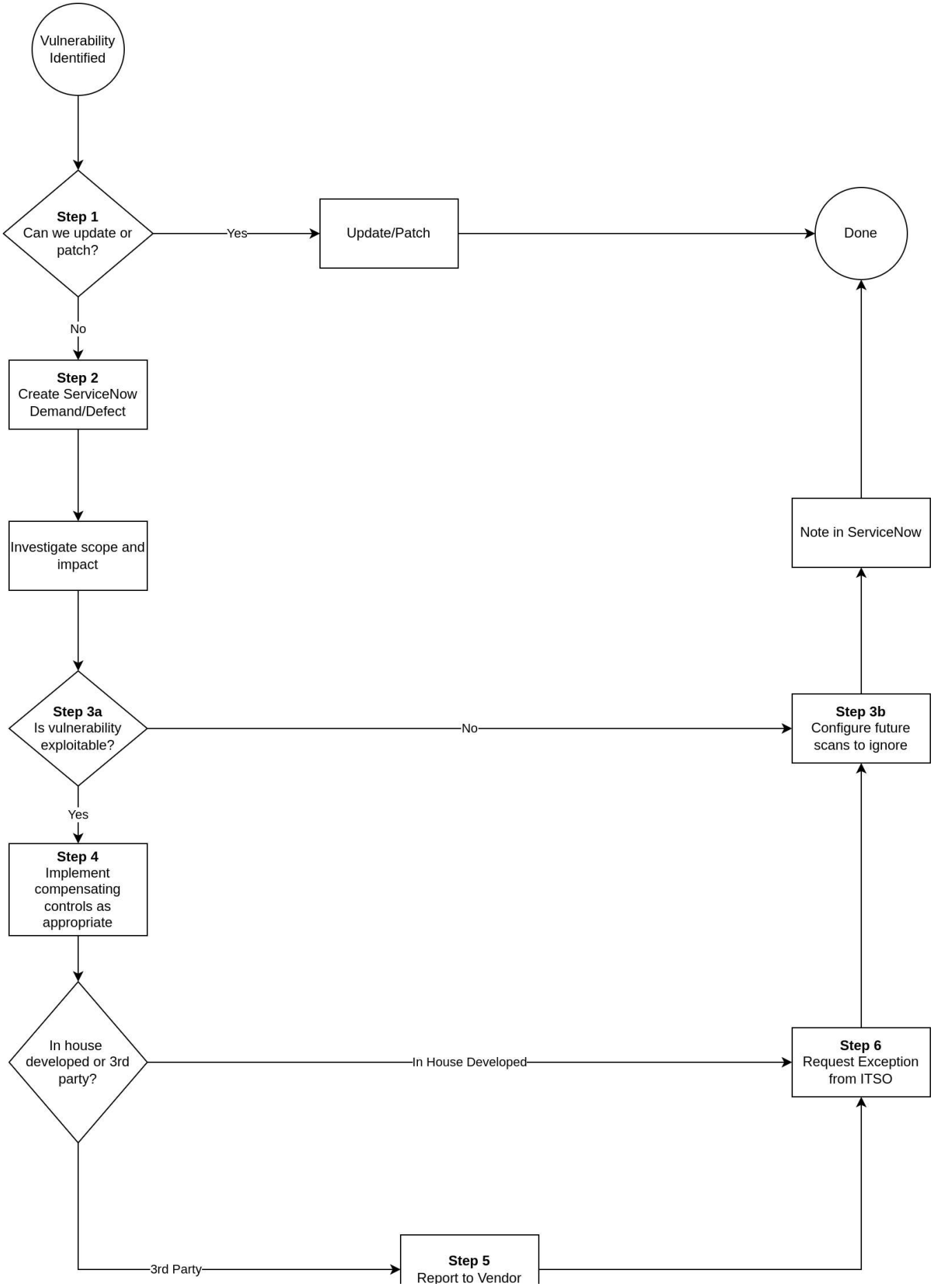
Vulnerability Identified
Step 1
Can we update or patch?
Yes
Update/Patch
Done
No
Step 2
Create ServiceNow Demand/Defect
Investigate scope and impact
Step 3a
Is vulnerability exploitable?
No
Step 3b
Configure future scans to ignore
Note in ServiceNow
Yes
Step 4
Implement compensating controls as appropriate
In house developed or 3rd party?
In House Developed
Step 6
Request Exception from ITSO
3rd Party
Step 5
Report to Vendor

Step 1: Update artifacts to non-vulnerable versions

Where possible, update the vulnerable artifacts to later, patched versions. This course of action is generally only available for applications that we have developed and even then there are times that patched versions have not yet been released or have other problems that make them impossible to use. Since this document concerns vulnerability scanning that takes place at build time, it is assumed that the appropriate Service Now records for active development work (Enhancement, Project, etc) already exist and remediation that takes place at this step can be tracked there.

If it is a vendor application, we are unlikely to have control over which artifacts are included but we should ensure that we are on the latest version of the vendor product if possible.

Step 2: Track the Defect in Service Now

Reaching Step 2 means that your application is going to include vulnerable artifacts. This is an undesirable, but often unavoidable, state that should be tracked in Service Now. Create a Demand (Category: Operational, Type: Defect) and get it approved by the appropriate Demand Manager. Once approved, create the downstream Defect that will detail the overall progress and remediation of the vulnerable application. Since each individual vulnerability may be remediated at different times and in different ways (as detailed in Steps 3-5), hang a Story record off of the Defect for each vulnerability and work them until Complete. As long as the vulnerability remains, the Story should stay open.

Step 3: Exclude non-exploitable vulnerabilities

Not all vulnerabilities are created equal. Some vulnerabilities require attacks that cannot be replicated in our environment. Investigate the CVE and if you conclude that the vulnerability is not exploitable in our environment, that specific vulnerability should be added to a file in the gitlab repo referenced by the gitlab-ci variable TRIVY_IGNORE_FILE (e.g. `TRIVY_IGNORE_FILE: ./trivyignore`) in `.gitlab-ci.yml`. Each entry should be accompanied by a comment detailing the date, participants, Story, and results of the investigation. Once added to the ignore file, update the associated Story but leave the Story open until the vulnerability has been truly patched.

TRIVY_IGNORE_FILE example

```
# 2022-12-25 (lcovey) - This application is not vulnerable to made up CVE's. Reference STRY0001234 in SN.
CVE-1971-1234
# 2022-08-02 (rquintin) - We deploy on tomcat, so jetty vulnerabilities are not applicable. Reference STRY0006789 in SN.
CVE-2022-0001
```

Step 4: Implement Compensating Controls

In some cases, the risk identified in the previous analysis is great enough to require compensating controls. A couple of examples of these controls would be limiting access to the web application to the VPN or specific IP addresses, or adding additional rules to the Web Application Firewall. Once these controls are in place, the CVE may be added to the TRIVY_IGNORE_FILE but the associate Story should remain open until the vulnerability is truly patched.

Step 5: Inform the vendor and work with them to remediate

If we are dealing with a vendor application, we will likely be unable to freely swap out the vulnerable artifacts and will need to work with them to obtain a remediated version of their product. Contact the vendor through their preferred support channels to let them know that what we have discovered. If they have a workaround, implement it and mark the appropriate Story as Complete. If there is no workaround, request an estimated delivery date and record that in the Story but do not mark the Story as complete since the vulnerability remains.

Step 6: Inform ITSO of remaining vulnerabilities and either implement a compensating control or request an exception

In order to reach Step 5, an application has to have a known, exploitable vulnerability that we cannot remediate. Inform ITSO and work with them to either implement a compensating control or to request an exception via Requestable Item in ServiceNow. Record the control or exception in the Story but do not mark the Story as complete since the vulnerability remains.

Step 7: Close the Defect and Demand

Once all of the Stories on a Defect are closed (i.e. no exploitable vulnerabilities remain), mark the Defect as Closed and its associated Demand as Complete.