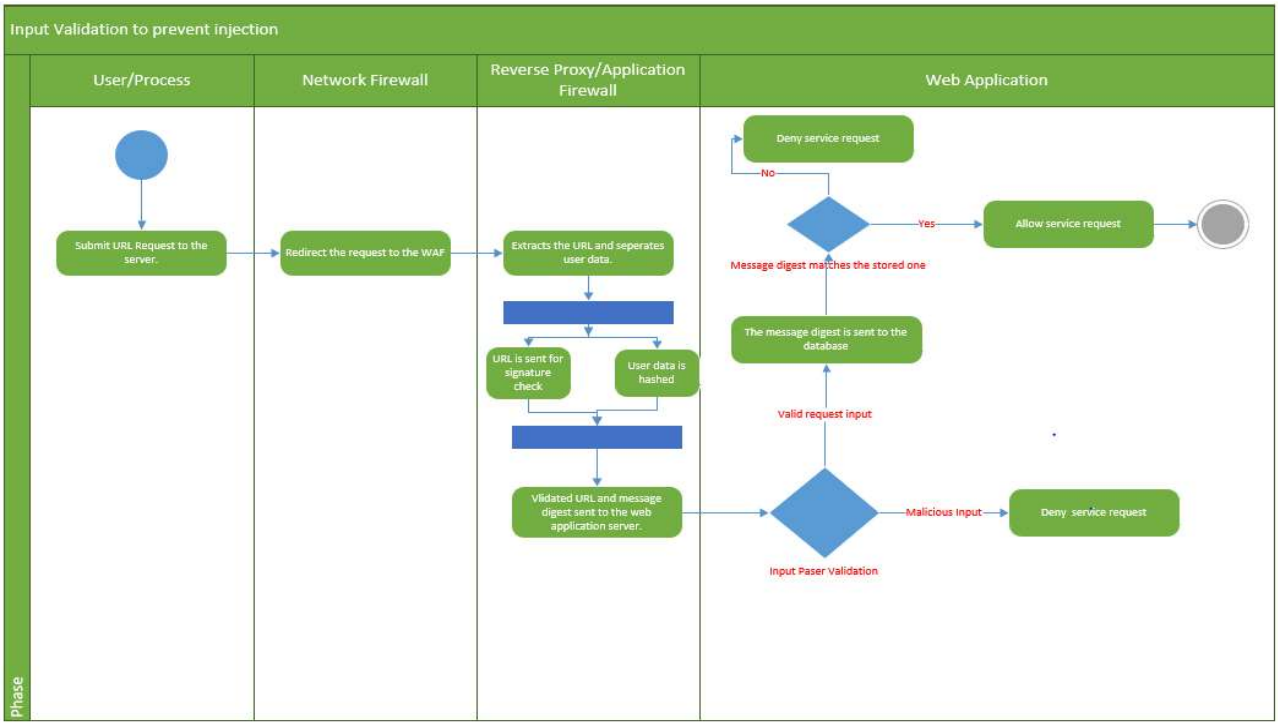
Structured Query Language (SQL) injections rank among the top ten (10) security threats based on the OWASP Top 10 vulnerability report. SQL injection is the technique used to inject malicious code into existing SQL statements to execute such code on the database server (Amin et al., 2018).

Unified Modelling Language(UML) consists of two diagram types, structural and behavioural diagrams. One of the behavioural diagrams is the activity diagram, which helps capture detailed activities and describing synchronisation and concurrency between activities. It focuses on the flow of activities involved in a single process and captures how activities depend on one another. In the case of the SQL injection, depending on how the attacker exercises the vulnerability, you can have inbound injection when you have the attack and exploit in the same channel or out-of-band injection. An activity diagram can assist in capturing the flow or sequence of activities in each case by decomposition of activities involved in a single process (Syriani and Ergin, 2012).

Partitions can help investigate responsibilities for interactions and associations between objects and actors. Swimlanes often clarify steps and responsibilities for sub-processes of the business process, which is essential in identifying a vulnerability within the system. Exposure such as failure to validate data input before submitting it to the database server is identifiable to where it may occur in the process.

The activity diagram is suitable for modelling injections compared to other behavioural diagrams because it captures the control flow between activities. It also captures activities that are made of more minor actions.



References :

Amin, M. *et al.* (2018) ‘Review of SQL Injection : Problems and Prevention’, *International Journal on Informatics Visualization*, 2, pp. 215–219.

Syriani, E. and Ergin, H. (2012) ‘Operational semantics of UML activity diagram: An application in project management’, *2012 2nd IEEE International Workshop on Model-Driven Requirements Engineering, MoDRE 2012 - Proceedings*, pp. 1–8. doi: 10.1109/MoDRE.2012.6360083.