# SymphonyAI Summit - SDLC Process Document

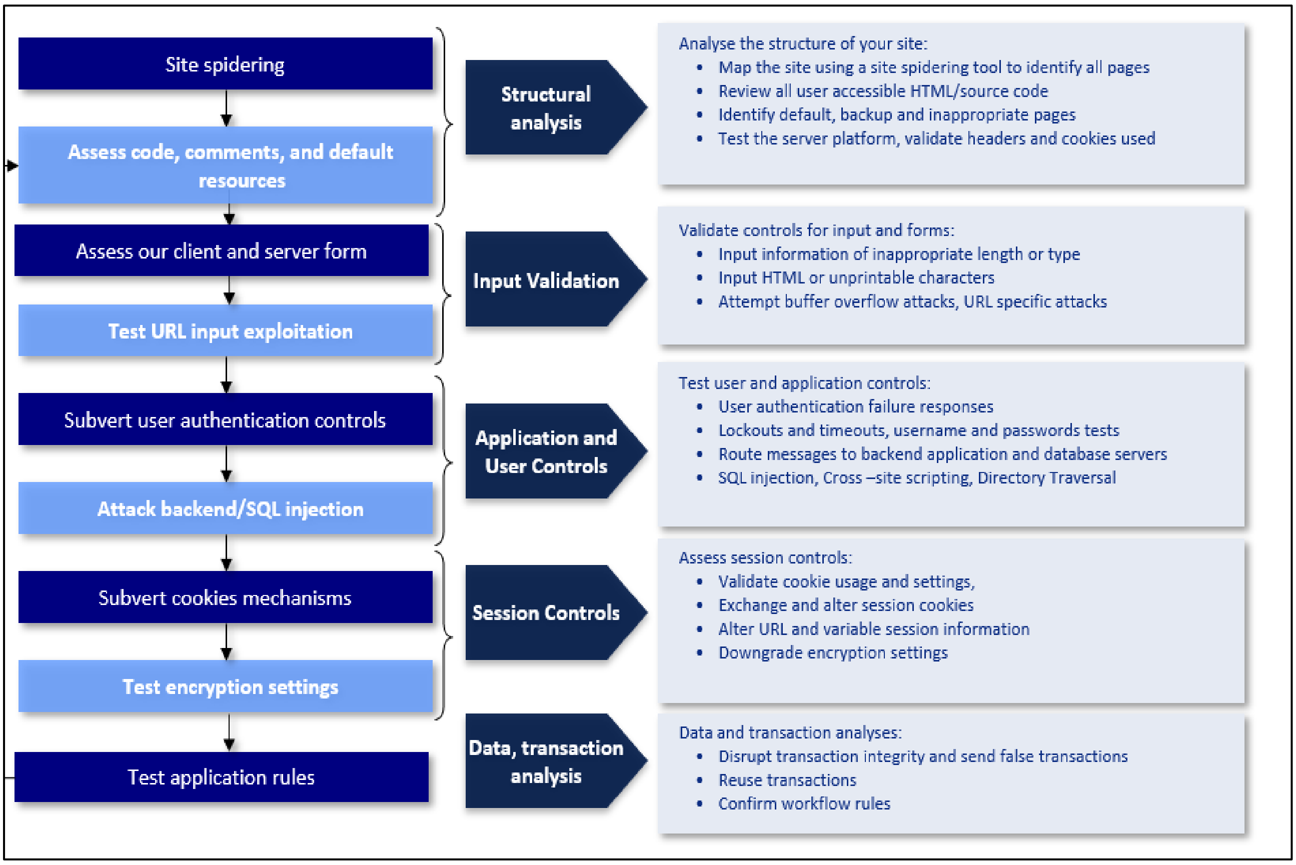
**Published On: 20th July 2020**

*SymphonyAI Summit Documentation Team*

# Penetration Test Process

4NQPO4VNNU Application is also tested for Security Vulnerability by a Lea ding Security Consultant. The Process for the same is depicted below.

## WEB APPLICATION TESTING



# Mobile APPLICATION TESTING

## Server-side Security

The server-side security testing is carried out using one of the approaches described in the application security assessment methodology.

## Client-side Security

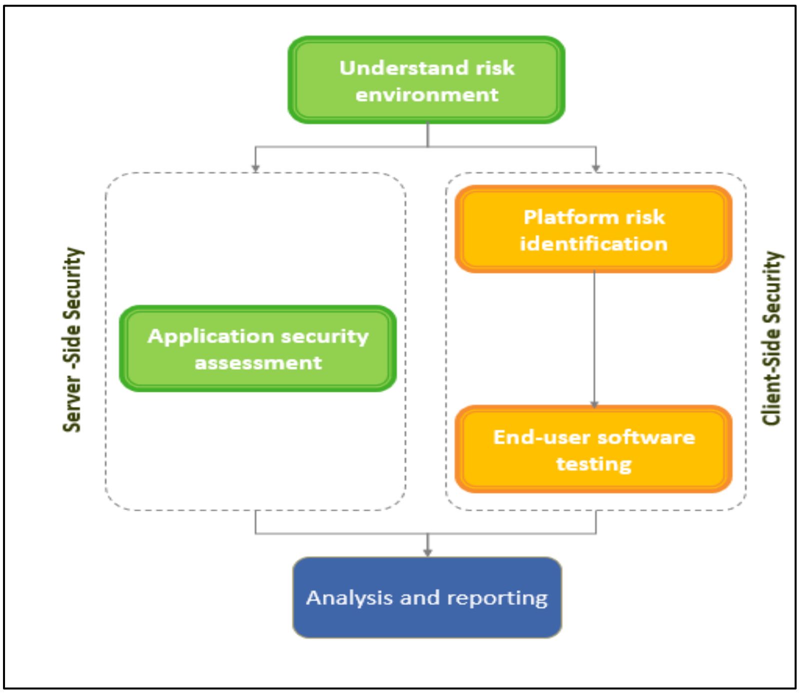
The client application is tested either using a platform emulator typically provided together with SDK and/or actual hardware device.

- **Platform Risk Identification**

  Functionality of the client application is thoroughly analysed to identify assumptions about platforms of executions that may not be always true, for example:
  - An application relies on GPS data being accurate, then such data may be spoofed if the application is executed on an emulator;
    - Storage and exchange of cryptographic keys or shared secrets between application and a security device such as SIM card cannot be intercepted by other applications;

- **End-user Software Testing**

  The data exchange between client-side application and server-side application is intercepted using various tools and the client-side application is being supplied with invalid responses to trigger erroneous behaviour. Fuzzing tools are used where possible to cover the maximum attack surface followed by manual investigation of suspicious behaviour.



# Documentation Management

The Documentation Development Life Cycle begins with a Planning Meeting. In the Planning Meeting the Tech Writers get to know about the features that are being planned for the release. Based on the features that are planned for the release, Tech Doc tasks, related to the User Stories, are created in the Team Foundation Server (TFS). The Knowledge Transfer (KT) sessions, for features planned in each Iteration, are conducted by the Development Teams. For the current Iteration, the Tech Writers document the features based on the KT received from the Development Teams. The draft version of the content created by the Tech Writers are shared to the Development and Quality Assurance teams for Technical Review. After the content is approved by the Development and QA teams, the technically reviewed content is shared for Peer Review amongst the writers in the Technical Documentation team. After the Peer Review, the content is finally shared for the Editorial Review. After the Editorial Review, the content is published, and the respective Tech Doc tasks are updated in the TFS. After updating the tasks, the writers close their respective Tech Doc tasks on completion of the documentation activities. The above process is then repeated in each iteration.

# Release Management

The release Management starts from the point where all the Features scoped for a release are reviewed, Tested and Documented. The release document is reviewed before publishing it to the released version of the application. The Build Engineer builds the final tested version of the application with Installation and Proxy guides. The same information collated with release Document is then sent out as a communication to all interested parties and Stake Holders. The version stands released.

# HF Process and Straetegy

SymphonyAI Summit has a periodic Hotfixes Releases for the customer reported issues, after the release of a Main Version and/or upgradation to the Staging environment. There is a workflow for the same and the below points are critical to adherence to the HF process and Strategy for smooth collaboration between HF process and Core Development Process:

- *Support and Implementation team* at LEVEL 1 handles all the customer reported issues. This team analyzes the issue and understands the configurations made in the customer environment to ensure the system is configured as it should be. After all the analysis if the issue still exists, a work order is raised in the portal and pushed to the Quality Team to verify and re-confirm.
- *Quality team* tries to reproduce the issue. If the issue is reproduced then the Quality team raises a workorder in the portal and is aligned, according to the priority, for the next upcoming Hot fix Release and follows the Kanban method of going through the whole implementation process.

Now, below are the key Points which are modified based on the inputs from different teams and to ensure, the Hotfix process flows seamlessly with the ongoing Sprint Activities for a particular release.

- It was discussed and Confirmed that, the list of issues is pulled from the portal, every 20th of a particular month, and those issues are aligned to be released in subsequent Month end.
- The list is shared with the interested parties (Here mainly the Support and Implementation Team), to make sure, the Issues which are to be taken up are in align with the expectation in

terms of what should be taken and what should not. This step is very important to ensure, that no P1 and P2 are missed after Development team starts working on it.

- The issues which are taken up, will be fixed in the latest branch of the Customer version and the latest branch. For example:
    - **Issues from v5.5 and below, v57sp+ will be fixed in v57sp5hf04, HF05, HF06 and So On.**
    - **V56+ will be fixed in v56 sp2 hf14, HF15 and so on.**
    - **V5.8 ALPS+ will be fixed in ALPS Sp1 HF01, HF02 and So on.**
- It is suggested to create one Work Order for a reported issue instead of multiple Work Orders. This suffices issues related to updating and closure of multiple Work Orders by Multiple teams in the portal.
- In terms of the communication channel for the teams to be on Same page, make use of a common Platform. The said platform is in Discussion Phase.  OR have
- Post starting on the agreed upon issues, if there are any P1 or P2 (Decision need to be taken after discussing with Management). The decision is based on the Criticality of the issue to the OPERATIONAL factor for the customer. This decision will also take counter of the type of Customer.

Post Development and QA cycle the released items can be in TFS for a particular release.