Automation Testing for VAMS and CAPS Applications of NOKIA

Dr. Siddaraju[1], Dr Nandini N[2], Ms. Shwetha S[3]

[1] Professor, Department of Computer Science and Engg, Dr.AIT, Bangalore,India.
[2] Associate Professor, Department of Computer Science and Engg, Dr.AIT, Bangalore,India.
[3] PG Student, Department of Computer Science and Engg, Dr.AIT, Bangalore,India.

Abstract:

Automation testing plays an important role in the software development process because of the growing complexity of applications. It improves software quality and reduces project costs. Though automation testing brings many benefits, it cannot totally replace manual testing. Manual and automation testing need to be performed in parallel. A challenge of the project and of automated tests is that they are vulnerable to the change of the tested software. An update to the application may cause an unexpected testing result if the structure of the tests is not well organized. So, maintenance will become an issue when the number of test cases increases tremendously. Many automation tools have been created and more new and improved libraries for Robot Framework will be re-leased to meet the needs of the automation software testing community. With the Tool “Robot Framework” I automated the whole CAPS and VAMS process, so it has reduced the time of manual testing. It covers the regression testing, where first test case should not fail after adding up the new testcase.

1.Introduction

1.1VAMS (Vulnerability Assessment and Management System)

VAMS is an application which enables the easy tracking and reporting of vulnerabilities in Nokia products. It does this by integrating vulnerability information from various sources. The VAMS application gets vulnerability inputs from information generated by commercial scanners run in Nokia labs, Vulnerabilities detected during penetration testing in Nokia labs, Latest available security patches information from Siemens CERT. Security experts from R&D input additional information about the applicability of the vulnerabilities and, as needed, resolution plans. This is called Risk Assessment.

This then enables the generation of reports.

Vulnerability management in VAMS interacts with the software change management tools. New vendor patches are linked to tracking IDs in the change management system. VAMS supports the generation of several reports, such as customer security assessment reports, internal patch, and vulnerability reports, etc.

1.2 CAPS (Compliance, Audit and Privacy System)

CAPS is Nokia's system for managing R&D and Services activities required to meet NSA, China CSL, GDPR and customer compliance. CAPS function is grouped as R&D Compliance, Service Compliance, Access Control Compliance. It will be used to generate the new product notification prior to the commercial deployment of a new product, to generate the product testing evidence for a release, to generate CALEA report for product if applicable, to generate the incident report for security vulnerabilities in product and also to provide the scrambling capability for Nokia product logs to meet NSA and customer requirements

2. Preliminaries

VAMS and CAPS functionalities are automated using the Robot Framework automation tool.

2.1.1 Need For VAMS

Vulnerability Assessment and Management

System mainly used to fix the vulnerabilities uploaded by the customer to the VAMS, the creation od security assessment reports for customers and government bodies was an error-prone and time-consuming task. VAMS automates much of the work to ensure more accurate and timely reports.

The key capabilities implemented by VAMS are to support the inputs from all commonly used commercial scanners, to maximizes the efficiency of R&D assessment of Vulnerabilities, in that it will automatically recognizes when the same vulnerability has been encountered before and reuses the assessment previously entered. In case where one assessment applies to multiple vulnerabilities, convenient copy/paste and bulk assessment functionalities are available. VAMS application is capable of automatic and periodic synchronization with change management system tracks fixes, and to provide consistent report format.

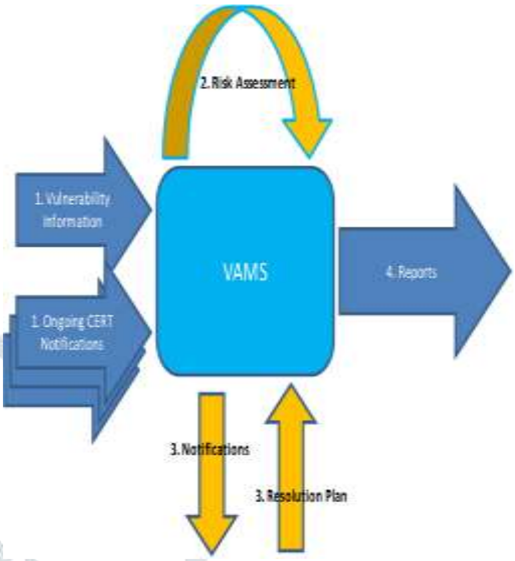
2.1.2 VAMS server access

Access is via SSO (there is no login page). Anyone with @Nokia.com email can access VAMS but will be provided by limited information. The user needs to request and be assigned a role in VAMS to fully access the benefits of data available in VAMS. As soon as we login in the link Request Access takes the user to the security page, which also has detailed instructions to request access. The product security managers and leads are provided as a reference if the user needs to reach out to them for approval. The user role must be approved by the product security lead at the product or release level. Other roles will be approved by the VAMS admin. The user roles are listed in the section VAMS User Roles. Google Chrome is the preferred browser for accessing VAMS (Firefox (windows) has been extensively tested as well and may be used). Microsoft IE Browser is not currently supported.

2.1.3 VAMS High Level Workflow

Vulnerability information will be given as input the VAMS application which will be uploaded by the customer using Scan upload feature of VAMS, vulnerability can also be ongoing CERT notifications.

In Risk Assessment there will be different modes of assessments based on the Vulnerability expert severity will be going to select Disposition and Remediation Type. Once the vulnerability is fixed corresponding report will be generated.



2.1.4 VAMS User Roles

Each VAMS user must be assigned a role when given access to VAMS. The assigned role defines the functions of the application that the user will be allowed to access.

Non-R&D Roles:

- Customer Security Lead: This role has access to Customer specific SLA.
- Customer Scan Support: This role is reserved for teams that need to upload customer scans and generate customer scan reports.
- Customer Support: The Customer Support role is reserved for teams that generate customer reports for select customers with contractual obligations.
- Quality Lead: A Quality Lead has read only access to VAMS vulnerability data and can generate reports.

- PSIRT: The PSIRT role is reserved for the PSIRT organization and has read access to all data across all products as part of their responsibility as a product security organization facing the customer. This role is reserved for the HSSE office and is authorized by VAMS admin.
- Product Security Manager: A Product Security Manager has read access to all data related to their product. The Product Security Manager is a member of the HSSE (security office) and not part of the R&D organization. This role is authorized during Product configuration and cannot be requested in VAMS.
- Security Auditor: This role has read access to VAMS vulnerability data, may upload customer scans, and generate reports.

R&D Roles:

- API-CI/CD Dashboard, API-CI/CD Pipeline and API-Read-only: These roles are used for CI/CD Pipeline access to VAMS.
- BG Escalation Lead: like report user role, but may access RAW SLA Data authorized by VAMS admin.
- BG Security Lead: Like Security Expert role, but at the BG level authorized by VAMS Admin.
- Product Manager: A Product Manager has read access to VAMS vulnerability data per authorized product along with the ability to upload scanner reports into VAMS, and to access and generate reports
- Product Security Lead: A Product Security Lead has access to all data related to their product, including assigning new users within their product. They are also responsible for configuring their product in VAMS such that vulnerability management can be done for that product. A Product Security Lead is a single point of contact for the product. VAMS allows for two PSLs (a PSL and an alternate) per product.
- Security Expert: A Security Expert has read/write access to VAMS vulnerability data per authorized product along with the ability to upload scanner reports into VAMS, and to access and generate reports. This is the primary role for R&D security users who are performing vulnerability risk assessment.
- Test Engineer: A Test Engineer has read access to VAMS vulnerability data per authorized product along with the ability to upload scanner reports into VAMS for Customer “Nokia R&D”, and to access and generate reports.

2.1.5 Features of VAMS

VAMS introduces a new VAMS Platform, new GUL pages and simplified procedures.

Framework

- Scalable architecture with flexibility to support multiple instances and disaster recovery
- Cloud based storage of Scan Files via Nokia S3 and use of Nokia EC2 computing cloud
- Django web framework with reuse of code with CAPS
- Angular Bootstrap with Material Design to present a Modern UI look and feel
- Single Sign On (SSO) support from day one (no login page)
- Increased security with backend validation of all data requests on both API and resource levels

Product Hierarchy

- Two level hierarchy: Product->Release

- Components are registered to a Release. Each Release can have the same or different components.
- Components can be optionally associated with a Team to group like components together.
- Configuration of components for contained Product/Releases is no longer necessary.

Dashboard

- Dashboard is the current default page which now introduces charts.
- Dashboard Vulnerability Details Table is always shown collapsed.
- Dashboard reuses Save Search Profiles from the Search page.
- Saved Search displayed content and/or Dates range can be altered dynamically.

Search Vulnerabilities

- The Search filters are optional (if not selected all authorized products are included in request).
- Date Range filter defaults to distribution date (same date SLA measurements are measured).
- Search criteria can be saved for later recall (after initial search) as Saved Search.
- Additional filters for further refinement of results, presented as Advanced Search, can be expanded or collapsed on the same page.
- The Saved Search list is reused by the Dashboard to define context.
- Search Results are presented per unique Vulnerability instance (this is the primary vulnerability).
- Specific Release Vulnerabilities are visible by expanding the primary vulnerability.
- User can hover over displayed data to view entire content.
- Search, Search Results, and Risk Assessment operations are all on the same page.

Risk Assessment

- Three modes of Risk Assessments are:
- Single Assessment follows the standard assessment process for a single Release Vulnerability.
- Copy & Paste allows an assessed Release Vulnerability (source) to be copied to one or more Release Vulnerabilities.
- Bulk Assess allows multiple unassessed Release Vulnerabilities to be simultaneously assessed.

User Authorization

- Users can request access to a specific Product from the new Security page. Users can also manage existing authorizations.
- The Product PSL will have authorization authority for Products. VAMS Admin will have authorization authority for non-Product groups/roles.
- Two new user roles added: Product Manager (R&D) and Customer Support (non-R&D)

Component Registration

- Users can register an existing component or request a new component to be added for vulnerability monitoring.
- VAMS will now include the entire catalogue of monitored components supported by our CERT vendor.
- Component Copy is available to move any or all components from one Release to another (will also copy all non-mitigate assessment details).
- New Component Status request is now available to track request submission and results.

- Users can register 3rd party Software Components manually via the Component Registration GUI or using a REST API to upload a Bill Of materials from Black Duck.

Scan Upload

- New GUI pages and steps for uploading scans – two-step process
- Create a Scan from Scan Management Page, where we can add scan and upload scan files
- Map and Commit from Scan Mapping Page, in which add Site Topo file and commit at release and /or team levels.

Reports

- Reports now use Distribution Date for calculating results instead of previous Publication Date when a Date Range is specified.
- Report criteria can be saved as a "Saved Report" for future recall.
- Saved Reports can be deleted when no longer relevant.
- PSM Report no longer requires it to be pre-configured before use.

2.2 Key Capabilities of CAPS (supporting R&D NSA Compliance)

2.2.1 Approach:

Automate workflows that coordinate and manage compliance tasks, and to leverage Nokia existing systems and integrate with existing business processes. Maximize use of off-the-shelf software to limit the development costs and reduce cycle time, emphasize security controls for the protection of sensitive customer and Nokia information.

2.2.2 Solution:

Rapid deployment across all Nokia R&D, and adoption in US Market, resulted in Nokia achieving compliance with critical NSA requirements. Seamless integration with Nokia process and systems eliminates need for manual work with significant cost. Built in scalability, flexibility, cost effectiveness and access controls give competitive advantage to Nokia in meeting increasing security compliance requirement worldwide.

2.3 Sensitive Data Handling in CAPS

2.3.1 Background:

- Nokia is subjected to increasingly stringent government and commercial compliance requirements placing unique demands on our product development, services and other business functions.
- With the acquisition of Alcatel-Lucent in 2016, U.S. National Security Agreement requirements became effective.
- In 2017, the Chinese government published to China Cybersecurity Law requirements
- In 2018, the EU General Data Privacy Regulation will become effective

2.3.1 Approach

- CAPS provide a secure, scalable, virtualized solution, utilizing state of the art open- source technologies:
- Redmine for role-based workflow management
- AngularJS for rapid Web GUI implementation
- Django to manage complex, high-performance database
- Nokia Private Cloud (using Amazon S3) for data storage
- Integrate with existing Nokia processes, systems, and tools

3. Robot Framework Automation Tool

Robot Framework is a generic open-source automation framework for acceptance testing, acceptance test driven development (ATDD), and robotic process automation (RPA).

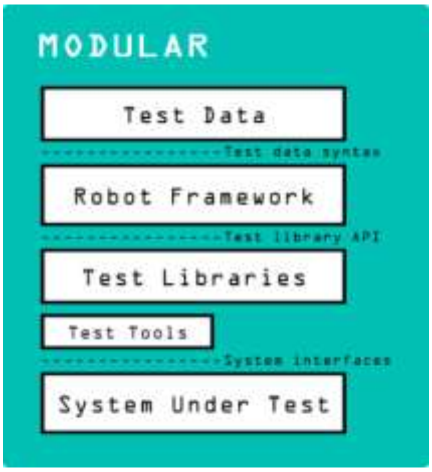
It has easy-to-use tabular test data syntax and it utilizes the keyword-driven testing approach. Its testing capabilities can be extended by test libraries implemented either with Python or Java, and users can create new higher-level keywords from existing ones using the same syntax that is used for creating test cases.

Robot Framework project is hosted on GitHub where you can find further documentation, source code, and issue tracker. Downloads are hosted at PyPI. The framework has a rich ecosystem around it consisting of various generic test libraries and tools that are developed as separate projects. Robot Framework is operating system and application independent. The core framework is implemented using Python and runs also on Python (JVM) and Iron Python (.NET).

Robot Framework itself is open-source software released under Apache License 2.0, and most of the libraries and tools in the ecosystem are also open source. The framework was initially developed at Nokia Networks and it is nowadays sponsored by Robot Framework Foundation.

3.1 Modular Format

Robot Framework was initially developed at Nokia Networks and it is used extensively around the whole company. It is used for testing different devices, software systems and protocols via GUIs, APIs, and various other interfaces



3.2 RIDE (An Integrated Development Environment)

The Robot Framework IDE(RIDE) is the integrated development environment to implement automated test for Robot Framework. It is a generic test automation framework. The latest available version of RIDE is currently 1.7.3.1. A big advantage of the ROBOT-IDE is the support in configuring different aspects of test suites.

Robot Framework is a Python-based, extensible keyword-driven automation framework for following feature:

- Acceptance testing
- Acceptance test driven development
- Behavior driven development
- Robotic process automation

It can be used in distributed, heterogeneous environments, where automation requires using different technologies and interfaces.

Higher-level keywords: Those are really testing a concrete aspect of the business logic of the system under test.

Lower-level keywords: To keep the implementation of the higher-level keywords at a decent size one is often breaking down the required functionality to several lower-level keywords.

Technical keywords: Those provide the technical implementation to access and thus test the system.

The framework has a rich ecosystem around it consisting of various generic libraries and tools that are developed as separate projects.

- Enables easy-to-use tabular syntax for creating test cases in a uniform way.
- Provides ability to create reusable higher-level keywords from the existing keywords.

- Provides easy-to-read result reports and logs in HTML format.
- Platform and application independent.
- Provides a simple library API for creating customized test libraries which can be implemented natively with either Python or Java.
- Provides a command line interface and XML based output files for integration into existing build infrastructure (continuous integration systems).
- Provides support for Selenium for web testing, Java GUI testing, running processes, Telnet, SSH, and so on.
- Supports creating data-driven test cases.
- Has built-in support for variables, practical particularly for testing in different environments
- Provides tagging to categorize and select test cases to be executed.
- Enables easy integration with source control: test suites are just files and directories that can be versioned with the production code.
- Provides test-case and test-suite -level setup and teardown.
- The modular architecture supports creating tests even for applications with several diverse interfaces.

The test data is in simple, easy-to-edit tabular format. Once Robot Framework is started, it processes the data, executes test cases and generates logs and reports. The core framework does not know anything about the target under test, and the interaction with it is handled by libraries. Libraries can either use application interfaces directly or use lower-level test tools as drivers.

The internal working process of Robot framework is explained in the Figure

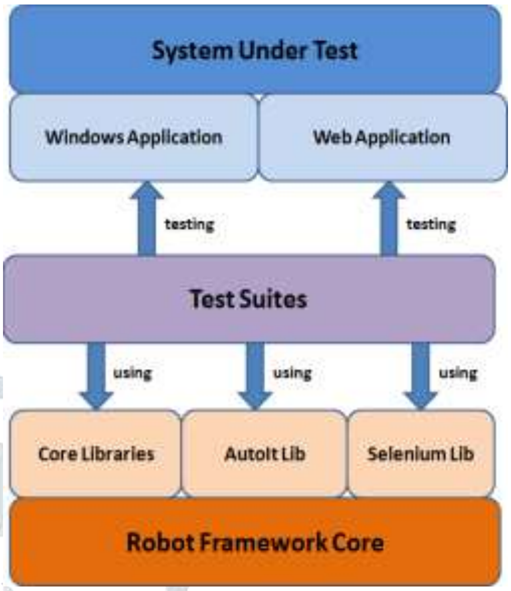


Figure: Architecture of Robot Framework

3.3 Implementation of Testcases in Robot Framework

Robot Framework data is defined in different sections, often also called tables, setting Variables, Test cases, Tasks, Keywords, and comments are explained below which is used to run the Robot Framework in the different platform.

Acceptance testing determines whether a system answers the acceptance criteria, defined by users' needs and requirements. When we execute acceptance testing, we verify the whole system as a single unit on a high level, to understand if the system under test can be used by the end users. Test engineers act as system users by executing steps and scenarios that come from requirements and business processes, by forming a set of predefined keywords. This approach to testing, which is based on set of keywords that can be re-used across all tests, is called keyword driven.

3.3.1 Browser Drivers:

The Selenium library is one of most used libraries for testing a web application interface. It interacts with the web application through its own driver. Each browser requires different Selenium drivers. Chrome and Internet Explorer (IE) need to have separate drivers; meanwhile Firefox does not require one.

3.3.2 Implementing Keywords in the Test Cases

A test case is composed from keywords. We can use keywords created by our own (they are named as user keywords) or import keywords from Robot Framework libraries. It's up to a test developer to choose which one to use. Keywords will make a test case easy to read and easy to understand. To be able to use keywords from external libraries (like Selenium Library) we need to import it. This should be done in the "Settings" section of the code in Robot with the setting "Library". The "Settings" section is used not only for importing external libraries and resources, but also for defining metadata for test suites and test cases.

Underneath the "Settings" section there is "Test Cases" section where we should add all the test cases within a test cases file. As you can see, we have used keywords from Selenium Library to open the browser and to select appropriate values from departure and destination drop-down lists.

In the Robot Framework, any keyword can accept any number of arguments. In our case, for example, the keyword "Open browser" accepts two arguments: the URL to open and the browser this URL should be opened in. The keyword "Select from List by Value" accepts the selector of the web element as the first argument and accepts the value to select as the second argument. The Tags section is used to assign a logical group to a test case. Furthermore, it can be used to execute tests only with the specified tag.

Conclusion

The primary goal of the project has been achieved with utilizing Robot Framework and Selenium2Libary to write and run automated test cases successfully. The user inter-face of a complicated application can be tested in a much shorter period. In addition to GUI testing, more available libraries such as API or Database can be used to serve specific testing purposes. Nowadays, automation testing plays an important role in the software development process because of the growing complexity of applications. It improves software quality and reduces project costs. Though automation testing brings many benefits, it cannot totally replace manual testing. Manual and automation testing need to be performed in parallel. A challenge of the project and of automated tests is that they are vulnerable to the change of the tested software. An update to the application may cause an unexpected testing result if the structure of the tests is not well organized. So, maintenance will become an issue when the number of test cases increases tremendously. Automation testing will keep growing in the future. Many automation tools have been created and more new and improved libraries for Robot Framework will be re-leased to meet the needs of the automation software testing community.

References

[1] *Nokia Networks*. (n.d.). Retrieved 10 31, 2019, from Wikipedia:
The Free Encyclopedia: http://en.wikipedia.org/wiki/Nokia_Networks

[2] .Robot Framework User Guide [online] available at https://robotframework.org/