CHAPTER III

SUVA SECURITY FRAMEWORK FOR CLOUD MIGRATION

3.1 INTRODUCTION

In this chapter, detailed description of SuVa Security Framework is presented. Initially, the objectives of the SuVa Security Framework are explained followed by the overview of framework. In the session SuVa Security Framework an introduction, framework is presented with its execution flow-work. Moreover, functional layers and architecture of SuVa Security Framework are illustrated. Further in following sessions business logic of SuVa Security Framework is described followed by its, work-flow management, scheduling [Ismail and Fardoun 2017] and Liu and Duan [2015], vulnerability scanning, penetration testing and results modules described. Thus, this chapter provides a detail overview of all the modules of the SuVa Security Framework and explaining its process, supported with necessary architectural diagrams.

3.2 OBJECTIVE OF SUVA SECURITY FRAMEWORK

There are hundreds of scanners which scan for many types of vulnerabilities such as network scanners, host scanners, web application scanners, multilevel scanners, database scanners, automated penetration testing, vulnerability scan consolidator etc. All these scanners are used in the cloud environment after the migration [Ismail et al. 2015] cloud of the IT infrastructure like databases, server and web applications. So, if there exists vulnerabilities, threats and risks in the on-premise database, it will be migrated to cloud as well. This causes the vulnerabilities and risks to spread across the cloud infrastructure by various technologies used in cloud like, for example during virtualization processes. So, corporates are now investing huge to overcome this problem. To mitigate this issue, SuVa Security Framework is presented. What SuVa Security Framework does is that this framework is deployed in the gateway of the cloud, so that all the migration of IT infrastructure, databases [Weis and Alves-Foss 2011], servers and web applications passes through the SuVa Security Framework enabling the scan for risks before it is being pushed to cloud. Hence, the SuVa framework helps in handling the security vulnerabilities without affecting the functionality of the cloud server.
The objective of SuVa Security Framework (SuVa SF) uses the combination of automated tools and techniques. This process undertakes an analysis of on premise’s (source) infrastructure and/or web application in order to determine the existence of and the extent of any vulnerability. All vulnerabilities are then categorized against criteria of Criticality, Exploit-ability, Impact; this will illustrate the true risk levels and provide "in-context" advice, how these vulnerabilities can be mitigated by means of possible solution, which shall be agreeable to both organization and cloud service providers. Vulnerability assessment service will provide with a detailed report on vulnerabilities and a range of recommendations to help to overcome issues listed. Further, vulnerability assessment will be carried out which is followed by penetration testing to validate vulnerabilities found during the scan. The findings of the vulnerability assessment & penetration testing are mapped against solution/action that needs to be taken.

Reports are tailored made by report engine, to meet the needs of the organization and cloud service provider. It shall provide delivering high-quality professional reports that outline clearly the vulnerabilities identified during the assessment, their potential impact and more importantly the report recommends processes for remediation. These reports are designed to be relevant and readable at all levels of an organization, in particular to the technical teams who are responsible for executing potential solutions indicated. Since the SuVa SF uses various techniques to scan a variety of vulnerable categories, the report will be compiled and delivered as consolidated specified by stakeholders.

3.3 SUVA SECURITY FRAMEWORK AN OVERVIEW

"SuVa Security Framework for Cloud Migration" methodology is presented in this session. During On-premise to Cloud migration, the applications and/or system data transferred to Cloud Infrastructure. The problem is that vulnerability and/or threats, which exist already in the premise, are migrated to Cloud (Figure 3.1). Only after the completion of migration, cloud provider initiates for the vulnerability scan and addresses the issue.
At present, on premise to cloud migration executed with various migration tools which are available in the market. Figure 3.1 shows on-premise to cloud migration without performing any security scan before or after migration. Only upon the request of security scan to the cloud provider, a security scan is initiated after data moved to the cloud. The SuVa security framework presented in Figure 3.2 provides the methodology wherein these vulnerability scans are done pro-actively before loading or transferring the data into a cloud environment.

Figure 3.2 shows the On-Premises to Cloud migration and the SuVa Security Framework at the entry "Gateway" point of cloud environment at a high level. This figure provides an idea where the framework exists in the cloud.
Figure 3.3: Cloud Computing Architecture

Figure 3.3 shows the basic cloud computing architecture without the framework been shown or deployed in the cloud framework.

Figure 3.4: Cloud Computing Architecture with SuVa Security Framework

SuVa Security Framework is embedded in Cloud Computing Architecture shown in Figure 3.4. So, in the course of data migration process, it goes through the SuVa Security Framework as a Gateway to enter into the cloud server. During this process of migration, data is simultaneously scanned for vulnerability assessment and corresponding reports and recommendations are provided. Thus, SuVa Security Framework is a methodology, which
pro-actively does the vulnerability assessment in cloud deployments. Each component of security framework is explained in detail in following sessions of this chapter. SuVa security framework is presented in the Cloud Computing architecture as ‘to-be’ scenario.

SuVa Security Framework (SuVa SF) thus can be represented as an additional layer in the Cloud Computing Architecture Layer. It is another form of representation within the Cloud Computing Framework Figure 3.4.

3.4 AN INTRODUCTION TO SUV A SECURITY FRAMEWORK

Under this session SuVa Security Framework explained. The SuVa Security Framework (Figure 3.5) has six modules namely; (a) Business Logic (b) Work-flow Engine (c) Scheduler (d) Scan Engine (e) Validation and (f) Reporting Engine. These six modules are grouped in to two categories namely; IT Management, Security Management. Furthermore, the execution flow of SuVa Security Framework is explained. SuVa Security Framework’s functional layers are described and continued with the framework’s system architecture. The details of the same are elucidated in detailed in the below sections.

3.4.1 SuVa Security Framework

This session illustrates the SuVa security framework in detail and shown in Figure 3.5. SuVa Security Framework has Security Module Platform in which the framework’s deployment modules are presented. Security module platform itself contains two major layers namely;

a) IT Management Layer and
b) Security Management Layer

IT management layer consists of three main modules which falls under the Business owner / the management team, who has the responsibility to take care of initial detailing and definition of these process. The modules areas follows;

a) Business Logic Module
b) Workflow Engine Module and
c) Scheduler Module

The detailed description of the above-mentioned modules will be discussed in the upcoming sessions. Here in this session, a high-level accountability of these modules explained in briefly.
Business Logic module will identify and record all the business rules appropriate to different scenarios taking into consideration of the risk levels, business priorities, business policies, and business processes etc. In Work-flow Engine module, the sequence of activities and processes involved are defined which is basically driven by the business logic criteria. This plays important role in predefining which should go first and others to follow etc. in the whole chain of activities. So, work-flow is always backed up by the rules defined in business logic and give it to scheduler modules as a relay mode. Scheduler modules mainly schedules the tasks in a predefined time interval as set in the work-flow engine. The time intervals are fixed by trial-and-error method. These time-intervals should be predefined by business logic together with work-flow processes, defining when these scheduled scan / task need to be executed and with defined frequencies. The frequency is scanning is set by trial-and-error method and it denotes number of scanning procedures to be performed in a specific time interval.

Under Security Management Layer, three modules are presented namely; a) Scan Engine Module
b) Validation Module and
c) Report Engine Module

Scan Engine module is the core module in the security management layer, which takes
care of finding out potential leakages in the system. This module’s main task is the scanning for vulnerabilities. Scan Engine module itself been categorized to three type of vulnerability scanning which are;

a) Network Based Scan
b) Host Based Scan and
c) Web Application Based Scan

Depending upon the input from IT Security module defined at the higher level, these scans are categorized and selected for the vulnerability scanning. These scans can be initiated by various scanning tools which are highly effective in their respective domains. The reports thus generated by the vulnerability assessment are further been validated in the Validation module. The term validation at this point indicates the quality assessment of the proposed security framework and is carried out with the help of penetration testing. The validation is thus being executed by means of penetration testing against the list of vulnerabilities found wherever applicable. Report Engine provides reports of vulnerability assessment as well for the penetration testing. These results can be viewed in different forms like in html, excel, and in special tailor made formats.

3.4.2 Process Flow of SuVa Security Framework

The objective of SuVa security framework is attained by following a standard process flow that involves several important processes. In order to outsource data to the cloud, the business logic is formulated. The term ’business logic’ decides the flow of execution and the business logic varies with respect to the kind of vulnerabilities, which are network, host and web application based scanning. Based on this, the business logic varies and the workflow of the SuVa is determined by the business logic. The business logic of SuVa security framework is explained in section 3.5. To be specific, the business logic defines the functionality of the framework. The work-flow is based on different procedures and processes. Based on the business logic and its work-flow, scheduler task is triggered which defines the time at which each activity gets performed.

It triggers basically scanning activity and subsequently checks whether tasks are completed. If either scan is not triggered or task is not complete then it goes back to loop and waiting for either trigger or task to get kicked-off. This is a continuous process loop until all scans are fully complete. Upon successful completion, report is generated which depicts vulnerability assessments. This report is basically a dashboard for stakeholders which de-
scribes what are the weak spot arenas. It is not necessary that all vulnerabilities need to go through penetration tests to validate whether identified threats are correct. Penetration test is meant for checking the robustness of the security mechanism and is not mandatory to check for the correctness of the security framework, all the time. This idea conserves resources and time as well. The penetration test is performed by introducing liabilities in Operation System (OS), web application and the network.

So, there is a decision box which decides whether vulnerability identified needs to be validated, if so, appropriate penetration tests are executed which confirm whether weak spots are the one to be addressed with remediation action plan. The penetration report which encompasses the problem identified and its remediation action items with its recommendations and suggestions.

Figure 3.6: SuVa Security Framework - Execution Flow Chart

The report of penetration generated from penetration execution is send to stakehold-
ers for them to take necessary action to re-mediate / mitigate. If the penetration test is not needed, then the flowchart ends with the results of validation assessment. Reports thus generated are compiled and designed in such a way it gives consolidated view of all categories of vulnerabilities. This work considers six important broad categories of vulnerabilities such as Byzantine failure, VM based rootkit, session riding, session hijacking, IP vulnerability, injection vulnerability, resource exhaustion and so on. There are two scenarios where in flowchart ends; 1) after creation of remediation report based out of penetration testing 2) If Penetration test is not relevant or needed. and outcome of penetration testing.

3.4.3 Functional Layers of SuVa Security Framework

The security framework is arranged in various layers from top to bottom approach namely; Presentation Layer, Business Layer, Programming Layer, Data Layer, Network Layer and User Interface.

![Functional Layer - SuVa Security Framework](image)

Figure 3.7: Functional Layer - SuVa Security Framework

The presentation layer consists of functions that are happening in various slices in the security framework; Dashboard, GUI, Reporting and Web Interface. By looking at the business layer, there are three functionalities are happening which are Business Logic, Workflow
and Scheduling. These functionalities are grouped into IT Management Layer of the SuVa Security Framework. In this layer, key activities like business requirements, sequencing and time machine activities are carried out.

The programming layer where actual execution happens, Vulnerability scanning, Penetration, parsing the results of both assessment and penetration testing which brings the nature of vulnerability and its possible fixes with suggestions and recommendations. Due to the fact of dynamic environment, as and when new threats appear, the same is updated, processed and gets stored in central database. In this way, we always do have updated information of vulnerabilities and its exploitation which enables to have right plug-in installed in both scanning and penetration systems.

The Data layer has few set of data; business logic requirements, work-flow (sequence/process flow), scheduling, threat data, exploit data and report data. It is imperative to say here is that everything revolves around data and the SuVa framework pro-actively finds out the vulnerabilities being present in the data, validate them with right penetration tests. In network layer, all server data, network information and rest-full server data are preserved. The last not but least, User is in the top most layer where in it provides means to trigger the whole process which having manual intervention.

3.4.4 System Architecture of SuVa Security Framework

This session will describe the system architecture of SuVa Security Framework, which comprises of many blocks which shall be detailed out in this section.
Figure 3.8: System Architecture - SuVa Security Framework

It has two categories of blocks namely; Front-End Web Portal, Back End. In the Front-end web portal, it has presentation information like dashboard, reporting, user input etc., whereas in the back end, it has number of blocks based on the specific functionalities. The Security interface is the component which is the bridge between back end and user front end. Based on the input fed, security interface gives it to Business Logic evaluator where in intelligence of business rules, business logic and its requirements are derived. This evaluator is merely the business intelligence mechanism where in all business logics are worked out and described. Basically, Business Logic is one which describes what is the requirements from the business domain in terms of its importance, risk level and its impact level.

The Workflow manager fundamentally based on the input from business logic evaluator, sequencing the various level of activities to be done. 3 plugins, 2 custom plugin and 1 plugin generator which supports Vulnerability scan system explained in fig 3.8 pg 54. This signifies that it ensures proper and right flow of information are tabled. Having this in place, constitutes entire bits and pieces are properly queued. It has three components namely; Sequential Manager, Scheduler and Performance Analyzer. The Sequential Manager puts all the data based on business logic intelligence is queued up and put it in sequential way,
Scheduler block is the one which schedule should be performed at what point of time with pause if applicable.

The Performance Analyzer is the one which is used for analyzing how it has performed in given point of time for performance dashboard. The controller block ensures that scheduling mechanism does follow according to the sequential manager.

All the tasks and associated data are stored in the server called Restfull server where in "Put/Get/Delete/Post" actions are taken place. This is treated as Security Management Server. From here onwards, the actual execution begin to happen. Using Server-Client architecture, client pulls the task/data from the server and hands it over to Vulnerability Scan System.

Another important aspect of vulnerability scan system is to have appropriate plugin installed to scan right vulnerabilities. 3 plugins, 2 custom plugin and 1 plugin generator which supports Vulnerability scan system explained in fig 3.8. How to choose appropriate is plug-in is described as follows; The Central DB is periodically updated by Vulnerability updater and Vulnerability processor, as and when new vulnerability appears, the updates gets the relevant information and pass it on to vulnerability processor. Upon processing the data provided by vulnerability updater, it updates the central data where it contains all types of vulnerabilities [Piazza and Olmsted 2016].

Based on the type of vulnerability, appropriate plug-in is generated by "plug-in generator" and the same is sent to Vulnerability scan system so that relevant plug-in is installed in order to carry out scanning. The actual scanning is performed at scanning system and the outcome of the same is presented to Result Parser, where in the scan results are parsed and identifies whether any penetration tests are relevant and needed. It is not necessarily that all scan results need to have to be exploited to be validated ensure that the weak spots are real weak spots.

There are two functional components "Report Generator" and "Result Analyzer" are the ones which does and determines whether any vulnerability needs to have any exploitation to be validated. The right plugin was chosen based on the vulnerability found, likewise right plug-in is needed for the exploitation purpose also. The right plug-in is generated by plug-in generator. Upon giving it to Exploiter it validates whether identified vulnerability is REAL threat. This mechanism confirms that the identified weak spot is the one which needs to be mitigated with remediation plan wherein right fix is recommended / suggested.

The report generator is the one which describes the validated vulnerabilities with rec-
ommendation and suggestions provided to both stakeholders; Organization and Cloud Service Providers. Please be noted that this is pro-active approach which enables our stakeholders to fix the problems before moving into cloud environment world.

3.5 MODULES OF SUVA SECURITY FRAMEWORK

In this section, various modules which are presented in the SuVa Security Framework within the IT management and security management layers are presented. The modules are Business Logic, Workflow Engine, Scheduler, Scan Engine (Network Scan, Host Scan, and Web Application Scan), Validation and Report Engine. An overview of all these modules has been explained.

3.5.1 Business Logic Module

First step is to identify the business rules applicable to various scenarios taking into consideration of source system’s data that might trigger logic necessary for three broader categories of vulnerabilities; network scan, host-based scan and web application based scanning. This business logic should be designed and developed based on the case-to-case basis. Business logic basically determines what level of details that scanning should perform.

The business logic is derived based on various inputs; Business Rules, Business Process, Data Design, Workflow, reporting requirements, Priority, Severity factor, scheduling and guidelines for escalation towards data governance. This logic is created based on the type of vulnerability assessment needed and its relevant penetration testing applicable.

The below presented diagram (figure 3.9) depicts the contribution of components to "Business Logic", which is designed in accordance with the business needs, type of data that organization migrate, the destination data model of cloud service provider, if applicable. Basically, business logic reflects the real time business requirements, which describes the structure and importance of data, sequencing etc., One of the business logic is that it describes how the data is interacting with one another.

Another example could be that, business logic would define to scan for agentless auditing, reporting and patch management integration in case of network-based scanning.
The impact of business logic varies from application to application. The design of business logic varies depending on the various inputs /criteria /aspects mentioned above in the picture. Business logic tests should be designed to ensure that (i) the control is in place to implement the business rule, (ii) the control is implemented correctly and cannot be bypassed or tampered with, and (iii) the control is used properly in all the necessary places. Business rules should be clearly defined and checked against during the different development phases of the framework: design, implementation, and testing. Clear documentation and threat modeling/abuse cases and code reviews should be used.

This module plays vital role in the whole SuVa frame architecture being the fact that this is entry component of the entire framework. Hence, the perfect business logic helps the succeeding components to take over the control.

3.5.2 Workflow Module and Scheduler Module

In this session, both the workflow module and scheduler module is combined because both are interrelated.

Basically, the workflow engine calls various type of vulnerability scans periodically as defined by the scheduler. Depending on the business logic already defined the workflow acts. For example, as defined in the business logic the workflow triggers the all three categories
of scan sequentially. After completing of the scan, the next action validation takes place. Figure 3.10 exhibits a simple flow chart of workflow and scheduler.

Scheduler is a module of SuVa Security Framework. Task scheduler has control on the indicated time if it is defined to do so. The task of a scheduler (references Shishira et al. [2017] and Choi and Lee [2016]) is to program the scripts or define a task in predefined time intervals. It has the ability to launch the programs at the scheduled time as predefined. The scheduler can be automatically started/stop for any specific task which is already been programmed. Task scheduler has also control on automated tasks to start or stop when it is not operational and to save the records in regular intervals or at an indicated time.

Start time need to be configured for each and every task or it can be defined to repeat at a defined interval. For example, schedule a task to run a backup results script every night. Then scheduler should be already programmed to trigger the certain workflow. Workflow is being already predefined to execute the certain task which processes are clearly linked to
business logic.

### 3.5.3 Scan Engine Module

SuVa Security Framework’s scan engine simple form is shown in the Figure 3.11. User interface triggers the scan engine for the vulnerability assessment process. Scan engine scans for the already detected vulnerabilities being present in the central database. After the assessment of the vulnerability the scan engine delivers the report.

![Figure 3.11: Scan Engine Architecture - SuVa Security Framework](image)

Vulnerability assessment process shown in Figure 3.11 starts with the user interface, to initiate the process of scanning action. First the systems IP ranges are selected for which the vulnerability assessment screening is necessary. Configuration manager makes sure the background configurations are set for the specific IP ids. Vulnerability assessment starts by acquiring the input from the task scheduler. Scanner uses the central vulnerability database in order to get the known vulnerabilities. Using this as the reference mark, scanner scans for the vulnerabilities in the targeted system. SuVa Security Framework has the intelligence to choose for either one of the network based scanner or host based scanner or web application based scanner. Once either one of the scan is executed, report engine generates the scan report.
Thus, generated scan reports are parsed for the types of the vulnerabilities, their severity and risk. After the view or downloading the report the actual process of vulnerability assessment ends.

### 3.5.4 Validation Module

In this session, the validation process is executed by exploiting the weakness. This is performed by the penetration testing / ethical hacking.
The process of penetration testing or ethical hacking is shown in the Figure 3.13. Results generated after the vulnerability assessment is taken as the start process of the penetration testing. Not all the vulnerabilities can be exploitable. There are restrictions for certain type of vulnerability which cannot be exploitable. Vulnerabilities are the weakness of the system, whereas exploit is a security attack that influence over the vulnerability. Hence vulnerability is not exploitable for three major reasons and they are

1. The inadequate information may prevent the attackers to exploit the vulnerability.
2. Exploitability may require access to the system, which the attacker may not have.
3. The tightened security level may prevent the attacker to exploit the vulnerability.

Thus, first the results should be analyzed and certain vulnerabilities can be recommended for the exploitation. The exploitation can be executed by the penetration testing tools. Results are reported after exploiting the weakness in the system and confirms the existence of vulnerability.

Reports delivered includes the remediation of the issues found and informed to the resource owner. The scan report summarizes the threat, impact, solution and results. The solution suggests the data owners to overcome the threats being detected in the cloud data. Kind of solutions in detail provided in the session 4.2.5, 4.3.5, 4.4.4 and 4.5.5.
3.5.5 Report Engine Module

Scan engine component consists of network scan, host scan, and web application scan. The SuVa SF executes three levels of scanning mentioned above. These are the core components of SuVa SF, where the actual vulnerability scans take place.

During this process, the data are scanned for all vulnerabilities found in the network based, host based and web application based. In common practices ‘as-is’ note that only certain type of tool does certain category of scans whereas, in SuVa SF, all category of vulnerability assessment scans executed together with penetration testing which takes place as per the work-flow instruction. When the scans completed, reports are stored in the report engine. The main task of the report engine is that it will compile the reports delivered by the various categories of scans and delivers a consolidated report. The consolidated report contains both vulnerability and penetration test reports. In ‘as-is’ scenario the reports are delivered only tools wise individual reports with separate reports for vulnerability assessment and penetration testing report.

The vulnerability assessment and penetration testing reports address all issues found the system/ server/ storage. Reports not only contain the issues to address but also deliver the possible solution to eradicate the issue. Reports can be download in various formats like .txt, .csv, .pdf and .html.
In this chapter, the SuVa Security Framework’s core design is explained in detail. SuVa Security Framework to be deployed in cloud framework as a gate way through which the data, web applications, storage, etc., are been transferred to cloud. SuVa Security Framework makes sure that the secured data are moved to cloud during on-premise to cloud migration. Objective of SuVa Security Framework is explained clearly with the overview of where the framework exists in cloud computing architecture.

SuVa Security Framework’s introduction session presents the core design of the framework under security module platform with explanation of layers namely; IT management layer and security management layer. Under both layers six core modules exists, three modules under each layer in higher level. In IT management layer, there exists modules namely; business logic, workflow and scheduler. Under security management, three modules presented namely; scan engine (network scan, host scan and web application scan), validation and report engine. Further, an execution work flow is presented explaining how this framework can be executed. Functional layers are described namely; presentation layer (Web Interface, Dashboard, GUI and Reporting), Business Layer (Business Logic, Workflow and Scheduling), Programming Layer (Vulnerability Updater, Vulnerability Processor, Vulnerability Scan System, Central Database, Result Parser and Exploiter), Data Layer (Workflow Queue, Scheduler Task Data, Vulnerability Data, Exploit Data, Report Data and Business Logic Data) and Network Layer (Restfull Server, Network and Storage).

In the System architecture of SuVa Security Framework, back end components are presented and their inter relation is explained in detail and how does it is connected to the front end is presented. In the backend, main components like security interface, business logic evaluator, workflow manager, scheduler, restfull server, vulnerability scan system, vulnerability processor, plugin generator, result parser, result generator, exploiter are described in detail.

Detailed description of business logic module is explained with the factors to be considered while defining the business logic. In workflow and scheduler module, a work flow of execution is presented explaining how the business logic initiates the flow to the work engine and scheduling takes place. Scan engine module is presented with its simple scan engine architecture with the vulnerability assessment process of SuVa Security Framework.

Validation module is described with the penetration testing process of the SuVa Se-
curity Framework, where the ethical hacking process is explained in detail. Report engine module explains the process undergone in reporting with details of scanned vulnerabilities which are network based, host based and web application based. Thus, the reports can be parsed and downloaded in various formats like .txt, .csv, .pdf and .html.