UNIVERSITY OF NAIROBI
SCHOOL OF COMPUTING AND INFORMATICS
SECURITY AS A SERVICE FRAMEWORK FOR BROADBAND USERS

BY

ROTICH ERICK KIPTANUI

REG. NO: P58/9099/2006

SUPERVISOR: ROBERT OBOKO

February, 2011

Submitted in partial fulfillment of the requirements of the Master of Science in Computer Science

DECLARATION

This project, as presented in this report, is my original work and has not been presented for any other University award.

Sign:  **Date:** 28/3/2011

ROTICH ERICK KIPTANUI
P58/9099/2006

This project has been submitted as partial fulfillment of the requirements for the Master of Science degree in Computer Science of the University of Nairobi with my approval as the University supervisor.

Sign:  **Date:** 16/5/2011

Robert Oboko
Project supervisor
School of Computing and Informatics
University of Nairobi

ACKNOWLEDGEMENTS

Special thanks go to my able supervisor Mr Robert Oboko for his continued assistance during this study. I also want to register my sincere gratitude to the Vlir UOS for the kind assistance they extended to my study by providing funds that gave me a big boost in doing my research. I am indebted to my family, my loving wife and children for their tireless support during this study. Last but not least thanks to my fellow students whom their continued encouragement made this study a success.

ABSTRACT

Currently most businesses rely on broadband to run networks and web applications. This means introduction of vulnerabilities in Universal Resource Locator (URL) that are in connection. Effective monitoring of URLs is crucial for any organization; each port has vulnerabilities associated with it. This research project explored the use of Security as a Service (SaaS) business model to deliver security to users through a web based technology. The project implements a web based technology software with an aim to assist broadband users to scan for vulnerabilities in a URL based on open ports or ports in use and suggest ways of mitigating the vulnerabilities discovered. Results from several sample websites are given, exceptions, constraints and achievements. The project also compares the findings to related previous work and then gives a conclusion and recommendation for further work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABREVIATION

SaaS	security as a service/Software as a service
IP	Internet Protocol
ASP	Application service provider
SQL	Structured query language
HTTP	Hypertext Transfer Protocol
URL	Uniform Resource Locator
TCP/IP	Transmission Control Protocol Internet Protocol
UMTS	Universal Mobile Telecommunication System
IANA	Internet Assigned Numbers Authority
UDP	User Datagram Protocol
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
VPN	A virtual Private Network
LDAP	Lightweight Directory Access Protocol
DMZ	Demilitarized Zone
SLA	Service Level Agreement
ADSL	Asymmetric Digital Subscriber Line
BOP	Business Operation Platform
SDF	SaaS Deployment Framework
TCO	Total Cost of Ownership
MARS	Monitoring Analysis and Response System
PTA	Practical Threat Analysis
CASE	Computer-aided Software Engineering
RMF	Risk Management Framework
HIPAA	Health Insurance Portability and Accountability Act
NIST	National Institute of Standards and Technology
ALE	Annual Loss Expectancy
IVR	Interactive Voice Response
CRM	Customer Relationship Management
SMS	System Management Server
CTAP	Cyber Threat Analysis Program
P2P	Peer to Peer

CHAPTER 1: INTRODUCTION

1.1 Background

Computer data and applications security is a fundamental aspect of computing that if taken care of can improve users computing experience. To safeguard users from malicious attacks and disruptions, most computers are protected using PC based anti-virus or anti-malware applications and personal firewalls. These measures though are effective in normal computer environments, are no longer effective for businesses that have embraced the broadband and fiber optic connectivity.

The emergence of broadband and fiber optic connectivity provides opportunities for users to improve their work environment while at the same time opens the users to wide range of data and network security threats. These threats are a result of fiber optic connectivity design that provides their users with at least one public static or dynamic IPs. Most users are aware of the situation while others are not. Those that are aware of their public IP still have no way of ensuring their applications, data, and network components are safe.

This research project explored software as a service approach to discovering and mitigating network and web applications vulnerabilities. The research aimed at helping broadband users to discover their security vulnerabilities and ways they can mitigate them by engaging security as a service provider.

1.1.1 Difference between SaaS and application service provider (ASP)

With the SaaS model, application functions are delivered remotely over the Internet and by a subscription model. Customers don't own the software, and have no choice of what type of hardware and middleware are used to host the software.

In the ASP model, customers buy the software, which is hosted by the service provider, who may decide to bring it, in-house at any time. The infrastructure may be tailored to customer needs.

1.2 Definitions and Concepts

1.2.1 Vulnerability scanning

Vulnerability scanning is similar to packet sniffing, port scanning. Vulnerability scanning is the automated process of proactively identifying vulnerabilities of computing systems in a network in order to determine if and where a system can be exploited and/or threatened. While public servers are important for communication and data transfer over the Internet, they open the door to potential security breaches by threat agents, such as malicious hackers. Vulnerability scanning employs software that seeks out security flaws based on a database of known flaws, testing systems for the occurrence of these flaws and generating a report of the findings that an individual or an enterprise can use to tighten the network's security.

Vulnerability scanning typically refers to the scanning of systems that are connected to the Internet but can also refer to system audits on internal networks that are not connected to the Internet in order to assess the threat of rogue software or malicious employees in an enterprise.

1.2.2 Port scanning

Is the Sending of queries to servers on the Internet in order to obtain information about their services and level of security. On Internet hosts (TCP/IP hosts), there are standard port numbers for each type of service. Port scanning is also widely used to find out if a network can be compromised. A port scan consists of sending a message to each port, one at a time. The kind of response received indicates whether the port is used and can therefore be probed for weakness.

1.2.3 Web application scanning

Web Application scanning is a method used to test web applications for common security problems such as Cross-Site Scripting, SQL Injection, Directory Traversal, insecure configurations, and remote command execution vulnerabilities. Scanning tools crawl a web application and locate application layer vulnerabilities and weaknesses, either by manipulating HTTP messages or by inspecting them for suspicious attributes.

1.2.4 Network scanning

Network scanning is a procedure for identifying active hosts on a network, either for the purpose of attacking them or for network security assessment. Scanning procedures, such as ping sweep and port scans, return information about which IP addresses map to live hosts that are active on the Internet and what services they offer. Another scanning method, inverse mapping returns information about what IP addresses do not map to live hosts; this enables an attacker to make assumptions about viable addresses.

1.2.5 Software as a Service (SaaS)

Software as a Service - also known as SaaS, is new business concept that has potential to revolutionize the way people have used software traditionally. In Software as a Service (SaaS) model - software is hosted generally on centralized network servers to make functionality available over web or Intranet. From developer point of view we can say that such SaaS can be managed and maintained easily compared to hassles involved in other software models. For end users such service is easy to consume and there is a possibility of using such service on pay per use basis.

1.3 Problem statement

Currently most businesses rely on broadband to run network and web applications. This means introduction of vulnerabilities through open ports at any instance in URLs that are in connection.

Each open port has vulnerabilities associated with it, hence the dilemma of which ports are Open? What are the vulnerabilities to these open ports? How can these vulnerabilities be mitigated?

Figure 1 below shows how a classical security setup is.

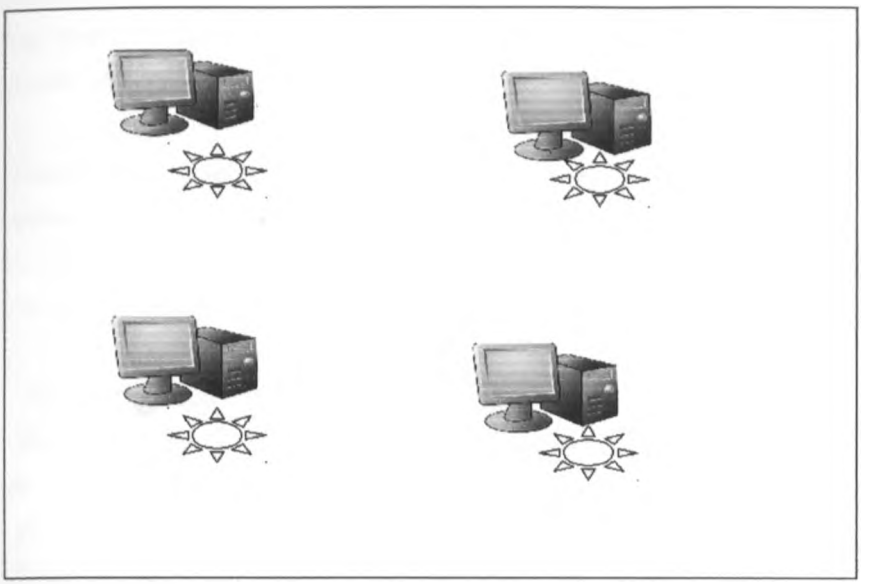

FIG 1: CLASSICAL SECURITY SETUP

Each workstation in this setup has anti-virus/anti-malware software.

1.4 Objectives

From the problem statement of this research, our focus was to achieve the following key objectives:

a) To study software as a service framework.

b) To create Agents to perform port scan to find open ports.

c) To show vulnerabilities associated with the open ports

d) To suggest possible mitigation.

1.5 Significance of the study

This research project will be of great significance given the current state of vulnerabilities facing the broadband users. The main significances of this study are outlined below.

a) Network vulnerability discovery and Suggested mitigation

Network perimeter vulnerabilities will be discovered and mitigation measures Suggested. Security alerts can be provided for specific flows thus raising the level of awareness.

b) Improved security manageability

Through special dashboards users will have an informative look of their security status. The visualization tools will help users to address the most important areas of their security.

c) Enhance broadband and fiber optic adoption

The high level of security awareness will facilitate higher adoption rates for broadband. Users will be comfortable with connectivity since their data and a dedicated provider will monitor application security.

d) Maximize Return on Investment for security services

Since software as a service is a model that involves metered usage from a reliable provider, users will only pay for what they use as opposed to current means of software procurement. The security providers have expertise in the area and are therefore able to give services at cheaper rates.

1.6 Assumptions and limitation of the research

Small Business entities on spotlight use Internet connectivity with static/dynamic IP settings and have a firewall.

i. The applications running under the firewall are TCP/IP applications
ii. Vulnerabilities are the flows that can be detected on the TCP/IP

1.7 Summary

This chapter prepares way for an in depth research on Security as a service for broadband and fibre optic connectivity. The research was carried out by a programming approach.

CHAPTER 2: LITERATURE REVIEW

2.1 Review of literature and related work

2.1.1 F-Secure

3 Italia now offer their mobile broadband users Internet security provided by F-Secure. F-Secure is the first operator in the world to launch commercial UMTS services in 2003, have signed an agreement to offer protection for PCs to all customers using 3 Italia's Internet dongles to access the web. 3 Italia's Internet security solution which is provided by F-Secure, is a full security suite which, in addition to antivirus, includes several other functionalities such as Browsing Protection, Firewall, Antispam and Parental Control which cares for the safety of children when they access the web.

3 Italia's Antivirus Security Suite by F-Secure provides a complete security solution for PCs against all Internet threats. It offers outstanding performance that is light on memory use and overall system impact, but fast in detecting and blocking malware with the latest protection technologies.

F-Secure – Protecting the irreplaceable makes sure that the clients are protected and safe online whether they are using a computer or a smart phone. They also backup and enable you to share your important files. Their services are available through over 200 operators around the world and trusted in millions of homes and businesses. Founded in 1988, F-Secure is listed on NASDAQ OMX Helsinki Ltd. (http://www.f-secure.com)

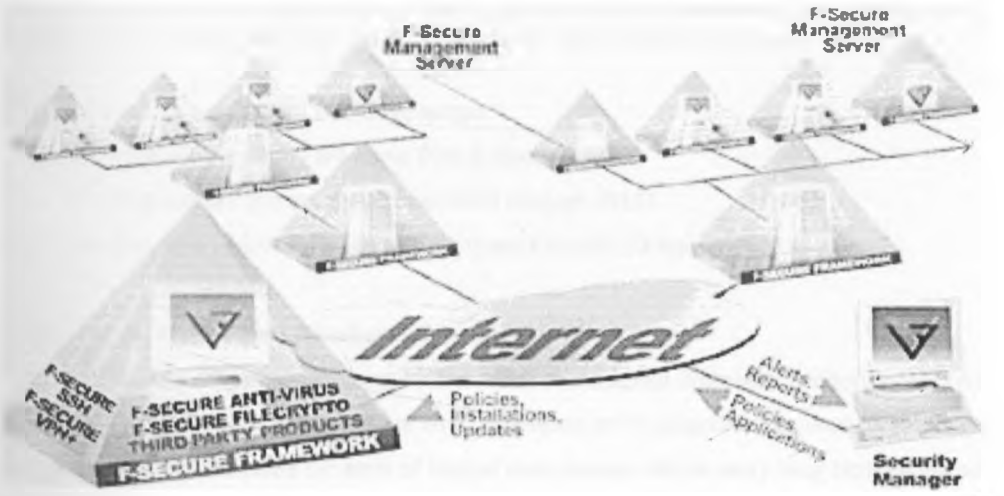

FIG 2: THE F-SECURE SOLUTION

2.1.2 BinarySEC© SaaS

BinarySEC© SaaS is an example of security as a service solution which blocks all abnormal traffic on a company's website before it reaches its server and protects against data theft, denial of service, identity theft and new attacks from the web. (http://www.binarysec.com)

2.1.3 McAfee Security-as-a-Service

McAfee Security-as-a-Service solutions are designed to provide organizations of all sizes, from small to large enterprises, with a comprehensive set of security products built on a Software-as-a-Service model. This strategy leverages McAfee's core strength in threat prevention. (https://www.mcafee.com)

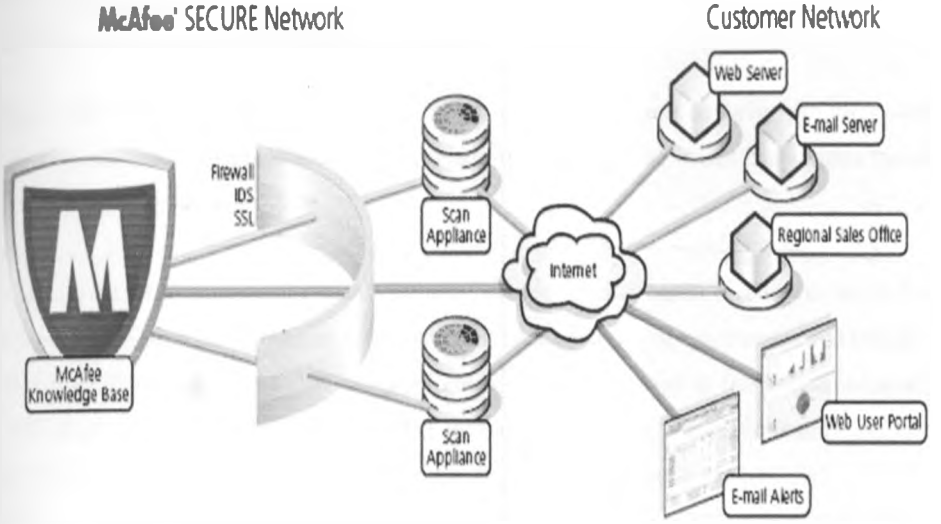

FIG 3: MCAFEE SECURITY-AS-A-SERVICE

2.1.4 Port Numbers

A port is an interface on a computer to which you can connect a device and a port number is part of the addressing information used to identify the senders and receivers of messages.

The port numbers are divided into three ranges

i. The Well-Known Ports are those from 0 through 1023.
ii. The Registered Ports are those from 1024 through 49151.
iii. The Dynamic and/or Private Ports are those from 49152 through 65535

2.1.5 Well Known Port Numbers

The Well-Known Ports are assigned by the Internet Assigned Numbers Authority (IANA) and on most systems can only be used by system (or root) processes or by programs executed by privileged users. Ports are used in the TCP to name the ends of logical connections, which carry long term conversations. For the purpose of providing services to unknown callers, a service contact port is defined. This list specifies the port used by the server process as its contact port. The contact port is sometimes called the "well-known port". To the extent possible, these same port assignments are used with the UDP .

The range for well-known ports managed by the IANA is 0-1023.(http://www.iana.org/assignments/port-numbers)

2.1.6 Security-as-a-Service: How SaaS Can Improve Your Organization's Security

Security-as-a-Service deliver security products and services in an on-demand model. The front-runners in this space are the anti-* products: anti-virus, anti-spam and anti-spyware where signature files are often updated on a daily basis. The Security-as-a-Service vendor manages the constant software and signature updates, eliminating the hassle of keeping the myriad security scanners up-to-date.

This SaaS model can also be extended to traditional in-house security products – firewalls, intrusion detection systems (IDS) and intrusion prevention systems (IPS) which can require specialized expertise to configure and manage.

SaaS can be implemented for email security by not routing email directly to an in-house server instead email is routed to a third-party provider where it is scanned for Spam, viruses and unsafe links using always up-to-date threat signatures. The cleansed email is then delivered to the in-house email server for internal distribution. Email administrators can log into a web console to modify scanning rules to meet company security policies. They can then see scanning statistics and generate reports.

We're seeing the *-as-a-service model being implemented in other areas e.g. Storage-as-a-Service, Hardware-as-a-Service, Database-as-a-Service. On-demand is a compelling model. (http://supplychaintechnology.wordpress.com)

2.1.7 Webroot® E-Mail Security SaaS

Webroot® E-Mail Security SaaS delivers enterprise-class security with better manageability, better value and better protection than any other e-mail security solution. There is no hardware or software to deploy meaning a lower total cost of ownership and guaranteed protection against Spam, viruses and service downtime. (http://www.asl-webroot.co.uk/)

2.1.8 Online protection and email storage from McAfee and Reboot Twice

The power of online security protection and email storage solutions is at your fingertips. With convenient online ordering, you can put SaaS security solutions to work for your business (http://www.reboottwice.com)

Zscaler provides risk mitigation and policy enforcement for businesses through its cloud service, while enriching the user's Internet experience. Organizations do not need to purchase, deploy, or manage countless point products. Companies simply define their corporate security control and compliance policy by accessing the Zscaler utility. The web traffic leaving the network firewall is easily redirected to data centers in Zscaler's global infrastructure. Based on an organization's policy, traffic is blocked, throttled, or

allowed to access the Internet. As the browser retrieves the web pages, Zscaler scans it for a range of malware threats and delivers clean traffic to the end user. (http://www.zscaler.com)

2.1.9 Technological Capability of SaaS

The core technology of SaaS is centered on its multi-tenant architecture. (Chong and Carraro, 2006) characterized SaaS as "Software deployed as a hosted service and accessed over the Internet." In order to provide efficient and effective services to SaaS clients, the SaaS providers must design their application architecture as "scalable, multi-tenant-efficient, and configurable" (Chong and Carraro, 2006).

2.1.10 Security is a broad term that can be broken down into three areas:

Data center, application and user. Each of these areas has its own security best practices schema and ignoring any one area presents a security vulnerability to the firm and its data. The best SaaS providers in the market discuss each of these areas with their customers, demonstrating how their SaaS solution is as secure, and frequently more secure, than what an internal IT department can provide.

2.1.11 Data Center Security

There are only two points of entry into a SaaS environment: The front-end, which the users utilize; and the back-end, used by the SaaS provider for maintenance and management. Limited entry eliminates all the ways in which data is lost or stolen. Front-end entry is always through a secure, encrypted VPN (A virtual private network) leveraging identity and role-based access.

2.1.12 Application Security

Application security is directly associated with identity and role based access permissions. Application security includes, but goes beyond the standard password access. By utilizing SaaS, application security also includes encryption of the password, logs the number of attempts to logon, and can encrypt field/text/attachments. Application security also disables Java Scripts, one of the leading causes of malware and malicious activities.

2.1.13 User Security

User security is rooted in role-based access and identity management. Identity management is maintained in the firm's (Lightweight Directory Access Protocol) LDAP directories. The firm's administrator controls permissions and denials. The directories can be either inside the firm's firewall, at the SaaS provider's site, or in a DMZ (demilitarized zone). The DMZ is the external touch point for your core business applications, business data, and file transfers. Having the firm control the identity management directories enables the administrator to move quickly to enable or disable users as needed.

2.1.14 Advantages of SaaS for Small Businesses

Due to the low initial cost of SaaS applications attracts small businesses. SaaS applications are charged based on the subscription rates per computer and any additional charges for storage or bandwidth usage. So the business ends up paying for only the actual use by its employees.

SaaS applications can be used on almost any device that can be connected to the Internet and is not dependent on the hardware quality or storage devices installed on each device. There is little to no learning curve involved especially where employees are already used to working on the Internet.

There is no need for the business to worry about upgrades and patches as the SaaS application provider would be doing these improvements on an almost daily basis and the upgrades would be available to all users at all times. So there is no downtime of computers frequently associated with such upgrades where the applications are installed on the computer.

Most SaaS applications are very easily integrated into any existing software with a business. They can also easily expand their capabilities to cater to any increased usage in an organization once the minimal hardware is in position.

The owner of a small business is easily able to log onto the SaaS applications from wherever he is and thus be in constant touch with the latest data available in his company's storage. This also allows a small business to authorize its travelling representatives to use the IT knowledge of the company from wherever they are.

Another major benefit is that some SaaS products, such as Severa 3 allow you to create a free account for a limited number of users. This allows the business to fully investigate the product without any initial expenditure. In fact, some businesses may be able to get everything they need within these free accounts.
SaaS applications have found very wide usage among HR and project management professionals as they are easily able to access monitoring and other software that they need for the implementation of their own tasks.

One of the tough challenges faced by small and independently owned business is finding software capable of handling project management needs while still being affordable and easy to use. Severa 3 is a web-based project management application that includes client management and invoicing tools, making it a very attractive solution for those seeking a single package that can handle all of their project-related and billing needs. Network-based access to, and management of, commercially available software

i. activities managed from central locations rather than at each customer's site, enabling customers to access applications remotely via the web

ii. application delivery typically closer to a one-to-many model (single instance, multi-tenant architecture) than to a one-to-one model, including architecture, pricing, partnering, and management characteristics

iii. Centralized feature updating, which obviates the need for end-users to download patches and upgrades.

iv. frequent integration into a larger network of communicating software

2.1.15 The roles of the major stakeholders in a SaaS environment

i. *SaaS provider*, who owns the SaaS framework and provides different services, For example, if the SaaS framework is deployed within a company or enterprise, the company or enterprise may be the SaaS provider.

ii. *Application owner or tenant*, who typically owns one or more applications in the SaaS platform. This stakeholder is responsible for providing features to meet end user requirements. The features and forms-driven processes in a sales application may be different from those in a HR application. If the SaaS framework is deployed within a company, different business units within the company could be the application owner.

2.1.16 Drawbacks of SaaS

a) For the Customer & End User:

i. **No direct control of the data** - One of the biggest hurdles to get over is the control of the data. Specifically, what happens when things go wrong? It is important that for the SaaS provider to ensure that data is safe. When the SaaS provider goes under, deeper implications surrounding the vital business data have to be considered.

ii. **Internet connection required** - it could affect your operations if you need to access an application and the Internet connection is down. A good set of companies are trying to solve this problem by allowing their applications to continue to work in a disconnected fashion for a period of time but at some point you will need to synchronize back up to the server.

iii. **Dependence on an outsider to run your business** - In a big way, you trust an outsider to help you run your business, and if they are not keeping their promise it can really affect you.

iv. **Security awareness** - Another big hurdle is security. This concern is the umbrella that is home to the concerns above, as the common thread among them all is that they make you consider how "secure" you feel with SaaS. You trust your really valuable data to someone else most security breaches occur because of disgruntled internal employees. Who end up selling or releasing the data when they are fired or when they quit, having your data managed and stored by an expert of the application is not a bad idea as long as they take it as seriously as you would.

b) **For the Provider:**

i. **Focus on customer satisfaction** - SaaS providers need to focus on customer satisfaction month in and month out or they will lose their customers. They need to earn their customer's business every month or they can simply leave. Contrary to on-premise deployments which are very costly and time consuming, if your customer is unhappy with the service he can leave at any time with very minimal cost.

ii. **Harder development process** - There are many different approaches to writing SaaS applications. Things like tenant isolation, provisioning and scalability to mention a few could be a hard thing to tackle where you wouldn't even have to think about if you were writing an on-premise application. It is hard to find the right talent as the skill sets required are more advanced than for its on-premise counterpart.

iii. **Compensation issues** - One of the early problems for SaaS providers is how to maintain operations when there is only very little money. SaaS deals are much smaller so initially it will be a lot harder to maintain operations unless you are properly funded so you can survive until you get enough money.

iv. **Success can be a problem** - You've heard many times that being too successful is a great problem to have but in the case of SaaS it can literally bring you to your knees if you are not prepared. As mentioned above SaaS development is hard. Things can grow out of control if the application is not architected properly and addresses scalability issues and your service can become unusable over time if it does not scale properly with the addition of new tenants.

2.1.17 Advantages of SaaS

a) **For the Consumer:**

i. **No client/server software installation or maintenance** – There is no planning and implementation guides.

ii. **Shorter deployment time** – Can take some few minutes to deploy as opposed to a phased implementation that could take months.

iii. **Global availability** – The technology exists to make on-premise software available outside of the premises, but we're talking about functionality that is available from anywhere on the internet natively.

iv. **Service Level Agreement (SLA) adherence** - Reported bugs can be fixed minus any rollout overhead. The provider actually has to fix the issue, but assuming they've deployed a moderately efficient SaaS application the rollout of a patch or fix should happen in the blink of an eye.

v. **Constant, Smaller, Upgrades** - When you use a SaaS application, it is in the best interest of the provider to keep you happy and they can do so by constantly improving the application experience. With SaaS this can come in the form of consistent miniscule changes that add up over time instead of monster patch and upgrades that cost you time and money to implement.

vi. **Ease Your Internal IT Pains** - Most of the last several points here highlight that SaaS offloads a great deal of IT pains incurred by software consumers in the traditional client/server model. This leaves IT personnel to focus on improving the day-to-day technical operations of your company instead of being called upon to troubleshoot third party software or maintain aging infrastructure. This leads to increase in throughput.

vii. **Redistribute IT Budget** - By outsourcing software functionality to a provider, the enterprise realizes a cost savings in infrastructure requirements and IT personnel knowledge requirements. This allows the enterprise to focus on core competencies. It also means that the cost savings from using SaaS applications can be flat out saved, or reallocated to boost productivity through other services.

b) **For the Provider**:

i. **Aggregate operating environment** - as a provider, you own your domain. You don't send technicians to fix or customize your software you have complete control to optimize an infrastructure to your SaaS application's specific requirements. This leads to financial savings as well as less headaches.

ii. **Predictable Revenue Stream**

The subscription model associated with SaaS means that your customers will pay you on a recurring schedule. If you make this cycle flexible enough, you can get a real handle on forecasting revenues.

iii. **Predictable Growth**

The fact that users hit your site to access the application means that with the right tools you can monitor their usage pretty closely - something that's not so easy with all your customers running the application on premise.

iv. **Focus On Smaller Upgrades Instead of Monster Patch Rollouts**

Your development teams can focus on fixing core application functionality, tackling bugs and enhancing features in smaller incremental rollouts because it's just easier to do so.

v. **Sales Becomes Customer Relationship Management**

When you are selling a subscribable service, the game of gaining subscribership becomes one of balancing user retention versus attrition. It's important to have a team out there to sell your application.

2.1.18 Broadband Internet access

Often shortened to just "broadband", is a high data rate connection to the Internet— typically contrasted with dial-up access using a 56k modem. Dial-up modems are limited to a bit rate of less than 56 Kbit/s (kilobits per second) and require the dedicated use of a telephone line — whereas broadband technologies supply more than double this rate and generally without disrupting telephone use. Although various minimum bandwidths have been used in definitions of broadband, ranging up from 64 Kbit/s up to 4.0 Mbit/s ("Birth of Broadband". ITU. http://www.itu.int/osg/spu/publications/birthofbroadband/faq.html. Retrieved July 21, 2009). The 2006 OECD report ("2006 OECD Broadband Statistics to December 2006". OECD. http://www.fcc.gov/cgb/broadband.html. Retrieved June 6, 2009).

The (US) Federal Communications Commission (FCC) as of 2010, defines "Basic Broadband" as data transmission speeds of at least 4 megabits per second (Mbps), or 4,000,000 bits per second, downstream (from the Internet to the user's computer) and 1 Mbit/s upstream (from the user's computer to the Internet). ("Sixth Broadband Deployment Report". FCC. http://www.fcc.gov/Daily_Releases/Daily_Business/2010/db0720/FCC-10-129A1.pdf. Retrieved July 23, 2010).

The trend is to raise the threshold of the broadband definition as the marketplace rolls out faster services. Data rates are defined in terms of maximum download because several common consumer broadband technologies such as ADSL are "asymmetric"—supporting much slower maximum upload data rate than download. "Broadband penetration" is now treated as a key economic indicator. ("Birth of Broadband". ITU. http://www.itu.int/osg/spu/publications/birthofbroadband/faq.html. Retrieved July 21, 2009). ("OECD Broadband Report Questioned". Website Optimization. http://www.websiteoptimization.com/bw/0705/. Retrieved June 6, 2009.)

2.1.19 Cloud Provisioning - SaaS Framework

The Cordys Business Operations Platform (BOP) is web-based and fully SaaS (Software-as-a-Service) enabled, with no client implementation requirements other than a web browser. The Cordys SaaS Deployment Framework enables companies to mix and match legacy software with services available in the Cloud or to launch new web-based services to customers.
(http://www.cordys.com/cordyscms_com/cloud_provisioning.php)

2.1.20 Dynamic cloud provisioning

The SaaS framework creates new application usage patterns, in which users may add applications and services on the fly, new SaaS consumers (tenants) may join or leave on a daily basis and customers are billed per usage. To address the requirements of these usage scenarios, Cordys provides the SaaS Deployment Framework (SDF) which enables dynamic cloud provisioning of applications and rapid on-boarding of tenants; manages the relationships between applications, tenants and users; and measures (meters) utilization of billable entities.
(http://www.cordys.com/cordyscms_com/cloud_provisioning.php)

2.1.21 Multi-tenant SaaS framework

The concept of self-service is fundamental to the Cordys SaaS Deployment Framework. With Cordys, no direct contact is necessary between the SaaS service provider and the administrators of each individual tenant. Cordys enables delegation of administrative responsibilities from the SaaS provider to the tenant itself, by providing tenants with complete control over the provisioning aspects of the platform.

(http://www.cordys.com/cordyscms_com/cloud_provisioning.php)

2.1.22 SaaS Framework

A critical success factor of your SaaS application is its architecture. The SaaS application delivery is a single instance, one-to-many model which is supported by a multi-tenant architecture. Migrating from a traditional, isolated software delivery model to a SaaS model requires a paradigm shift in architectural design and application development methodology. Sabre's SaaS Framework consists of a robust, layered architecture built on Java/JEE and Open Source Technologies.

The Framework is geared towards enabling rapid SaaS application development while reducing development cost and easily embeds all critical SaaS architectural elements.
(http://www.free-press-release.com/news-techcello-recently-launched-ver-2-0-of-cellosaas-1290517154.html)

2.1.23 Metadata Services

The Metadata services of the Framework enables configuration of the application to suit each tenant's specific requirements. It will accommodate customization of the user interfaces, configuration of workflows, data model extensions and access controls. (http://metadata.library.cornell.edu/)

2.1.24 Presentation Layer

Web applications with rich user interfaces can be built on the presentation layer.
(http://ntrg.cs.tcd.ie/undergrad/4ba2/presentation/)

2.1.25 Business Layer

Highly specific business logics, workflows will be built in. Essential components of a SaaS application such as Metering, Billing, monitoring and security is addressed through this layer.
(http://www.harborobjects.com/AllenBerezovsky/post/2009/09/24/Business-Logic-in-Stored-Procedures-or-Business-Layer.aspx)

2.1.26 Cisco Security Management Suite

The Cisco® Security Management Suite is a framework of next-generation security management tools designed for the operational management and policy administration of the Cisco Self-Defending Network. This suite of integrated applications simplifies the management process by automating tasks associated with the functional areas of security management: configuration, monitoring, analysis, mitigation, identity, and auditing. The result is an increased level of security assurance, better organizational productivity, and lower overall TCO.

The primary components of the Cisco Security Management Suite include the Cisco Security Manager and the Cisco Security Monitoring, Analysis, and Response System (MARS).
(http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns171/ns625/ns647/net_brochure0900aecd80400060.html)

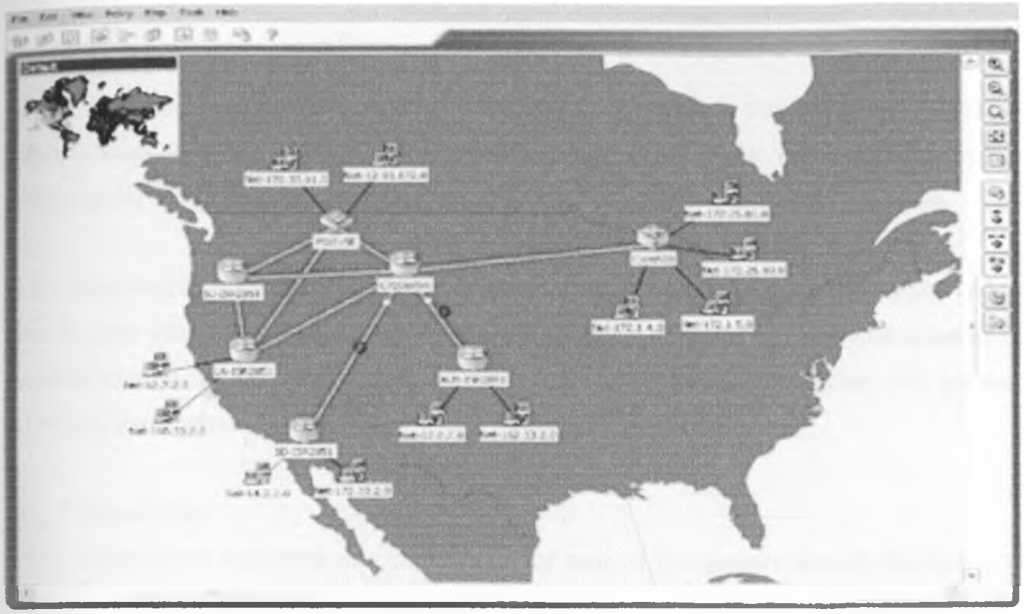

FIG 4: CISCO® SECURITY MANAGEMENT SUITE

FIG 5: SECURITY IN A CRITICAL CONDITION

2.1.27 **Introducing Practical Threat Analysis (PTA)**

The PTA process and companion software tool (PTA Professional) enable effective analysis and remediation of operational and security risks in complex software systems by an existing team. It provides an easy way to maintain dynamic threat models that are capable of reacting to changes in system's assets and vulnerabilities. With PTA an analyst can maintain a growing database of threats, create documentation for security reviews and produce reports showing the importance of various threats and the priorities of the corresponding countermeasures.

PTA automatically recalculates threats and countermeasures priorities and provides decision-makers with updated action item lists that reflect the changes in threat realities. Countermeasure priorities are expressed as a function of system's assets values, degrees of damage, threat probabilities and degrees of mitigation provided by countermeasures to the threats.

A software development team should use PTA from day one of design and throughout the system lifecycle. PTA provides intuitive and easy ways for iterative interaction between threat analysts and developers. It

supports a collaborative process of evaluating threat risks and ranking the cost-effectiveness of proposed countermeasures. The team's "threat analyst" can be the program/product manager, system architect or development lead who can start being productive with the CASE tool within hours.
How does PTA relate to security standards such as HIPAA, PCI DSS, ISO 27001, FIPS 199, COBIT and others?

Some standards (such as HIPAA via the RMF (risk management framework) NIST SP-800-66 Rev. 1 require a top-down risk analysis. Other standards (such as ISO 27001) prescribe a set of countermeasures (or controls) without delving into the underlying vulnerabilities or considering asset value.

PTA complements such standards by supplying the means for improving an organization's understanding of the complex interaction between threats, vulnerabilities and proposed countermeasures and taking the right purchasing and implementation decisions and defining a common terminology and process for threat modeling and analysis by PTA.

a) **Vulnerability**
This is a weakness, limitation or a defect in one or more of the system's elements that can be exploited to disrupt the normal functionality of the system. The weakness or defect may be either in specific areas of the system, its layout, its users, operators, and/or in its associated regulations, operational and business procedures.

b) **Countermeasure**
This is a procedure, action or mean for mitigating a specific vulnerability. A specific countermeasure may mitigate several different vulnerabilities. In some standards documentation, countermeasures are called "controls" or "safeguards".

c) **Threat**
This is a specific scenario of a sequence of actions that exploits a set of vulnerabilities and may cause damage to one or more of the system's assets.

d) **Threat's Risk** is a quantified measure of the likelihood of loss and/or damage that may be caused to one or more of the system's assets due to the specific threat. In some documentation the threat's risk is called "Annual Loss Expectancy" (ALE).

e) **Threat's Recommended Countermeasures** is a set of all the possible countermeasures that reduce the threat's risk. This set is based on the countermeasures that mitigate the threat's vulnerabilities.

f) **Threat's Actual Countermeasures** (AKA Threat's Mitigation Plan) is a subset of threat's recommended countermeasures that is assumed to be the most effective for mitigating a specific threat. The decision which of the recommended countermeasures will be included in the Threat's Mitigation Plan is made by the analyst, who uses his expertise to decide which countermeasures are most effective when applied together.

g) **Countermeasure's Cost** is the financial value that is associated with the implementation of a specific countermeasure.

h) **Countermeasure Cost-Effectiveness** is the degree of mitigation introduced by a specific countermeasure to the overall risk in the system in relation with the cost of implementing this specific countermeasure.

i) **Attacker** is a person (or group of persons) that may perform the steps of a specific threat scenario.

j) **Attacker Types** are the various classes of attackers that are differentiated according to their motivation, qualification, available attack tools and their accessibility to the attacked system's resources.

k) **Entry Points** are the "doors", either in the system itself or in the human operation associated with it, that are used by attackers to penetrate the system, e.g. Web site, IVR service, SMS server, CRM representatives called by customers over the phone etc.

l) **Area Tags** are descriptive tags that are relevant to assets, threats, vulnerabilities and countermeasures. Classifying the various security entities in the threat model according to their areas improves the readability of complex threat models. The following we describe the sub-steps of building a threat scenario and mitigation plan for a single threat.

i. The output of the PTA process
ii. The PTA process provides a number of reports and management level information's.
iii. List of system's threats sorted by their risk
iv. List of system's threats sorted by the financial damage they cause
v. List of individual countermeasures sorted by their overall risk mitigation effect
vi. List of countermeasures sorted by their cost effectiveness (mitigation divided by implementation cost)
vii. Maximal financial risk caused to each asset by existing threats
viii. Maximal financial risk caused to each asset by existing threats after all mitigation plans are implemented
ix. Maximal financial risk caused to each asset by existing threats after partial implementation of mitigation plans (use the 'already implemented' flag in countermeasures)
x. Total financial risk including all assets
xi. Total financial risk after all mitigation plans are implemented
xii. Total financial risk after partial implementation of mitigation plans

Reviewing these results may help the analyst in improving the threat model and in refining the parameters of the entities. It is most productive to check how the model behaves in response to changes in the input data and running various "what if" scenarios since this provide additional insight of the systems' realities. (http://www.software.co.il/application-security/26-practical-threat-analysis-of-complex-systems.html)

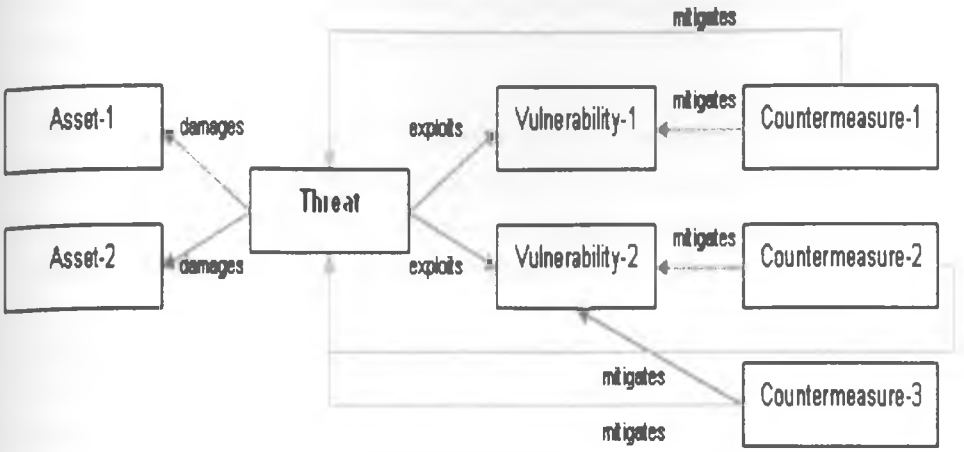

FIG 6: PRACTICAL THREAT ANALYSIS

2.1.28 Symantec™ Cyber Threat Analysis Program

One of the greatest challenges facing security organizations today is the ability to efficiently and cost effectively integrate products and services within existing network infrastructures, without disrupting established policies and procedures. The current economic downturn has also put additional emphasis on the cost of integration and the ability to leverage past investments while delivering enhanced security.

Cyber security threats aimed at corporations and government organizations arrive faster and are more sophisticated than ever before. Cyber incident identification, analysis, and response are often limited by an organization's view into the global threat landscape and a lack of validated cyber intelligence to substantiate an effective response.

The products, services, and resources an organization employs to defend its assets can exacerbate this. While a security professional's intent is to create both a proactive and reactive defense posture to mitigate cyber risk, their organization may lack access to the critical components and the intelligence necessary to meet this requirement. As threats evolve, even the intended functionality or configuration of a solution can add to this problem. These limitations can create a false sense of security and lead to compromises, which ultimately increase an organization's risk.

According to many independent sources, the sophistication, complexity and targeted nature of these attacks will continue to grow. Trying to determine where the next attack will come from is a daunting challenge and without effective intelligence, it can be a futile effort.

The Symantec Cyber Threat Analysis Program (CTAP) mitigates cyber risk with a comprehensive approach to threat identification, intelligence gathering and validation, and response to protect critical client

information. The result is a highly customized solution that integrates multiple components that address the specific security requirements of customers.

CTAP is designed to integrate with your existing IT environment. This provides the opportunity to consolidate resources and introduce heightened security awareness while enhancing workflow that creates a more secure enterprise security model. The CTAP approach leverages and extends your current investments in both technology and human capital.

2.1.28.1 Delivering Effective Security Protection

The ability to implement a comprehensive threat mitigation strategy within any organization requires the correlation of data across disparate networks within the infrastructure. This correlation must incorporate credible and validated external datasets that identify threats while ensuring the integrity of the information for a decisive and efficient threat mitigation response. However, without skilled cyber analysts to integrate data from multiple sources and interpret findings, effective mitigation can be diminished.

The Symantec Cyber Threat Analysis Program provides customers with access to an enhanced view of the threat landscape supported by onsite subject matter experts and specialized tools to enact response. CTAP analysts leverage the SymantecTM Global Intelligence Network as well as the same proprietary tools and Symantec™ Cyber Threat Analysis Program

2.1.28.2 Program Overview

Infrastructure that Symantec uses internally to detect, block and remove threats, resulting in increased protection for client networks and assets around the world. CTAP analysts enhance situational awareness by creating a client-specific contextual landscape that is formed by integrating client-specific incident data, Symantec's global vulnerability and threat intelligence data, and open source data. This unique insight is then augmented with access to Symantec's internal proprietary tools and specialized infrastructure to enhance response capabilities. This correlation ultimately assists organizations in their understanding of both strategic and tactical cyber threats to create a comprehensive mitigation strategy to defend its networks.

Work can be performed in a secure environment, and no customer data is removed, unless specifically directed by the client. Proactive methodologies assist in delivering mitigation strategies to protect against today's sophisticated threats – both existing and new, unseen threats. The result is an enhanced protection from your network and technology investments. CTAP analysts have access to a wide range of detection technologies such as the Symantec distributed honeypot, spam, underground, P2P, and crawler networks. These automated technologies analyze system behaviors and network communications to detect and actively block threats.

a) Validate security threats

i. Leveraging customer investment in existing detection technologies
ii. Help identify attacks that may have already occurred
iii. Provide incident analysis and briefings on threat dispositions
iv. Identify groups of bad actors
v. Identify persistent methods and procedures of criminals and other cyber attackers
vi. Block malicious behavior
vii. Provide proactive threat analysis
viii. Prevent attacks before they happen
ix. Harden the infrastructure to make it more difficult for attackers to get in, thus Increasing the cost to attackers so that they give up and go elsewhere
x. Remove the threat
xi. Provide countermeasure support and implementation guidance
xii. Support a decision matrix that outlines courses of action to defend against attacks
xiii. Counter electronic espionage through data leakage remediation

This comprehensive program allows you to proactively define threats and take decisive action. CTAP ensures the confidentiality, integrity and availability of your networks, but also ultimately improves your situational awareness within the threat landscape in the face of critical events.

b) Sophisticated Tools Deliver Enhanced Analysis Capabilities

Symantec CTAP analysts leverage sophisticated tools via a proprietary remote interface with embedded tool sets and internal systems that enable Symantec CTAP analysts to quickly identify emerging cyber security threats, develop countermeasures, and enact solutions. By integrating client data and public data with Symantec Cyber Threat Analysis Program intelligence, Symantec can help determine the true nature of an attack and/or the required mitigation steps. CTAP analysts have secure access to Symantec's proprietary cyber intelligence catalogs, research facilities, proprietary tools, and human capital, as well as the greater CTAP community on behalf of our clients.

Examples include:

i. Attack, vulnerability, and malicious code intelligence
ii. Phishing, spam, data leakage, spyware, adware, and virus intelligence
iii. Underground cyber economy and honey pot network intelligence
iv. Symantec's internal Infrastructure and toolsets
v. Subject matter experts
vi. Related analysis and cyber-specific reports
vii. Client Confidentiality and Sensitivity

Symantec recognizes the specific requirements expected by our clients in the delivery of CTAP services. Symantec offers a unique and special relationship with clients who implement CTAP services, which differ

from traditional client/vendor relationships. CTAP analysts reside within the client's domain and work as part of their own teams using the Symantec internal tools and data to deliver an effective security platform. While our clients are happy to have this level of partnership, they are not necessarily open to advertising this. Symantec understands this and is very conscious in protecting and ensuring the confidentiality of our CTAP clients.

The work we perform is sensitive and so are the environments we work in. CTAP analysts are part of a community of CTAP analysts and are trained to respect each client's confidentiality. They do not share client data or information, but they do help propagate better mitigation strategies across the team. We are committed to protecting our clients in every way.
(http://eval.symantec.com/mktginfo/enterprise/white_papers/bsymc_cyber_threat_analysis_program_WP_250478.en-us.pdf)

FIG 7: SYMANTEC CYBER THREAT ANALYSIS PROGRAM REMOTE INTERFACE

2.1.29 The Purewire Web Security Service

The Purewire Web Security Service is a security software-as-a-service (SaaS) that provides unmatched protection against malicious people, places and things on the Web™. By integrating sophisticated technology and intelligent analysis, Purewire brings the highest levels of security, performance and control to Web users and brings trust back to the Web.

The Purewire Web Security Service is deployed as a SaaS-based secure Web gateway that sits between a company's network and the Internet to protect the users as they surf the Web it:

i. Inspects outbound Web traffic for safety and compliance
ii. Analyzes Web site response traffic for malicious programs and untrustworthy users
iii. Provides global visibility through comprehensive and flexible reporting

Fig 8: Purewire web security service

2.2 Maturity levels

SaaS architectures can generally be classified as being at one of four "maturity levels", whose key attributes are configurability, multi-tenant efficiency, and scalability. (Wainewright, Phil (October 2007). "Workstream prefers virtualization to multi-tenancy". http://blogs.zdnet.com/SAAS/?p=400. Retrieved 2008-05-24). Each level is distinguished from the previous one by the addition of one of those three attributes:

2.2.1 Level 1 - Ad-Hoc/Custom:

At the first level of maturity, each customer has its own customized version of the hosted application and runs its own instance of the application on the host's servers. Migrating a traditional non-networked or client-server application to this level of SaaS typically requires the least development effort and reduces operating costs by consolidating server hardware and administration.

2.2.2 Level 2 - Configurable:

The second maturity-level provides greater program flexibility through configurable Meta data, so that many customers can use separate instances of the same application code. This allows the vendor to meet the different needs of each customer through detailed configuration options, while simplifying maintenance and updating of a common code base.

2.2.3 Level 3 - Configurable, Multi-Tenant-Efficient:

The third maturity level adds multi-tenancy to the second level, so that a single program instance serves all customers. This approach enables more efficient use of server resources without any apparent difference to the end user, but ultimately comes up against limits in scalability.

2.2.4 Level 4 - Scalable, Configurable, Multi-Tenant-Efficient:

The fourth and final SaaS maturity level adds scalability through multitier architecture supporting a load-balanced farm of identical application instances, running on a variable number of servers. The provider can increase or decrease the system's capacity to match demand by adding or removing servers, without the need for any further alteration of applications software architecture.

SaaS architectures may also use virtualization, either in addition to multi-tenancy, or in place of it. (Wainewright, Phil (October 2007). "Workstream prefers virtualization to multi-tenancy". http://blogs.zdnet.com/SAAS/?p=400. Retrieved 2008-05-24).

2.3 The research location

This research falls in the domain of Network design application and security.

2.4 The Evolution of Application Delivery

2.4.1 Traditional on Premise Installed Application

i. On premise hardware, server, networks, database, backup provisioning at customer
ii. Ongoing maintenance and management performed by customer
iii. Customer is responsible for providing logical and physical security
iv. Typically lengthy rollout/update cycles

2.4.2 Managed Service Application

i. Applications installed, managed and maintained by a third party
ii. High involvement of human resources for application management
iii. Installation on customer premise or also centralized model
iv. Most applicable for highly specialized applications
v. Typically single tenant (dedicated systems) off-the-shelf applications

2.4.3 Software as a Service (SaaS)

i. Typically no (or very limited) on-premise hardware, server, database, backup
ii. SaaS vendor provides all maintenance, management, and infrastructure
iii. Application usage via browser
iv. Economy of scale due to full automation from the provider
v. Fast rollout and innovation/update cycles
vi. Multi tenant architecture (multiple customers share some or all layers of the stack

2.5 SaaS Deployment Models

2.5.1 Pure SaaS application

i. Web browser is single interface point with customer
ii. All intelligence is centralized at SaaS provider
iii. Limited integration between customer and SaaS provider
iv. Examples: CRM, Email filtering, Payroll, Customer support applications

2.5.2 SaaS with customer side software agent

i. Web browser is interface point with customer
ii. Additional small client-side software agent (permanent or transient)
iii. Enables stronger integration of customer systems and SaaS service
iv. Examples: Application sharing, Web-filtering, Online-Backup

2.5.3 SaaS with customer side appliance

The Web browser is interface point with customer the additional hardware appliances are remotely managed on customer premise. This enables deep integration of customer systems with SaaS providers. Examples: Intrusion Detection, Security Management and Vulnerability Assessment

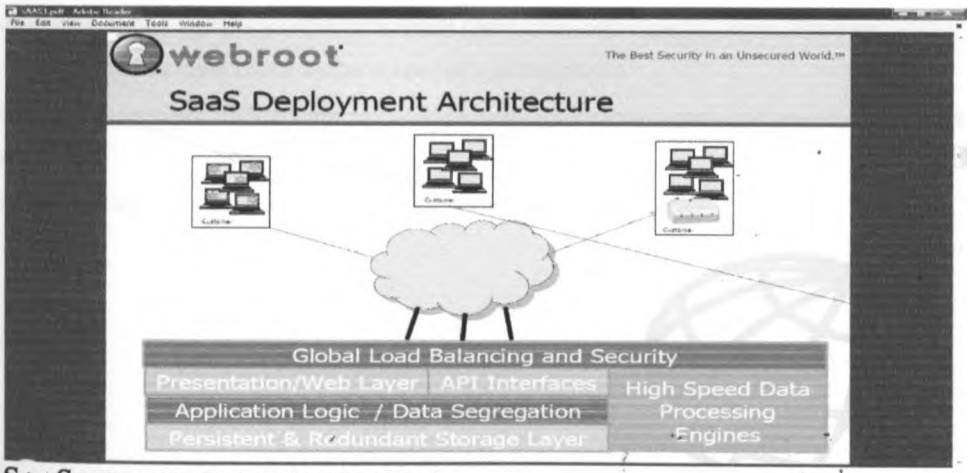

FIG 8: SAAS DEPLOYMENT ARCHITECTURE

2.6 Why Security SaaS makes Sense

Subscription model SaaS provides the pay as you go, per user, per time making SaaS services cost effective. There is also reduced risk in terms of performance, uptime, reliability and scalability, lower rollout cost, there is no additional IT overhead, improved security, rapid deployment and implementation and compliance requirements e.g. audit trails, archiving, and logging and hence allow focusing on core business.

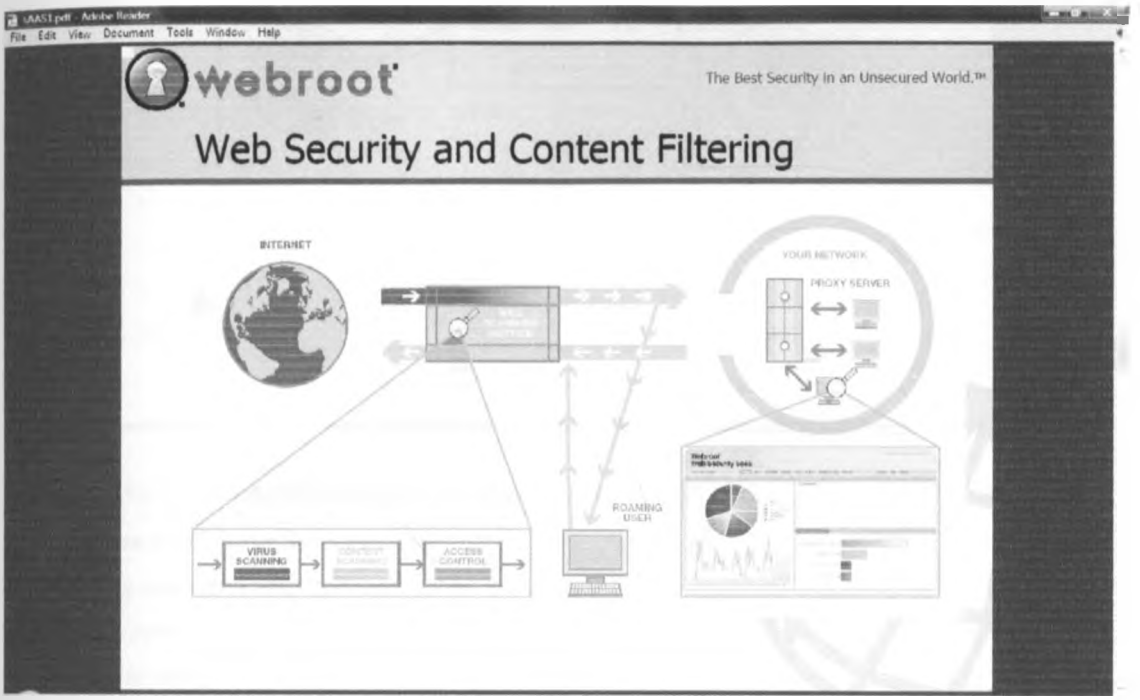

FIG 9: WEB SECURITY AND CONTENT FILTERING

Challenge: Sharp increase in spam and email-born malware

(High network bandwidth utilization)

a) Multi-level engine approach required to catch spam, zero-day exploits, virus, Spyware
b) Moving protection layer closer to the source of spam/malware
c) Transparently analyze inbound and outbound emails for content, spam and malware as well as content leaks
d) Email archiving to satisfy compliance requirements

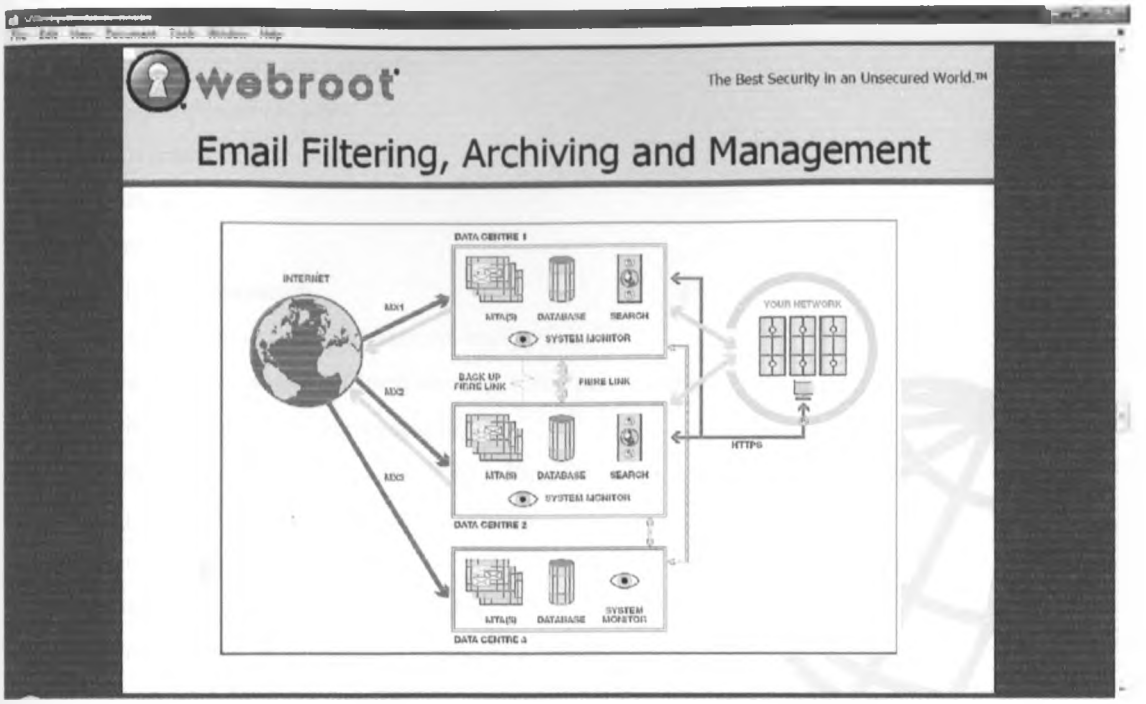

FIG 10: EMAIL FILTERING, ARCHIVING AND MANAGEMENT

Ten question to ask your SaaS provider

1. Existing customers and deployment size
2. Service renewal rates
3. Financial strength
4. Availability of integration capabilities
5. Ability to customize the SaaS application
6. Global data center footprint
7. Availability of training
8. Quality and presence of customer support
9. Frequency of major and minor updates
10. Service level commitments and actual performance

2.7 Vulnerability and Compliance Management

a) Challenge: Requirement for third party security assessment and compliance management

i. Discovery, assessment, and prioritization
ii. Internal and external view
iii. Validation against security policies
iv. Remediation tracking

b) Service Levels

i. Service availability and reliability
ii. Latency and performance
iii. Effectiveness
iv. Accuracy
v. Security

c) Security Considerations

i. Data storage model and architecture (encryption)
ii. User account management (provisioning, roles, permissions)
iii. Identity management (single-sign-on)
iv. Security process and certifications (SAS 70, ISO)
v. Backup, recovery, physical hosting facilities
vi. Business continuity

d) The Inside View from a SaaS Provider

i. Known platform provides better application quality
ii. Global deployment and distribution
iii. Simple application/revision management
iv. Load management and scale driven by business growth
v. Instant update for all customers
vi. Ability to scale quickly - unlimited scalability
vii. Integrate and deliver best-of-breed technologies
viii. Strong customer commitment and support is critical

2.8 Online Port scan

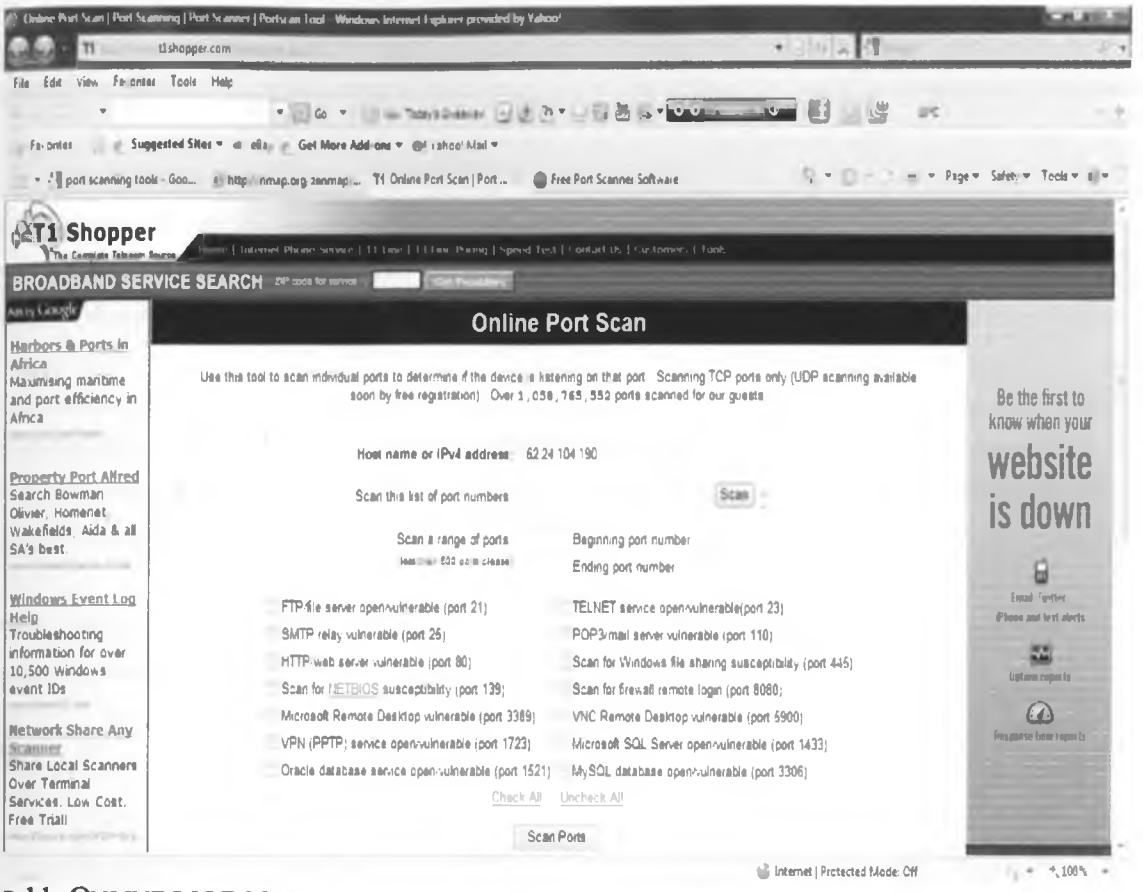

FIG 11: ONLINE PORT SCAN

(http://www.t1shopper.com/tools/port-scan/)

2.9 Nsauditor network security auditor

Nsauditor Network Security Auditor is a network security scanner that allows auditing and monitoring network computers for possible vulnerabilities, checks your network for all potential methods that a hacker might use to attack it. Nsauditor is a complete networking utilities package that includes more than 45 network tools for network auditing, scanning, monitoring and more. (www.nsauditor.com)

2.10 Advanced Port Scanner

Advanced Port Scanner is a small, fast, robust and easy-to-use port scanner for Win32 platform. It uses a multithread technique, so on fast machines you can scan ports very fast. Also, it contains descriptions for common ports, and can perform scans on predefined port ranges. (http://www.radmin.com/products/utilities/portscanner.php)

CHAPTER 3: METHODOLOGY

3.1 Introduction

The research was set up to explore ways of scanning for open ports and discovers vulnerabilities and suggested mitigation.

3.2 Context of the research

The conceptualized setup of the model consists of small business/startup company local area network with several web applications running on TCP/IP servers such as email server, company web site, enterprise/business applications, database servers, e-commerce applications.

The LAN is opened to public users on the Internet through the router with firewall setup to filter each application that can be accessed. The business systems are exposed to good users who intend to carry out normal business activities. The systems are also exposed to malicious users and applications whose intention is to disrupt normal usage of business service.

On the same front with normal and malicious user or applications, we have scan agents that can be launched to scan the URLs to discover open ports, applications and vulnerabilities facing them.

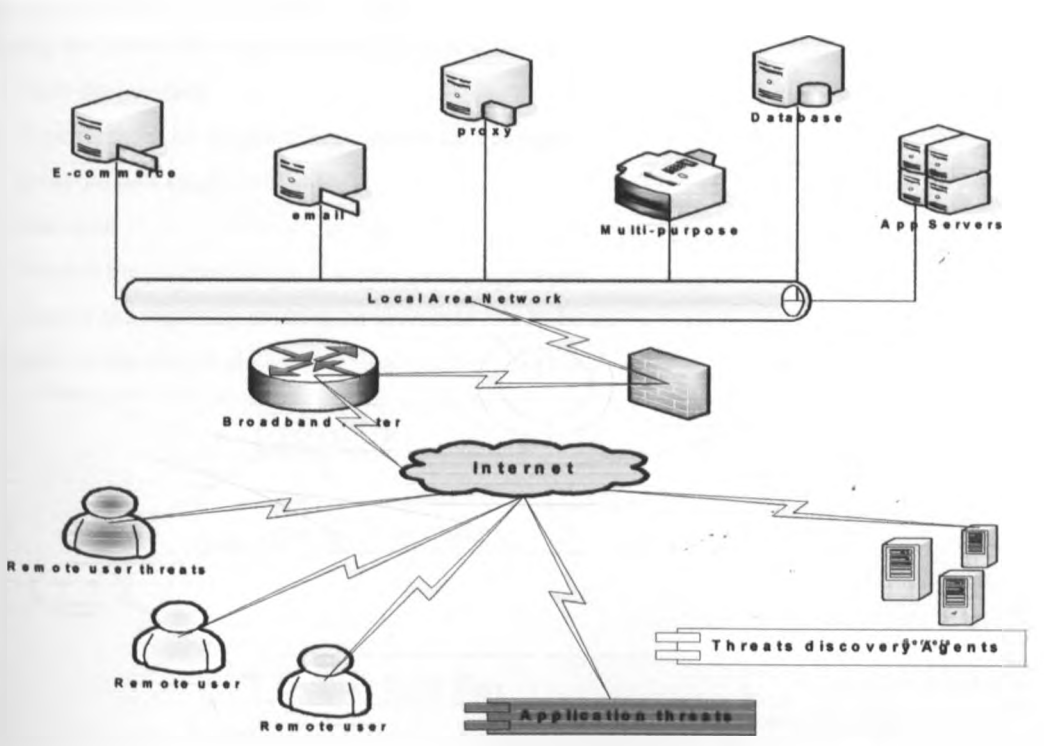

FIG 12: CONTEXT OF THE RESEARCH

3.3 Objectives

The objectives of the research are:

a) To test if some ports are open

b) To list possible vulnerabilities associated with such open ports

c) To suggest to users possible mitigation on the vulnerabilities found

3.4 Research setup

To be able to address the objectives, a system was designed with the following key features.

a) Port scanning

b) Display of general URL scan results

c) Mapping of open ports to possible vulnerability and suggestion of mitigation.

d) Display of vulnerabilities and Suggestion of mitigation to the users.

A user interface was designed to make it easy for the user to obtain and understand the results because of user friendly presentation.

3.5 How the research was conducted

Before the actual port scanning commenced, a hypothetical application server/web server was set up. Then, a broadband router and firewall were configured.

In using the system, the user is required to follow the following steps:

a) Open the interface

b) Type the URL of the site being scanned for open ports

c) Enter the port range for scanning

d) Start scan

e) Monitor the progress of the scan using the progress bar

f) Display the results by clicking on view scan results button

The order of the steps is shown graphically in fig 13 below

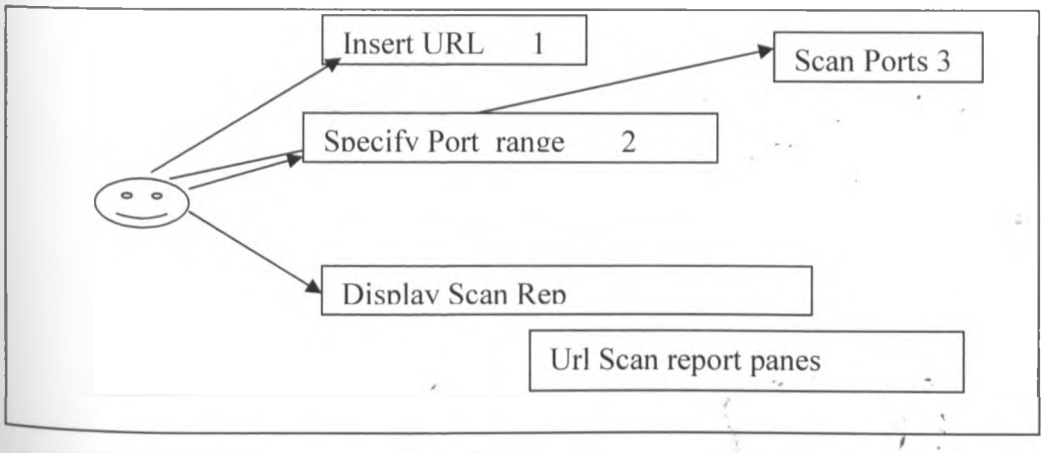

FIG 13: USER INTERFACE CASE

3.6 Data collection

Data collection was mainly done through scanning the ports of a given domain. The user provides the URL of the domain during the initialization of scanning.

The system was subjected to various types of websites:

i. E-commerce

ii. Educational web sites

From the scans, the data collected included:

a) Open ports

b) Closed ports

c) HTTP, MySql specific scans

3.7 Data analysis

The port number was used to determine whether a site is vulnerable or not. The vulnerability of website or domain depends on

a) Port Number.

b) Type of vulnerability associated with the open port.

CHAPTER 4: ANALYSIS, DESIGN AND IMPLEMENTATION

4.1 Introduction

This chapter outlines the Analysis, Design and Implementation of Port scanning. The chapter deals with preparations for actual implementation of the components. The discussion begins by analyzing the scanning environment that scans for open ports and asses the vulnerabilities associated with the open ports and suggested mitigation.

4.2 Scanning environment analysis

4.2.1 Httptest agent

The http scan agent scans the provided URL for:

i. /Images/ directory if it exists then it means it is possible for a hacker to view the contents of the directory.

ii. Use of Post or Get methods for sending forms data to a server. Use of get method is considered vulnerable than the post method.

This is displayed on the general URL test pane **(Appendix A-2)**.

4.2.2 MySQLTest agent

Mysql scan agent scans Mysql database to find out whether:

i. If it uses Mysql default password (root –root) -This makes it easy for hacker to gain access to the database to view Mysql database contents and possibly copy, delete or change its contents.

ii. No Password in Mysql database - Meaning the entry is free.

iii. Password is empty – an attempt to protect the database was initiated but no password was entered. **(Appendix A-3)**.

4.2.3 Portscann agent

The Portscann agent pick port numbers in the range provided by the user and sends them one at a time to another agent (Scanjob) **(Appendix A-4)**. The scanjob agent determines whether the port is open at the time of scan or not. If the port is open, the port number is written to the ports table and its status is flagged **yes**. Likewise if the port is closed or not in use its port number is written to ports table and its status is flagged **no**. **(Appendix A-5)**.

4.2.4 Features on the interface

The scan user interface is used to capture the scan parameters on the interface we have:

a) URL textbox – to enter the URL or Domain to be scanned

b) Start and stop textboxes – to specify start and stop port range to perform a scan on

c) Scan button – use to initiate the scan

d) Close button – use to exit the interface

e) Progress bar – Indicates that the scan is in progress

f) View Scanned Results button – after the scan completes the user uses this button to display results

g) Ports not in use pane – Displays the port numbers that are not in use

h) Ports in use pane – Displays the port numbers that are in use

i) General URL test pane – Displays the URL specific vulnerability checks

j) Vulnerability and suggested Mitigation pane – Displays vulnerabilities associated with the open ports and suggested mitigation

k) Textarea to display the port numbers ranges as shown below

i. The Well-Known Ports are those from 0 through 1023.

ii. The Registered Ports are those from 1024 through 49151.

iii. The Dynamic and/or Private Ports are those from 49152 through 65535

The user can enter short ranges or long ones depending on what ports they are interested in. a port range can be specified as follows

Start	Stop
0	500
500	3000
0	65535
49151	65535
80	80

Table 1. Port number range examples

4.2.5 Agent linking

The agents are linked together by the main interface frame1.java as shown in flowchart below.

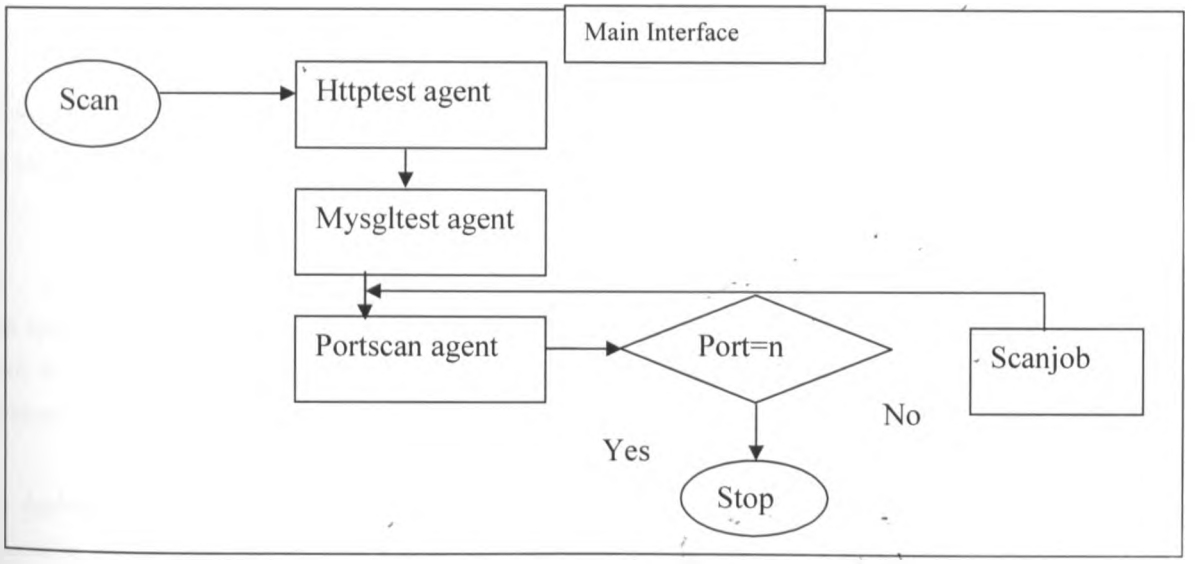

FIG 14: AGENT LINKING

4.3 Scanning environment design

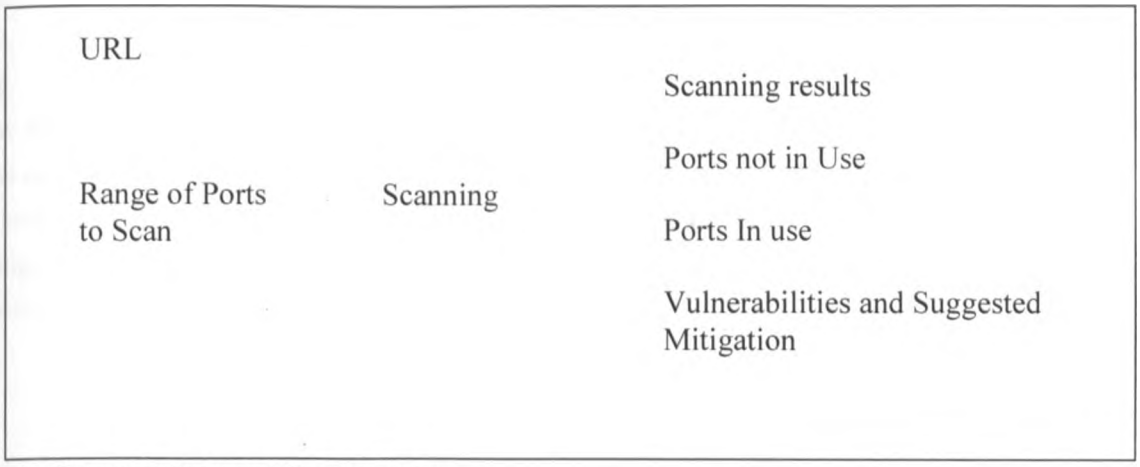

FIG 15: DESIGNED SCANNING ENVIRONMENT

4.4 Agent platform Design

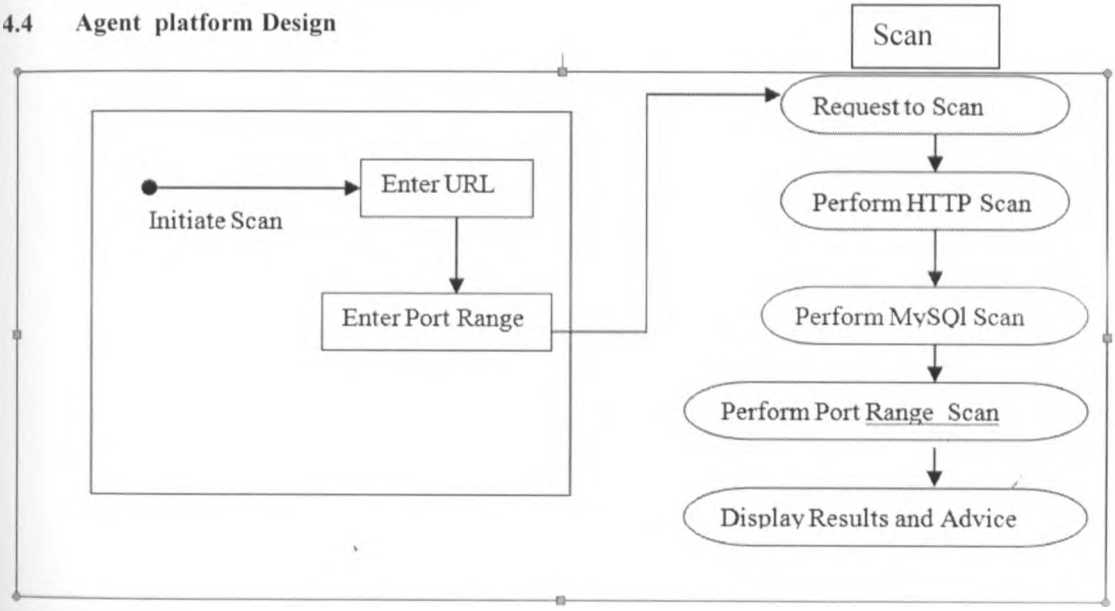

FIG 16: AGENT DESIGN PLATFORM

4.5 Deployment

This application will be deployed using the SaaS with customer side software agent model where desktop client is interface point with customer and additional small client-side software agent (permanent or transient) this enables stronger integration of customer systems and SaaS service.

The deployment model will be achieved by the personalized packing of the java desktop client as an executable jar file that can be launched on any operating system platform. To achieve this the host

computer will require java run time environment pre-installed and Internet connectivity to access to the public IP

4.6 Summary

The design of the Scanning environment was eased by the availability of the open source platform (Java). The main challenge in the running of the scanning environment was that of the availability of the network connectivity and the range with which the users specify port numbers to scan. Identifying and suggesting mitigation was successful since if a port was discovered open it was then mapped to vulnerability table to retrieve the vulnerability and suggested mitigation associated with port open.

CHAPTER 5: EXPERIMENTAL RESULTS

5.1 Introduction

In this chapter, illustration of how the scanning model achieves the key functions of discovery in the port scanning and Mitigation suggestions

The scan results are generated based on whether the port is open

The following were the results we obtained when the following URLs were subjected to the system.

Port number range 0-2000

5.2 General results

The extent to which an open port is deemed vulnerable depends on the open port and the vulnerability strength. The following results gave an overview of the number of ports open at an instance of a scan in given domain and the vulnerabilities associated with the ports.

5.2.1 Site A

When we subjected Site A to the system one port number (53) was discovered to be open and the following vulnerability and the system suggested mitigation is shown in table 2 below

Port open	Vulnerability	Mitigation
53	This could open the way for a Trojan that uses port 53 to bypass the firewall.	Define DNS server explicitly in the firewall configuration.

Table 2. Scanning Status of SITE A

From the above results Site A at the time of scan had only one open port this indicated that the URL was safe.

5.2.2 Site B

When we subjected Site B to the system four ports (110,143,21,53) were discovered to be open and the following vulnerabilities and the system suggested mitigation are shown in table 3 below

110	Remote users can gain privileged (root) access to systems running vulnerable versions of POP servers	If you determine that your POP server is vulnerable disable the POP server.
143	When this port is opened and exposed to the outside world can create serious vulnerabilities for the users PC.	If you cannot close this port, then use a NAT router or personal firewall.
21	An attacker could connect to port 21 and instead of sending expected data, one could	web-hosting websites, allow its customers to upload there files to manage there

	send something unexpected.	website. Its not really un-secure to be port 21 open,
53	This could open the way for a Trojan that uses port 53 to bypass the firewall	Define DNS server explicitly in the firewall configuration

Table 3. Scanning Status of SITE B

Scan for results Site B at that instance indicated that there were more open ports and hence an indication that the URL was vulnerable to threats.

5.2.3 Site C

When we subjected Site C to the system eight ports (110,143,53,993,995,443,465,1081) were discovered to be open and the following vulnerabilities and the system suggested mitigation (some of which were not in the table at the time of run) are shown in table 4 below

53	This could open the way for a Trojan that uses port 53 to bypass the firewall	Define DNS server explicitly in the firewall configuration
110	Remote users can gain privileged (root) access to systems running vulnerable versions of POP servers. No specific description of vulnerability	If you determine that your POP server is vulnerable disable the POP server
143	When this port is opened and exposed to the outside world can create serious vulnerabilities for the user's PC.	If you cannot close this port, then use a NAT router or personal firewall
993,995, 443, 465,1081	As per time of scan the vulnerabilities and suggested mitigation had not been put into our database	No suggestion

Table 4. Scanning Status SITE C

Scan results for Site C at that instance indicated that there were more open ports and hence an indication that the URL was more vulnerable.

5.3 Malicious Users

The best way to protect yourself is to find the open ports before an attacker finds them hence address the issue of potential malicious users who can scan your URLs for open ports and launch attacks.

CHAPTER 6: DISCUSSION

6.1 The main findings and observations

The results from simulations done with the scanning model developed during the research process in the previous chapter lead to the following main findings and observations:

a) Scanning URL was successful.

The model successfully achieved URL-request scanning and required services discovery. The ports could be scanned and the status determined as open or closed which are then written to a port table for further processing. The open port status was flagged YES, this was then used to retrieve vulnerability and Mitigation associated with the ports from Vulnerability table. Every time a scan is performed a new port table is created as long as a correct URL and start and stop port numbers are provided. The scan was successful when the URL provided was in connection and ports ranges to scan were provided. The scan fails if the URL is not in connection or the URL is using a proxy server.

b) Port range

The scanning process was faster with shorter range as opposed to longer ranges. The reduced performance is attributed to the computers used. No scanning would proceed if start port number or stop port number or both were missing.

c) View scanned results

There are four panes to display scan results. The scan results were available only when there was a successful scan as discussed in (a) above. Table 3 below shows what is displayed in each pane

Pane 1	Pane 2	Pane 3	Pane 4
Ports closed	Ports opened	General Tests	Vulnerabilities and Mitigation suggested

Table 5. Scanned results template

If the scan as described in (a) above was not successful then all the panes are blank. If no port is found open then the ports opened pane and vulnerability and mitigation panes are blank. A successful scan would always display results in pane 3 because the tests here are specific to HTTP and Mysql hence don't defend on open ports.

6.2 Exceptions

The scanning process cannot be successful if the URL is not in connection. From the research it can be concluded that if the connection uses a proxy then the scan is a challenge.

6.3 Relationship to previous work

This study is closely related to Nmap by Fyodor from (www.insecure.org). Nmap uses a variety of active probing techniques and changes the packet probe options to determine a host's operating system. Nmap offers its users the ability to randomize destination IPs and change the order of and timing between packets.

This functionality can obscure the port scanning activity and thus fool intrusion detection systems. Other port scanners include queso, checkos, and SS.

6.4 Achievements

The work demonstrates that it is possible to scan for open and closed ports in a given URL that is in connection. Once the scan process is complete it is possible to generate a list of vulnerabilities associated with the open ports and suggest mitigation.

This research project has been developed using open source software and as software as a service it can be used for free. The users need to download the product and follow some steps to install it in their computers.

6.5 Constraints

The test environment requires that machine used in carrying out the experiment be in a constant connection to the Internet. There is need to acquire a dedicated leased line or unlimited wireless broadband connections.

Some users may have problems in downloading and running the product meaning that a user intending to use this product must have some knowledge in java programming language.

CHAPTER 7: CONCLUSIONS AND FUTURE WORK

7.1 The conclusion

In section 1.4 of the research work, the key objectives were shown. The main aspects of the system developed and tested were to scan ports associated with a URL and establish their status. Based on the scanned port it was then possible to tell what vulnerabilities were associated with the URL's open port(s) and then suggest Mitigation to the discovered vulnerabilities.

7.2 Recommendations and Future Work

This research has focused on the scanning of URLs to establish which ports are open. The scanning and display of possible vulnerabilities in this research depend wholly on manual input, which forms a building block for future automated input. The recommendation for future work would be to develop agents to automatically generate input built to databases as subscription, list vulnerabilities and suggest mitigation.

Further research on effects/impacts of related ports on security can be conducted. There is need also to explore how ports that are required to be open for the users to perform certain activities are given attention in relation to vulnerabilities that are associated with them.

REFERENCES

1. Birth of Broadband. ITU. http://www.itu.int/osg/spu/publications/birthofbroadband/faq.html. Retrieved July 21, 2009.
2. Chong and Carraro, 2006.
3. http://eval.symantec.com/mktginfo/enterprise/white_papers/bsymc_cyber_threat_analysis_program_WP_250478.en-us.pdf
4. http://metadata.library.cornell.edu/
5. http://ntrg.cs.tcd.ie/undergrad/4ba2/presentation/
6. http://supplychaintechnology.wordpress.com
7. http://www.asl-webroot.co.uk/
8. http://www.binarysec.com
9. http://www.cisco.com/en/US/solutions/collateral/ns340/ns394/ns171/ns625/ns647/net_brochure0900aecd80400060.html
10. http://www.cordys.com/cordyscms_com/cloud_provisioning.php
11. http://www.free-press-release.com/news-techcello-recently-launched-ver-2-0-of-cellosaas-1290517154.html
12. http://www.f-secure.com
13. Fyodor http://www.insecure.org/nmap
14. http://www.harborobjects.com/AllenBerezovsky/post/2009/09/24/Business-Logic-in-Stored-Procedures-or-Business-Layer.aspx
15. http://www.iana.org/assignments/port-numbers
16. https://www.mcafee.com
17. www.nsauditor.com
18. http://www.radmin.com/products/utilities/portscanner.php
19. http://www.reboottwice.com
20. http://www.software.co.il/application-security/26-practical-threat-analysis-of-complex-systems.html
21. http://www.zscaler.com
22. OECD Broadband Report Questioned. Website Optimization. http://www.websiteoptimization.com/bw/0705/. Retrieved June 6, 2009.
23. OECD Broadband Statistics to December 2006. http://www.fcc.gov/cgb/broadband.html. Retrieved June 6, 2009.
24. Sixth Broadband Deployment Report. FCC. http://www.fcc.gov/Daily_Releases/Daily_Business/2010/db0720/FCC-10-129A1.pdf. Retrieved July 23, 2010
25. (http://www.t1shopper.com/tools/port-scan/)
26. Traudt, Erin; Amy Konary (June 2005). "2005 Software as a Service Taxonomy and Research Guide". IDC. pp. 7.

27. Wainewright, Phil (October 2007). "Workstream prefers virtualization to multi-tenancy". http://blogs.zdnet.com/SAAS/?p=400. Retrieved 2008-05-24

APPENDIX A

```
/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/
package rotich.agents;
import java.sql.*;
import java.awt.Color;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.Vector;
import javax.swing.*;
import java.awt.event.ActionListener;
import javax.swing.table.DefaultTableModel;
import java.awt.event.ActionEvent;
import java.util.Calendar;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.text.SimpleDateFormat;
import java.lang.*;
import java.util.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
//import java.beans.*;
import java.util.Random;
/**
 *
* @author Erick Rotich
*/
```

Appendix A-1

```
public class frame1 {
JFrame frame, frame1;    JButton scan, vulnerability, wellknownports, report2,scanExit,startButton;
JTextField url, start, stop;    JPanel panel;    JLabel url2, start2, use, nouse,pvul,mitiga,counter;    JLabel
stop2, scan2, ports, status;
```

```
JComboBox devices;
JList list, report, list1,report1,list2;
JTextArea list3;
public DefaultListModel model, model3, model1,model5,model6;
JScrollPane pane, pane2, pane3, pane4, pane5,pane6,scrollpane;
DefaultTableModel model2;
JTable table;
public int val1,val2;
JProgressBar progressBar;
JTextArea taskOutput;
public frame1() {
frame = new JFrame("                SCAN          AGENT");
scan = new JButton("scan");
scanExit = new JButton("Close");
vulnerability = new JButton("view vulnerability");
wellknownports = new JButton("view wellknown ports");
report2 = new JButton("View Scanned Results");
list3 = new JTextArea(5, 30);
JScrollPane scrollPane = new JScrollPane(list3);
list3.setEditable(false);
scrollPane = new JScrollPane(list3);
JTextArea list3 = new JTextArea(
"The port numbers are divided into three ranges. \n" +
" i. The Well-Known Ports  0 - 1023. \n" +
" ii. The Registered Ports 1024- 49151.\n" +
" iii. The Dynamic  Ports  49152 - 65535 "
);      // list3.setFont(ITALIC, 16));
list3.setLineWrap(true);      list3.setWrapStyleWord(true);      list3.setEditable(false);      url2 = new
JLabel(" ENTER THE URL");
start2 = new JLabel("start");
stop2 = new JLabel("stop");
scan2 = new JLabel("CLICK TO SCAN");
ports = new JLabel("enter the start and stop port range");
status = new JLabel();
counter = new JLabel();
use = new JLabel("Ports not in use");
nouse = new JLabel("ports in use");
```

```
pvul = new JLabel("General Url Tests");
mitiga = new JLabel("Vulnerabilities and Suggested Mitigation");
devices = new JComboBox();
model1 = new DefaultListModel();
list1 = new JList(model1);
pane4 = new JScrollPane(list1);
progressBar = new JProgressBar();
model = new DefaultListModel();
list = new JList(model);
pane = new JScrollPane(list);
model3 = new DefaultListModel();
model5 = new DefaultListModel();
report = new JList(model3);
report1 = new JList(model5);
pane3 = new JScrollPane(report);
pane5 = new JScrollPane(report1);
panel = new JPanel();
panel.setLayout(null);
url= new JTextField();
start = new JTextField();
stop = new JTextField();
url2.setBounds(20, 20, 200, 20);
url.setBounds(20, 60, 180, 20);
ports.setBounds(20, 80, 200, 40);
start.setBounds(20, 140, 70, 20);
start2.setBounds(20, 120, 70, 20);
stop2.setBounds(90, 120, 70, 20);
stop.setBounds(90, 140, 70, 20);
vulnerability.setBounds(20, 260, 140, 20);
report2.setBounds(20, 180, 200, 20);
pane.setBounds(20, 260, 100, 280);
pane3.setBounds(300, 260, 140, 280);
pane5.setBounds(500, 260, 650, 280);
pane4.setBounds(150, 260, 100, 280);
report2.setVisible(false);
scan.setBounds(600, 60, 120, 20);
list3.setBounds(220, 100, 320, 100);
```

```
scanExit.setBounds(600, 100, 120, 20);
progressBar.setBounds(600,140,200,20);
counter.setBounds(600, 160, 200, 20);
scan2.setBounds(600, 20, 200, 20);
status.setBounds(20, 200, 600, 20);
use.setBounds(20, 240, 180, 20);
nouse.setBounds(150, 240, 180, 20);
pvul.setBounds(300, 240, 180, 20);
mitiga.setBounds(500, 240, 300, 20);
progressBar.setStringPainted(true);
panel.add(scan);
panel.add(list3);
panel.add(scanExit);
panel.add(url);
panel.add(start);
panel.add(stop);
panel.add(url2);
panel.add(start2);
panel.add(stop2);
panel.add(scan2);
panel.add(ports);
panel.add(pane);
panel.add(pane3);
panel.add(pane5);
panel.add(status);
panel.add(pane4);
panel.add(use);
panel.add(nouse);
panel.add(pvul);
panel.add(mitiga);
panel.add(report2);
panel.add(progressBar);
panel.add(counter);
scan.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
try {
truncate();
```

```
truncate2();
} catch (Exception ex) {
ex.printStackTrace();
}
if (url.getText().equalsIgnoreCase(" ")) {
frame1 = new JFrame();
JOptionPane.showMessageDialog(frame1, "please insert url");
} else if (start.getText().equalsIgnoreCase("")) {
frame1 = new JFrame();
JOptionPane.showMessageDialog(frame1, "please insert the start port range");
} else if (stop.getText().equalsIgnoreCase(" ")) {
frame1 = new JFrame();
JOptionPane.showMessageDialog(frame1, "please insert the stop bit range");
} else {
try {
System.out.println("received a scan request for " + url.getText());
HTTPTest scan2 = new HTTPTest();
scan2.ddos(url.getText());
scan2.dirListing(url.getText());
MySQLTest sql = new MySQLTest();
sql.defaultPasswd(url.getText());
PortScann scan = new PortScann();
scan.portscanner(start.getText(), stop.getText(), url.getText(),progressBar, counter);
status.setText("scanned " + url.getText() + " from port range " + start.getText() + " to port range " +
stop.getText() + " as at " + now());
report2.setVisible(true);
} catch (Exception ex) {
ex.printStackTrace();
JOptionPane.showMessageDialog(frame1, "please check the url or netwok connection");
                    }
                }
        }
});
/*vulnerability.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
try {
vulnerability();
```

```
                        } catch (Exception es) {
es.printStackTrace();
                }
        }
});*/
scanExit.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
try {
System.exit(0);
                        } catch (Exception es) {
es.printStackTrace();
                }
        }
});
report2.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
try {
model.clear();
model1.clear();
model3.clear();
model5.clear();
retrieve(5);
retrieve();
retrieve3();
retrieve2();
retrieve4();
} catch (Exception es) {
es.printStackTrace();
                }
        }
});
wellknownports.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
try {
wellknownports();
} catch (Exception s) {
s.printStackTrace();
```

```
                }
            }
        });
}
public void vulnerability() throws Exception {
Vector<Vector<String>> data;
Vector<String> header;
Vector<Vector<String>> employeeVector = new Vector<Vector<String>>();
Class.forName("com.mysql.jdbc.Driver");
//MYSQLTest.default//java.DriverManager.getConnection("jdbc:mysql://localhost/scan_db",  "root",
"root");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from scan_db.vulnerabilities");
//populate vulnerability table
while (rs.next()) {
Vector<String> employee = new Vector<String>();
employee.add(rs.getString(1)); //id
employee.add(rs.getString(2)); //port
employee.add(rs.getString(3)); //service
employee.add(rs.getString(4)); //address
employeeVector.add(employee);
}

/*Close the connection after use (MUST)*/
if (con != null) {
con.close();
}
data = employeeVector;
//create header for the table
header = new Vector<String>();
header.add("tid"); //id
header.add("service"); // service
header.add("Port"); // port
header.add("test");
header.add("vul id");// vid
model2 = new DefaultTableModel(data, header);
```

```
table = new JTable(model2);
pane2 = new JScrollPane(table);
pane2.setBounds(260, 100, 500, 300);
panel.add(pane2);
}
public void wellknownports() throws Exception
{
Vector<Vector<String>> data; //used for data from database
Vector<String> header; //used to s
Vector<Vector<String>> employeeVector = new Vector<Vector<String>>();
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from scan_db.well_known_ports");
while (rs.next()) {
Vector<String> employee = new Vector<String>();
employee.add(rs.getString(1)); //id
employee.add(rs.getString(2)); //port
employee.add(rs.getString(3)); //service
employee.add(rs.getString(4)); //address
employeeVector.add(employee);
}
/*Close the connection after use (MUST)*/
if (con != null) {
con.close();
}
data = employeeVector;
//create header for the table
header = new Vector<String>();
header.add("tid"); //tid
header.add("port"); // port
header.add("service"); // service
header.add("host addr"); // host address
model2 = new DefaultTableModel(data, header);
table = new JTable(model2);
pane2 = new JScrollPane(table);
pane2.setBounds(260, 100, 500, 300);
```

```
panel.add(pane2);
}
public void truncate() throws Exception
{
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
stmt.executeUpdate("truncate scan_db.ports");
}
public void truncate2() throws Exception {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
stmt.executeUpdate("truncate scan_db.vulnerability");    }
public void retrieve5() throws Exception {
      Class.forName("com.mysql.jdbc.Driver");
      Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root",
"EK33771");
      Statement stmt = con.createStatement();
      ResultSet rs = stmt.executeQuery("select port from scan_db.ports where status = 'yes'");
     // Vector<String> employeeVector = new Vector<String>();
      while (rs.next()) {
            val2=rs.getInt(1); //id
             searchRecord2(val2);
       }
}
  public void searchRecord2(int val2) throws Exception {

          //if(txtNumber.getText() !="")
          {
      Class.forName("com.mysql.jdbc.Driver");
      //MYSQLTest.default//java.DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root",
"root");
      Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root",
"EK33771");

      Statement stmt = con.createStatement();
```

```
        stmt.executeUpdate("DELETE FROM ports where port=" +val2+ " and status ='no'");
stmt.close();

        }
}
public void retrieve() throws Exception {        Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select port from scan_db.ports where status = 'yes'");
Vector<String> employeeVector = new Vector<String>();
while (rs.next()) {
employeeVector.add(rs.getString(1)); //id
}
String w = employeeVector.toString();
String[] s = w.split(",");
for (int i = 0; i < s.length; i++)
{
model1.addElement(s[i]);
}
}
public void retrieve3() throws Exception {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select port from scan_db.ports where status = 'no'");
Vector<Vector<String>> employeeVector = new Vector<Vector<String>>();
while (rs.next()) {
Vector<String> employee = new Vector<String>();
employee.add(rs.getString(1)); //id
employeeVector.add(employee);
}
String w = employeeVector.toString();
String[] s = w.split(",");
for (int i = 0; i < s.length; i++) {
model.addElement(s[i]);
}
}
```

```
public void devices() throws Exception {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select device_id from scan_db.client_devices");
Vector<Vector<String>> employeeVector = new Vector<Vector<String>>();
while (rs.next()) {
Vector<String> employee = new Vector<String>();
employee.add(rs.getString(1)); //id
employeeVector.add(employee);
}

devices.addItem(employeeVector);
}
public void display() {
frame.setBackground(Color.green);
frame.setSize(1200,600);
frame.setLocation(20, 20);
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.add(panel);
frame.setVisible(true);
}
public String now() {
String DATE_FORMAT_NOW = "yyyy-MM-dd HH:mm:ss";
Calendar cal = Calendar.getInstance();
SimpleDateFormat sdf = new SimpleDateFormat(DATE_FORMAT_NOW);
return sdf.format(cal.getTime());
}
public void report(String v) throws Exception {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();          stmt.executeUpdate("insert into scan_db.vulnerability
(vulnerability) values('" + v + "')");
}
public void retrieve2() throws Exception {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
```

```
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from scan_db.vulnerability");
Vector<Vector<String>> employeeVector = new Vector<Vector<String>>();
while (rs.next()) {
Vector<String> employee = new Vector<String>();
employee.add(rs.getString(1)); //Empid
employeeVector.add(employee);
}
String w = employeeVector.toString();
String[] s = w.split(",");
for (int i = 0; i < s.length; i++) {
model3.addElement(s[i]);
}
}
public void retrieve4() throws Exception {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select port from scan_db.ports where status = 'yes'");
// Vector<String> employeeVector = new Vector<String>();
while (rs.next()) {
val1=rs.getInt(1); //id
searchRecord(val1);
}
}
public void searchRecord(int val1) throws Exception {
//if(txtNumber.getText() !="")
{
Class.forName("com.mysql.jdbc.Driver");
//MYSQLTest.default//java.DriverManager.getConnection("jdbc:mysql://localhost/scan_db",        "root",
"root");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("select * from myvulnerabilities where sport=" + val1 +"");
Vector<Vector<String>> employeeVector = new Vector<Vector<String>>();
while (rs.next()) {
Vector<String> employee = new Vector<String>();
```

```
employee.add(rs.getString(1)+ "  * Vulnerability *  " +rs.getString(2)+" * Mitigation * " +rs.getString(3));
//Empid
employeeVector.add(employee);
}
String w = employeeVector.toString();
String[] s = w.split(",");
for (int i = 0; i < s.length; i++) {
model5.addElement(s[i]);
}
stmt.close();
// }
// catch ( SQLException sqlex )
// {
// System.out.println( sqlex.toString() );
//  }
}
}
public static void main(String[] args) throws Exception {
frame1 frame = new frame1();
frame.display();
frame.devices();
System.out.println("built ");
}
}
/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/
package rotich.agents;
import java.net.*;
import java.io.*;
import java.util.logging.Level;
import java.util.logging.Logger;
/**
*
* @author Erick Rotich
*/
```

Appendix A-2

```
public class HTTPTest {
public static void dirListing(String url) throws Exception {
URL rconn = new URL("http:///" + url + "/images/");
BufferedReader in = new BufferedReader(new InputStreamReader(rconn.openStream()));
String inputLine;
if ((inputLine = in.readLine()) != null) {
System.out.println(inputLine);
System.out.println("prone to directory listing");
frame1 frame= new frame1();
frame.report("prone to directory listing");
}
else
{
System.out.println(" not prone to directory listing ");
frame1 frame = new frame1();
frame.report("not prone to directory listing");
}
in.close();
}
public static boolean ddos(String url) throws Exception {
URL rconn = new URL("http:///" + url);
BufferedReader in = new BufferedReader(new InputStreamReader(rconn.openStream()));
String inputLine;
String outputline;
while ((inputLine = in.readLine()) != null) {
if(inputLine.indexOf("method=\"post\"") != -1)
{
frame1 frame = new frame1();
frame.report("post method is used");
return true;
}
if(inputLine.indexOf("method=\"get\"") != -1)
{
frame1 frame = new frame1();
frame.report("get method is used");
return true;
```

```
}
}
in.close();
return false;
}
public static void main(String[] args)
{
try {
} catch (Exception ex) {
Logger.getLogger(HTTPTest.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/
package rotich.agents;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
/**
*
* @author Erick Rotich
*/
```

Appendix A-3

```
public class MySQLTest {
public static boolean defaultPasswd(String device) {
try {
Class.forName("com.mysql.jdbc.Driver");
Connection conn = DriverManager.getConnection("jdbc:mysql://" + device + ":3306/mysql", "root",
"EK33771");
return true;
}
catch (Exception expp)
{
try {
```

```
frame1 frame = new frame1();
frame.report("uses default MysQl");
} catch (Exception ex) {
Logger.getLogger(MySQLTest.class.getName()).log(Level.SEVERE, null, ex);
}
}
try {
Class.forName("com.mysql.jdbc.Driver");
Connection conn = DriverManager.getConnection("jdbc:mysql://" + device + ":3306/mysql", "root",
"EK33771");
frame1 frame = new frame1();
frame.report("No Password in MysQl DadaBase");
return true;
} catch (Exception expp) {
try {
frame1 frame = new frame1();
frame.report("Password is empty");
} catch (Exception ex) {
Logger.getLogger(MySQLTest.class.getName()).log(Level.SEVERE, null, ex);
}
}return false;
}
}
/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/
package rotich.agents;
import java.net.*;
import java.util.logging.Level;
import java.util.logging.Logger;
import java.sql.*;
import java.util.*;
import javax.swing.JLabel;
import javax.swing.JProgressBar;
```

Appendix A-4

```
public class PortScann {
```

```
JProgressBar pro;
JLabel label;
public void portscanner(String start, String stop, String s, JProgressBar progressbar, JLabel lab)
{
int startPortRange = 0 ;
int stopPortRange = 0;
this.pro =progressbar;
this.label = lab;
startPortRange = Integer.parseInt(start);
stopPortRange = Integer.parseInt(stop);
//String host = s ;
for (int i = startPortRange; i <= stopPortRange; i++)
{
ScanJob j = new ScanJob(i, s, pro,label);
j.start();
if(i%1000 == 0)
{
try {
Thread.sleep(500);
//pro.setVisible(false);
} catch (InterruptedException ex) {
Logger.getLogger(PortScann.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
}
public static void scanDump(String ha, String port)
{
try {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
stmt.executeUpdate("insert into well_known_ports(port_number, host_address) values('" + port + "', '" + ha
+ "')");
stmt.close();
con.close();
} catch (Exception ex) {
```

```
Logger.getLogger(PortScann.class.getName()).log(Level.SEVERE, null, ex);
}
}
}
/*
* To change this template, choose Tools | Templates
* and open the template in the editor.
*/
package rotich.agents;
import java.lang.reflect.InvocationTargetException;
import java.util.*;
import java.sql.*;
import java.net.*;
import java.util.logging.Level;
import java.util.*;
import java.lang.*;
import java.util.logging.Logger;
import javax.swing.JTextArea;
import javax.swing.*;
/**
*
* @author Erick Rotich
*/
```

Appendix A-5

```
public class ScanJob extends Thread {
int port;
String host = null;
JProgressBar prog;
JLabel lab1;
Runnable runner = new Runnable() {
public void run() {
int value = prog.getValue();
prog.setIndeterminate(true);
}
};
public ScanJob(int port, String host,JProgressBar jprogress, JLabel lab) {
this.port = port;
```

```
this.host = host;
this.prog = jprogress;
this.lab1 =lab;
//System.out.println(this.host);
}
public void run()
{
int min = prog.getMinimum();
int max= prog.getMaximum();
try {
SwingUtilities.invokeAndWait(runner);
} catch (InterruptedException ex) {
Logger.getLogger(ScanJob.class.getName()).log(Level.SEVERE, null, ex);
} catch (InvocationTargetException ex) {
Logger.getLogger(ScanJob.class.getName()).log(Level.SEVERE, null, ex);
}
try {
Socket ServerSok = new Socket(host, port);
String q = "";
q = Integer.toString(port);
scanDump(host, "" + port);
lab1.setIconTextGap(port);
dumpPorts("yes", port);
ServerSok.close();
}
        catch (Exception e) {
System.out.println("Port not in use: " + port);
lab1.setIconTextGap(port);
// report("Port not in use: " + port);
String q = "";
q = Integer.toString(port);
try {
dumpPorts("no", port);
} catch (Exception ex)
{
}
}
```

```
//prog.setVisible(false);
}
public void dumpPorts(String status, int port) throws Exception {
try {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
stmt.executeUpdate("insert into ports(port, status) values('" + port + "', '" + status + "')");
stmt.close();
con.close();
} catch (Exception ex) {
}
}
public void scanDump(String ha, String port) throws Exception {
try {
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/scan_db", "root", "root");
Statement stmt = con.createStatement();
stmt.executeUpdate("insert into well_known_ports(port_number, host_address) values('" + port + "', '" + ha
+ "')");
        stmt.close();
con.close();
prog.setVisible(false);
} catch (Exception ex) {
}
}
public void stopit()
{
prog.setVisible(false);
}
}
```

APPNDIX B – SIMPLE SCAN AGENT INSTALLATION GUIDE

a) Connect the machine to an internet connection that does not use a proxy server.

b) Copy the entire folder "ScanAgent" from the CD provided into a folder of your choice.

c) Open the folder and double click on run.bat batch file in the folder.

d) Enter the URL, start and stop port number range and start scan.

e) When the scan progress bar stops click on view scanned results button to view the scan results.