Reputation Scanner

Azeem J. Rios Soto
Master in Computer Science
Advisor: Jeffrey Duffany, Ph.D.
Electrical & Computer Engineering and Computer Science Department
Polytechnic University of Puerto Rico

Abstract *— The issue of security is paramount in any organization, especially one such as a bank. With the intention to improve bank security, a web application was developed. The application involves the scanning of uniform resource locator, showing the reputation of each uniform resource locator and sending notification via email with the result of the scanning. The administrator of the web application will add the different uniform resource locator that the bank wants to scan for reputation status, add the employee email which will be receiving the notification and creating the schedule of the scanning. The expectation from this project is to design an alternative web security system for the bank.*

Key Terms *– Cybersecurity, Malware, Reputation, Scanner.*

INTRODUCTION

The goal of this project is to design a web application for daily use on the information security department because web applications can be accessed through any web browser, no desktop installation or updates are required. The web base project system was developed using technologies like .Net, HTML, MS SQL, Selenium for web automation & virus total API.

The web application will scan different URL stored in a database. Using the virus total API, it will connect to the databases in the virus total site bringing the reputation of each URL (if it has been reported malicious or not) after the scan. Another process will start to do a search in the McAfee trusted source database. It will search for the reputation that was reported from the scanned URL. At the end of this process an email is sent to the analyst to review the results and take actions if needed. The bank lacked an automatic system which could have a list of URLs' and scan them quickly in different periods. To do this function, they had an analyst reviewing each of the URL's one by one in the McAfee Trusted Source database manually. This resulted in a large consumption of time and loss of efficiency. At the end of this project an automated system with a connection to a database to store the addresses and set the schedules to run the scan, was obtained. To evaluate the results the web application will send a final report by email to the security analyst.

The web application was developed using C#, the entity framework and the structure of Model View Controller. The HTLM used for the development is a free open source template that can be found online and was develop using Bootstrap framework. All the summary and tables information were connected on the HTML template. Also, the maintenance page for the data base chosen is Microsoft SQL because the company already uses that data base, so the connections of the tables and the database were done.

BACKGROUND

The information about Virus Total can be found in their webpage [1]:

"Virus Total stores all the analyses it performs, this allows users to search for reports given an MD5, SHA1, SHA256 or URL. Search responses return the latest scan performed on the resource of interest. Virus Total also allows you to search through the comments that users post on files and URLs, inspect our passive DNS data and retrieve threat intelligence details regarding domains and IP addresses. Fighting malware requires close collaboration. The overwhelming malware production rate, the growing

problem of false positives and the everlasting threat of false negatives cannot be counteracted without the determined engagement of all actors involved in end-user system security. Keeping this in mind, we have created Virus Total Community, a space where the antivirus industry and malware researcher can meet end-users to make internet a safer place. Virus Total Community allows you to rate and place comments on files and web sites".

Another component ion the web application is Trusted source. As explained in [2]:

"… is an Internet reputation system originally developed by Cipher Trust and now owned by Intel Security. It provides reputation scores for Internet identities, such as IP addresses, URLs, domains, and email/web content. Reputation data and content categories, as well as global email, web and other network traffic patterns observed by Trusted Source ecosystem, for any IP address, domain, or URL can be checked from the TrustedSource.org portal site. Trusted Source works by analyzing in real-time traffic patterns from email, web and network data flows from McAfee's global set of security appliances and hosted services, as well as those of partners like F5 Networks. Working off that data stream, it applies data mining and analysis techniques, such as Support Vector Machine, Random forest, and Term-Frequency Inverse-Document Frequency (TFIDF) classifiers to determine the degree of maliciousness and security risk associated with each Internet identity, as well as perform content categorization".

This tool also allows you to suggest an alternative categorization for a site. These requests will be addressed within an average of 3-5 business days with some requests requiring additional review and taking longer.

Selenium is explained by Nils in [3]:

"Selenium is a framework for the automated testing of web applications. Using Selenium, you can basically automate every task in your browser as if a real person were to execute the task. The interface used to send commands to the different browsers is called Selenium WebDriver. Implementations of this interface are available for every major browser, including Mozilla Firefox, Google Chrome and Internet Explorer".

One of Selenium's key highlights is the help for executing one's tests on numerous program stages. Selenium previously became deploy in 2004 when Jason Huggins was trying an inside application at ThoughtWorks. Being a brilliant person, he understood there were preferred employments of his time over physically venturing through similar tests with each change he made. He built up a Javascript library that could drive associations with the page, enabling him to naturally rerun tests against different programs.

Bootstrap is an open source front end improvement structure for the formation of sites and web applications. The Bootstrap system is based on HTML, CSS, and JavaScript (JS) to encourage the advancement of responsive, portable first locales and applications. Responsive structure makes it feasible for a site page or application to identify the guest's screen size and direction and consequently adjust the showcase likewise; the versatile first methodology accept that cell phones, tablets and assignment explicit portable applications are representatives' essential devices for completing work and addresses the prerequisites of those innovations in plan. Bootstrap incorporates UI parts, formats and Java Script devices alongside the system for usage. The product is accessible pre-gathered or as source code.

DESIGN

The design flowcharts and diagrams of how the application will work are display in this section. As explained in [4]:

"Software as a service is a model of software deployment where an application is hosted as a service provided to customers across the Internet. By eliminating the need to install and run the application on the customer's own computer, SaaS alleviates the customer's burden of software maintenance, ongoing operation, and support".

It reduces "up-front expense of software purchases through on-demand pricing [4]" for the

required services. For the web application logic see Figure 1.

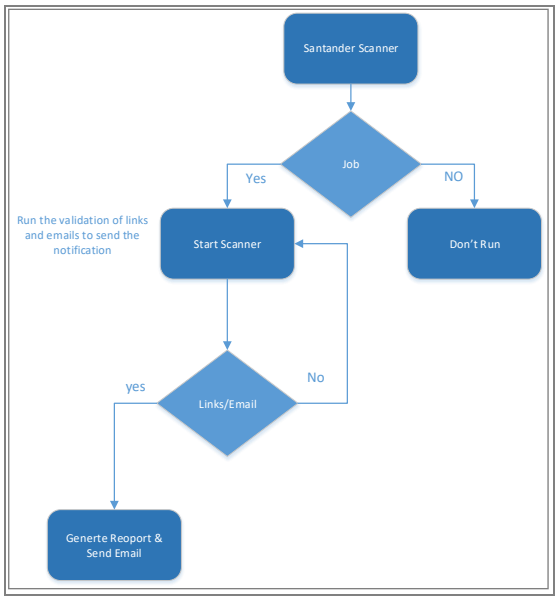

Figure 1
Flow Diagram

The diagram on Figure 2 is the infrastructure for the web application and its components.

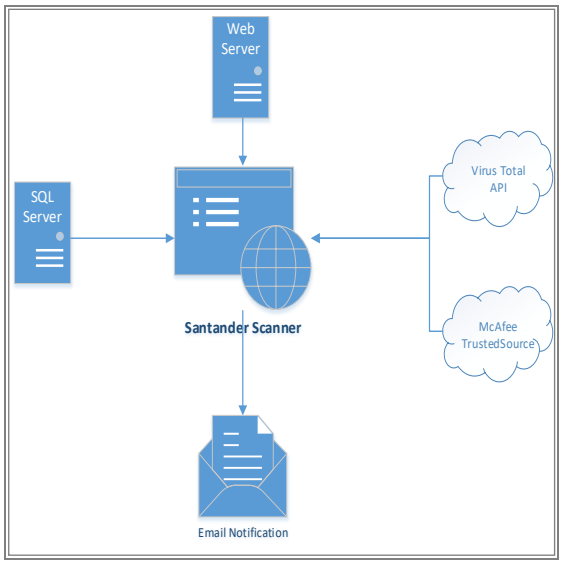

Figure 2
Infrastructure Design

The MSSQL tables describes the data types for each of the fields defined in all of the above described tables. It also defines the primary key and the foreign keys for each of those tables. Figure 3 shows the three tables that are created in the data base.

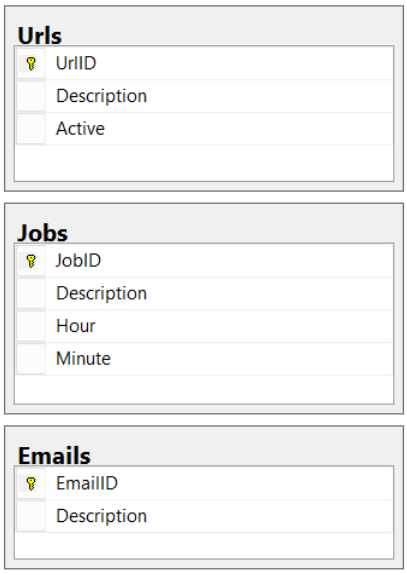

Figure 3
MSSQL Tables

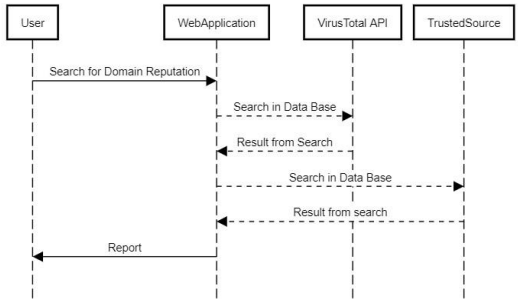

Figure 4
Sequence Diagram

A "significant workload shift [5]" will be achieved. "Local computers no longer must do all the heavy lifting when it comes to running applications [5]." "The network of computers … handles them instead. Hardware and software demand on the user's side decrease. The only thing the user's computer needs to be able to run is… [5]" the web browser interface software, which gives access to the website. Sequence diagram displays the sequence of interaction between objects participating in an action, which consists of the objects and the time at which they are interacting with each other. The time sequence diagram shows a clear picture of which object should be involved while developing a particular action. Figure 4 shows the time sequence diagram for the various actions of an employee while interacting with the system.

User Interphase

There are various software quality attributes that are taken into consideration:

- **Availability -** As scanner is a web-based service provided to the users, it will be available if server is up.
- **Interoperability -** The website is interoperable on various operating systems, thus increasing the applications usability and flexibility.
- **Maintainability -** The reputation scanner is a SaaS based service. Hence, all the updates are centralized making the maintenance convenient.

Figure 5 presents the scan result with the virus total API.

Figure 5
Virus Total Reputation Scan Result

Scan with trusted site result is demonstrated in Figure 6.

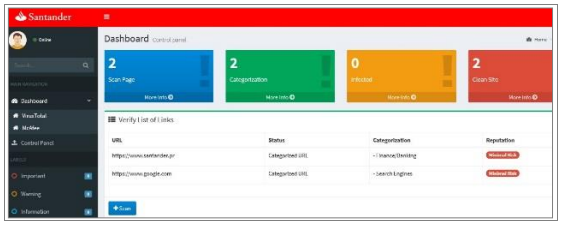

Figure 6
Reputation Scanner Result Page

Figure 7 demonstrate the maintenance web page.

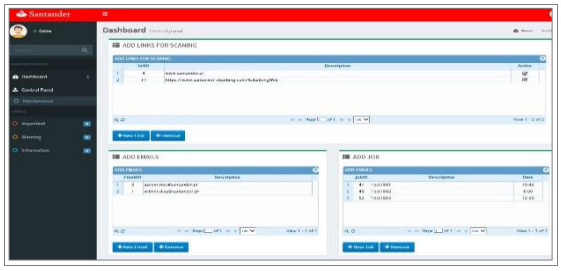

Figure 7
Maintenance Page

The email notification is sent after the scan with an attached document in .csv extension with a summary of the results as shown in Figure 8.

Figure 8
E-mail Notification

This application is designed as an object-oriented system for an Internet-based architecture using tow-layer architecture:

- **The Presentation layers:** This is the layer where the physical window and widget objects live. It will also contain Controller classes as in classical MVC. Any new user interface widgets developed for this application are put in this layer. In most cases this layer is completely developed within Grails.
- **The Data layer:** The data is managed by MySQL.

TEST PROCESS

The objective of testing was to show that the program leveled out contained bugs. Testing must not be mistaken for troubleshooting, which is the procedure of distinguishing and lessening the quantity of existing mistakes. Testing can never demonstrate that a code is a mistake, but instead check that blunders exist. Accordingly, need to consider the certainty that the mistake may originate from the test itself while the tried code may be right. Need to comprehend what the consequences of the test will appear before it has really been performed. The person who oversees doing the testing must, characterize what the result ought to be. If not, this will prompt bugs in either the program or the test or in both the program and the test. The beneficial thing

about testing is that it tends to be done with no earlier learning about the program.

For the development of this web application the requirements of the servers are minimal; windows server 2012 R2 with IIS 6 the role of ASP.Net and framework .Net 3.5 & .Net 4.5 need to running installed an 8 GB of RAM and HDD of 100 GB with a CPU of 4 Cores. In general, testing will only stop if the Web site Under Test (WUT) becomes unavailable. If testing is suspended due to the Web site becoming unavailable, testing will resume once access to the Web Site is reestablished. Certain individual test cases may be suspended, skipped or reduced if the prerequisite test has previously failed. Table I shows the results of tests made.

Table 1
User Test

ID	Actions	Expected Result	Result (Pass/Fail)
1	Reputation Scanner	Deploy on Production	Pass
2	API Connection	The web application will connect to Virus Total	Pass
3	Test Data Base	Add URL, E-mails, and create schedules on the data base	Pass
4	Send e-mail notification	A report will be sent by email with the result of the scan	Pass
5	Server Performance	The server will not suffer any impact in its performance.	Pass

FUTURE WORK

Through the development of the project new ideas are arising of what more to improve or how to improve the system, and what kind of new features to add. Another possible step to be implemented is an additional module which could add a list of hash to scan them, obtaining results of malicious categories for each. The notification could be implemented via text message in case of finding any malicious result to obtain a faster action of the security analysis if necessary.

CONCLUSION

The aim of the project was to make a complete, fully working web-based project reputation scanner for the company. Requirements from the company have been gathered and considered. The final product has a GUI interfaces take away all the incidences of errors and labor and give something manageable. The final objective of this project is to improve the process. The right tools were used, and it was assured that the page would be in standard and acceptable to every visitor. Finally, the whole system has been tested to ensure that everything functions correctly before the system processes actual data and produces information that people will rely on. This will improve the workflow and other business processes including the productivity of the information security analyst employee. The company was satisfied with the functions implemented and their reliability and robustness. Appropriate experience of the overall structure and concept of the system was acquired through the project, and during the process of implementation new ideas have resulted. It is expected in a near future to have time allocated and the possibility to improve the system.

REFERENCES

[1] Virustotal. (2019). *How to use VirusTotal Community* [Online]. Available: https://www.virustotal.com/en/documentation/virustotal-community/.

[2] Wikipedia. (2016, Jul 16). *TrustedSource* [Online]. Available: https://en.wikipedia.org/wiki/TrustedSource.

[3] N. Schütte. (2018, April 12). *Automating Your Feature Testing with Selenium WebDriver* [Online]. Available: https://www.smashingmagazine.com/2018/04/feature-testing-selenium-webdriver/.

[4] Trajectory. (2019). *Software as a Service* [Online]. Available: http://trajectoryllc.com/id14.html.

[5] A. Kumar. (2014, Jul 27). *Cloud Computing* [Online]. Available: https://akashshivanand.com/cloud/cloud-computing/.