PROPOSAL OF WEBAPPS SCANNER ON CLOUD

HUDA KHAN, DEVEN SHAH

Abstract : Considering general security risks associated with Webapps and issues with Webapps scanner, we are going to provide a web service through the cloud which will scan the client's web Application and generate a report in standard format for the subscribed client.

Keywords : Web Application Security, Webapps scanners, Vulnerabilities, Private cloud security.

Introduction: A web application security scanner is a program which communicates with a web application through the web front-end in order to identify potential security vulnerabilities in the web application and architectural weaknesses [1]. Vulnerabilities in web applications may result in stealing of confidential data, breaking of data integrity. OWASP (open web application security project) is the most efficient way of finding security vulnerabilities in web applications. It has two methods of scanning, first is manual scanning, this technique is very time-consuming and requires expert skills also, and is prone to overlooked errors. Another method is automated approaches to finding security vulnerabilities.

Web application security vulnerabilities such as cross-site scripting, SQL injection and cross-site request forgeries are acknowledged problems with thousands of vulnerabilities reported each year [13]. These vulnerabilities allow hackers to perform unwanted actions that range from gaining unauthorized account access to obtaining sensitive data such as credit card numbers. In the extreme case, these vulnerabilities may reveal the identities of intelligence personnel [3]. According to research in 2011, the importance of data connected to web applications make them the target of frequent hacking, the average web site had serious vulnerabilities.

According to Gartner cloud computing is "a style of computing where scalable and elastic IT-enabled capabilities are delivered as a service to customers using Internet technologies. The value of cloud computing based on the fact that it means users don't have to be concerned with the amount of storage required and they have enough compute power available to search quickly through all their records. Basically cloud provide instance of virtual machine, and that will give log isolation to individuality client, that is the best advantage of cloud, providing isolation. Keeping these things in mind, we are going to provide web apps scanner as a service on cloud, which will have ability to scan all OWASP vulnerabilities, in just one click and provide report in standard format which helps developer to sort out vulnerabilities in easy manner.

Ease of use:

Web application vulnerabilities: Web applications contain a mix of traditional flaws (e.g., ineffective authentication and authorization mechanisms) and web-specific vulnerabilities (e.g., using user-provided inputs in SOL queries without proper sanitation). Here, we will briefly describe some of the most common vulnerabilities in web applications (for further details, the interested reader can refer to the OWASP Top 10 List, which tracks the most critical vulnerabilities in web applications [6]):

– Cross-Site Scripting (XSS): XSS vulnerabilities allow an attacker to execute malicious JavaScript code as if the application sent that code to the user. This is the first most serious vulnerability of the OWASP Top 10 List.

– SQL Injection: SQL injection vulnerabilities allow one to manipulate, create or execute arbitrary SQL queries. This is the second serious vulnerability of the OWASP Top 10 List.

– Code Injection: Code injection vulnerabilities allow an attacker to execute arbitrary commands or execute arbitrary code. This is the third most serious vulnerability on the OWASP Top 10 List.

– Broken Access Controls: A web application with broken access controls fails to properly define or enforce access to some of its resources. This is the tenth most serious vulnerability on the OWASP Top 10 List.

Web Application Scanners :

Web application scanners can be seen as consisting of three main modules: a crawler, attacker module, and an analysis module, The crawling component is seeded with a group of URLs, retrieves the corresponding pages, and follows links and redirects to identify all the reachable pages in the application. In addition, the crawler identifies all the input points to the application, for example parameters of GET requests, the input fields of any HTML forms, and the controls that allow one to upload files . The attacker module analyzes the URLs discovered by the crawler and the corresponding input points. Then, for each input and vulnerability type for which the web application vulnerability scanner tests, the attacker generates values that are likely to trigger vulnerability. For example the attacker module would

try to attempt to inject JavaScript code when testing for XSS vulnerability, or strings that have a special testing is for SQL injection vulnerability. Input values are usually generated using heuristics, such as those contained in one of the many available XSS and SQL injection cheat-sheets [7, 1]. The analysis module analyzes the pages returned by the web application in response to the attacks launched by the attacker module to detect possible vulnerabilities and to provide feedback to the other modules. For example, if the page returns in response to input testing for SQL injection contains a database error message, the analysis module may infer the existence of SQL injection vulnerability [1].

Issues with WebApps Scanners

Whenever we are going to install web application scanner, it want some plug-ins or have to link with some dependencies. For example if you are going with RIPS tool, it want local host and Mozilla browser, second if you want to work with mutillidae framework you need to configure local host and need to configure some files. After successful installation we need to set some settings manually then go for scanning part. So to avoid these things webapps scanner on cloud will work efficiently, provide report in just single click.

The Problem Definition:

Problem definition indicates that we have to scan web application vulnerabilities through webapps scanner providing client's URL as input which will locate on cloud and scanning report provided by a web service in a standard format (i.e. in PCI DSS etc). To achieving 'Webapps on Cloud', First thing will be that how mapping methodology will work, because after mapping we have parameter for webapps scanner. Mapping methodology may have following steps:

1] WHOIS queries -means WHOIS is server, in order to gather information about the domain name. Tool may then retrieve primary DNS servers which handle the domain.

2] DNS Zone Transfer – Webapps scanner through web service tries to extract a list of hosts for the domain by requesting zone transfers from the main DNS servers that handle the domain.

3] DNS Reverse lookup – web service also attempts reverse DNS lookup for each IP inside the domain. This may enable it to retrieve host names for related IP addresses.

4] Discovery methods for open services - In a standard scan, web apps scanner on cloud scans TCP ports 23 25, 21, 23, 80, 53, 111, 110, 139, 443, 445. UDP ports 53, 111, 135, 137, 161, and 500, run trace route, and sends ICMP packets in order to discover running services.

5] Router detection - Mapping reports up to two meaning in the SQL language, such as SQL operators and when routers directly in front of each mapped host. This test is performed by TTL field in the response packet from the target host.

6] Operating system detection - OS detection is done by TCP/IP stack fingerprinting, for which only one open TCP port is required.

Above information gathering (steps) are exactly same as attacker techniques. Second thing is how web apps scanner access client's URL parameters for scanning. For that we have to create or provide a web services which will take all parameter of client's URL. You will notice that the client applications can be running on different platforms, a major application of web service is to integrate such different applications which may be working on heterogeneous platforms. The web service resides on the web server. A client computer can request for URL scan and consume this web service and terminate the service when desired. And scan report of web application vulnerability remediation has been integrated into the compliance process of major commercial and governmental standards for example the Payment Card Industry Data Security Standard (PCI DSS), HIPAA and the Sarbanes-Oxley Act to the client.

The Proposed Mechanism **:** The proposed mechanism is to build the private cloud, and make a virtual machine and put Web apps Scanner tool in that, creating the Image of it. Upload that image on the cloud. Finally make an interface(web service) which allow user to use uploaded image of scanner to check his Web Application vulnerability, in last tool produces result/report in standard format (i.e. PCI DSS) and will send this report to client, all these process produces a report in just one click.

Methodology Used:

Before actual implementation of Web apps Scanner, following tasks needs to be successfully performed.

Setting up a private cloud. (with the help of Eucalyptus framework)Creating virtual machine image of Web apps Scanner.Make web interface which will provide web service to user directly.To use our Web apps scanner services clients need to subscribe for the web service.

This image can be used further to scan client web Application.

Performance Evaluation:

For scanning, Client has to subscribe for our web service, and then web service will ask to the authentication, after authentication clients are able to scan their URL. When client enter his URL and click on scan button, that will automatically initialize VM of web apps scanner , and then that will take the parameters from the respective server and scan the web apps and finally report generated into standard

format and that will directly send to the client browser.

Apart from the detected vulnerabilities, a useful scanning report should give clear and concise information about fixing the problems uncovered, When administrators need to perform subsequent scans after initial scanning or configuration changes, or make comparison between the results of previous scans, a scanner with a back-end database that can keep an archive scanning results for trend analysis is preferable.

The accuracy with which critical vulnerabilities are identified which are more important than the number of vulnerability checks, because the same vulnerability may be counted more than once by the scanner. The effective number of vulnerability in terms of Common Vulnerabilities and Exposures (CVE) 3 can be compared in a list of standardized names for vulnerabilities and other information security exposures [1]. The content of a CVE is a result of a collectively effort by the CVE Editorial Board.

Implementation Platform:

HARDWARE:

- Fast processor (p4/p5 2 GHz Intel chipset)
- Min. 1 GB RAM
- 100 GB HDD (as clients are limited)
- Bandwidth (full bandwidth is available i.e. 100 mbps)

Software:

- Linux OS (Centos 6.3, Ubuntu etc)
- Eucalyptus Framework for cloud deployment
- Webapps scanning tool (Acunetix, web scarab, wapiti, OWASP ZAP)
- Java Environmental
- VMware for Linux
- SOA and web service
- Data management Tools

Skills / Expertise:

- Understanding of Tool & its technologies
- Understanding of generated report analysis.
- Understanding of network traffic analysis.
- Experience with trouble ticketing and change management tools.
- Ability to learn new skills quickly.

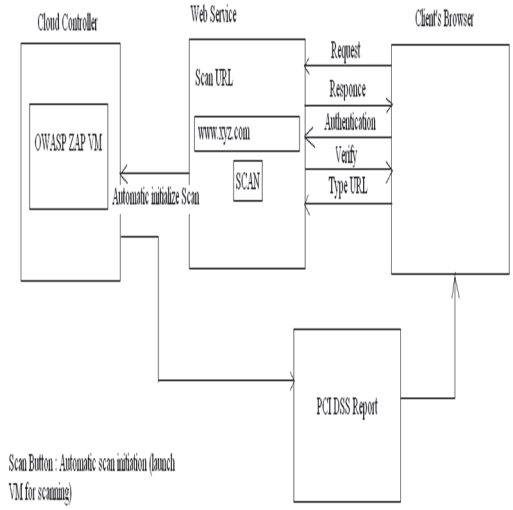

Fig. 1 Proposed mechanism

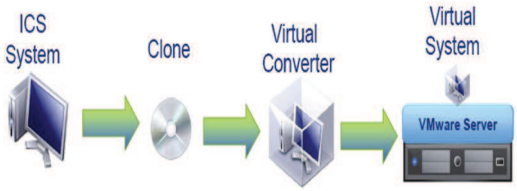

Fig. 2 VM image of Cloud

Conclusion: Web apps Scanner offers traditional scanning of entire web apps. In my paper scanner scan web apps through the one cohesive web service. Treating web applications as the business assets that we are, combined Web apps scanner with cloud approach to vulnerability life cycle management give you the most powerful and scalable vulnerability management with standard format.

References :

1. Acunetix_Analytics_Analysis_of_Blackbox_Web_Vulnerability_Scanners_rus.pdf
2. An overview of vulnerability scanners in February2008 http://www.infosec.gov.hk/english/technical/files/vulnerability.pdf
3. Arian J. Evans, "Software Security Quality: Testing Taxonomy and Testing ToolsClassificafi-cation" Presentation viewgraph for OWASP APPSec DC, October 2005.

4. D. Litchfield. SQL Injection and Data Security Breaches. [Online]. Available: http://www.davidlitch field.com /blog/archives/00000001.htm.
5. http://sectooladdict.blogspot.in/2011/08/commercial-web-application-scanner.html
6. Open Web Application Security Project (OWASP): OWASP Top Ten Project. http://www.owasp.org/ index.php/Top_10 (2010)
7. RSnake: Sql injection cheat sheet. http://ha.ckers.org/sqlinjection/ RSnake: XSS (Cross Site Scripting) Cheat Sheet. http://ha.ckers.org/xss.html
8. State of the Art: Automated Black-Box Web Application Vulnerability Testing by Jason Bau, Elie Bursztein, Divij Gupta, John Mitchell http://theory.stanford.edu/~jcm/papers/pci_0akland10.pdf
9. Web application vulnerability detection using Dynamic analysis With peneteration testing http://www.ijecbs.com/January2012/28.pdf
10. Web Application Scanners: Definitions and Functions by Elizabeth Fong and VadimOkunhttp://samate.nist.gov/docs/wapaper.pdf
11. Web Application Security Consortium Glossary, http://www.webappsec.org
12. Web Security Threat Classification. Web Application Security Consortium. [Online]Available:http:// www.webappsec.org/projects/threat/
13. Web Application Security FOR DUMmIES by Mike Shema
14. http://static.progressivemediagroup.com/Uploads/WhitePaper/912/96bb16b0-5a62-4345-aa92-0fd402e85953.pdf

* * *

401/Swagat Unique, Opp Laxmi Park, Naya nagar, Mira Road, Thane 401107/Lecturer/Information Technology/hudskhan@gmail.com
Mumbai, Principal, sir.deven@gmail.com.
1,2Information Technology Department, Terna Engineering College, Mumbai, India