WEB APPLICATIONS SCANNER ON CLOUD

HUDA KHAN

Abstract: Considering general security risks associated with Web applications and issues with Web applications scanner, we are providing a web service through the cloud which will scan the client's (developer) web Application, identify the potential vulnerability and generate a report in standard format for the subscribed client.

Keywords: Web Application Security; Web application scanners; vulnerabilities; private cloud security; Web services.

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, which is the best advantage of cloud, providing isolation. Keeping these things in mind, we are providing two type of cloud based service through our project i.e. IAAS and SAAS, WebApps scanner as a service on cloud, which will have ability to scan all web based vulnerabilities, in just few clicks 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 SQL 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 meaning in the SQL language, such as SQL operators and when 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 numbers of 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. For wapiti scanner you have to install java in your system, like wise some tool needed python etc, and after successful installation we need to set some settings manually then go for scanning part. Keeping all these terms in mind, WebApps scanner on cloud will work efficiently; provide report in just few clicks. No need to do all the above configurations.

The Problem Definition:

The main idea behind "WebApps Scanner on Cloud" is to detecting vulnerabilities in Web Applications, using vulnerability Scanner which is on Cloud. And provide the result in standard format, which is in readable mode developer, can easily understand and sort out vulnerabilities in easy manner.

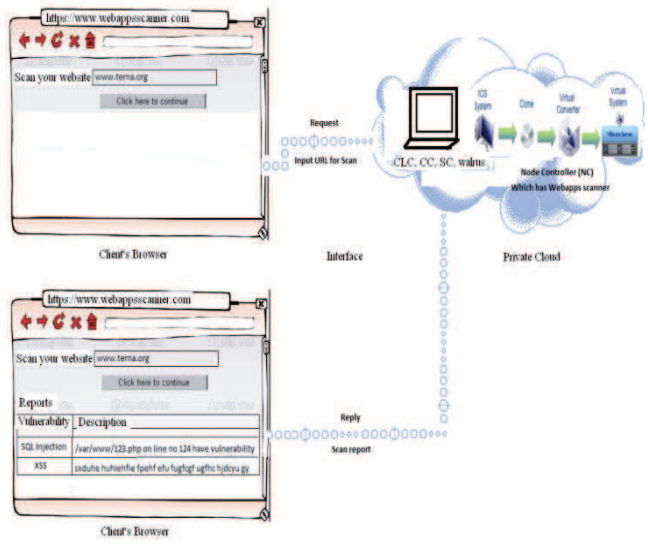
The Proposed Mechanism:

For proposed mechanism, I have made private cloud (put WebApps scanner in .img file and upload on cloud), and web service (with web interface script). I wrote a web service script in such a way that will accept the URL from client that will comes to cloud, cloud will boot the instance, run the configured tool, Generate the result in standard format that will help developer to sort out vulnerabilities in easy manner. With few clicks, our mechanism provides efficient report and save lots of installation and configuration time of client's (developer)

Methodology used:

For Web apps Scanner on Cloud, following tasks have done successfully.

- Set up a private cloud. (with the help of Eucalyptus 3.2.1 framework)
- Created virtual machine image configured with WebApps Scanner.
- Make web interface that is used to connect cloud and web service.
- For WebApps scanner service clients (developer) need to subscribe with our web service.
- Below image is the scenario of the '*WebApps Scanner on cloud*'.



Proposed mechanism

Implementation Platform:

COMPUTER REQUIEMENT:

- Fast processor Intel or AMD 2 GHz cores.
- Linux OS (Centos 6.3, Ubuntu etc), OS support 64bit architecture and using NTP
- KVM configured on NC machines.
- Verify that all machine in your network allow SSH

NETWORK REQUIREMENT:

- All NC must have access to minimum 1gb Ethernet network connectivity
- All components have at least one NIC card.
- Allow VLAN trucking.

STORAGE REQUIERMENT:

- 100GB for Walrus and Storage controller.
- 50 to 100 GB for Node controller.
- 500GB for Cloud or cluster controller.
- Each machine needs minimum 4 GB RAM, NC should be VT enabled.

PROJECT REQUIERMENT:

- Web Application scanner for e.g.: Ratproxy, wapiti, webscarb etc.
- Nusoap web based service.
- PHP and shell scripting.

Results and Discussion:

Results of "*WebApps scanner on cloud*" are shown in the following section. Its shows services of cloud, how to use WebApps scanner of images on cloud, how web service look like and in last report.

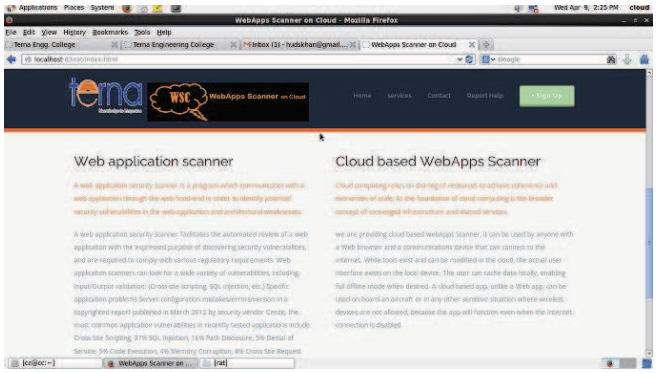
First, check whether cloud is working properly or not with the help of following commands and screen short:

```
[root@cc ~]# euca-describe-services -E
SERVICE cluster        cc             cc             ENABLED      16      http://192.168.200.1:8774/axis2/services/EucalyptusCC   arn:euca:eucalyptus:cc:cluster:c
c/
SERVICE storage        sc             sc             ENABLED      16      http://192.168.200.1:8773/services/Storage      arn:euca:eucalyptus:sc:storage:sc/
SERVICE bootstrap      bootstrap      192.168.200.1  ENABLED      16      http://192.168.200.1:8773/services/Empyrean     arn:euca:bootstrap:::192.168.200.1/
SERVICE walrus         walrus         walrus         ENABLED      16      http://192.168.200.1:8773/services/Walrus       arn:euca:bootstrap:walrus:walrus/
SERVICE dns            eucalyptus     192.168.200.1  ENABLED      16      http://192.168.200.1:8773/services/Dns          arn:euca:eucalyptus:::dns:192.168.200.1/
SERVICE eucalyptus     eucalyptus     192.168.200.1  ENABLED      16      http://192.168.200.1:8773/services/Eucalyptus   arn:euca:eucalyptus:::192.168.200.1/
[root@cc ~]#
[root@cc ~]#
[root@cc ~]# euca-describe-availability-zones verbose
AVAILABILITYZONE        cc      192.168.200.1 arn:euca:eucalyptus:cc:cluster:cc/
AVAILABILITYZONE        |- vm types     free / max   cpu   ram  disk
AVAILABILITYZONE        |- m1.small    0002 / 0002  1    512    5
AVAILABILITYZONE        |- c1.medium   0001 / 0001  2    512    10
AVAILABILITYZONE        |- m1.large    0000 / 0000  2    1024   15
AVAILABILITYZONE        |- m1.xlarge   0000 / 0000  2    2048   20
AVAILABILITYZONE        |- c1.xlarge   0000 / 0000  4    4096   20
[root@cc ~]#
```

Availability Zone output on CC (Cluster Controller)

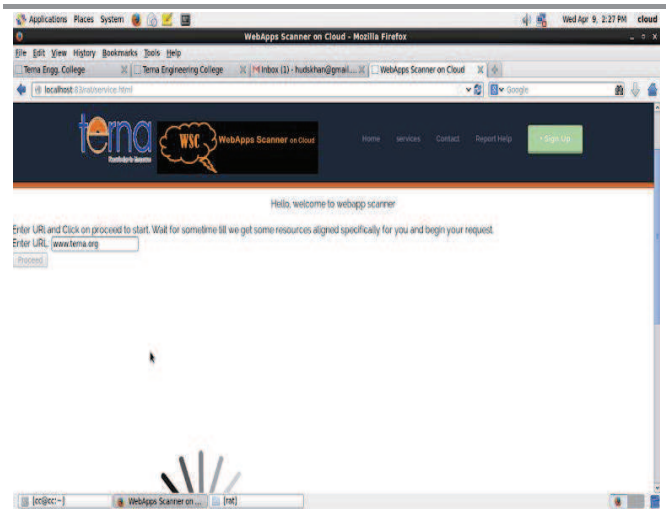
Second, for starting web service open the system browser and type your URL of web service i.e. http://localhost:83/index.html (web service hosted on local host). It shows the home, services and contact page, Home page introduced, what is cloud and Cloud based WebApps scanner.

Service page has our actual service which will start the WebApps scanner of our cloud and provide the service of WebApps Scan. It has one text box, in which user can enter URL (which is to be tested). Then click process button, it will connect with cloud and cloud will boot image which has ratproxy scanner (Ratproxy scanner in browser based proxy setting tool which is semi automated scanner mean it has active crawler and manual proxy setting).



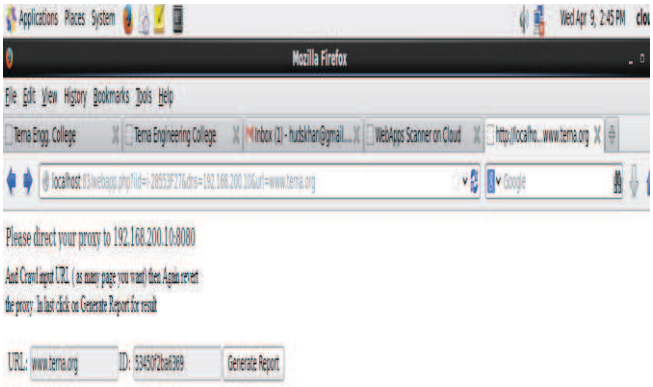
WebApps Scanner on Cloud service "Home Tab"

After that it will provide one page which has two text boxes which has provided URL and generates id and generate report button. Before generating the report user have to set the proxy then browse the URL then again revert the proxy then in last press generate report. It will provide standard report which will easily read by developers (clients)



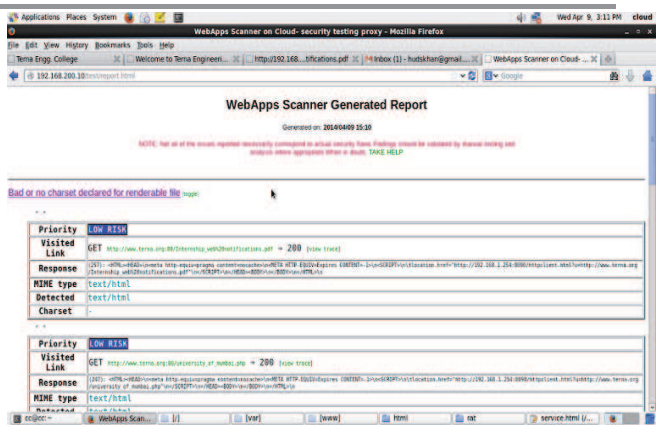
WebApps Scanner on Cloud service "Service Tab(scan)"

When we pass URL (which we want to be scanned) through cloud with web application scanner i.e. Ratproxy. It will take the URL as input to scanner and scanned the provided URL. Since we are using ratproxy tool so we have to set the proxy of system where our scanner is located, that is on cloud. After that crawl the URL then again revert the proxy then click on Generate Report tab.

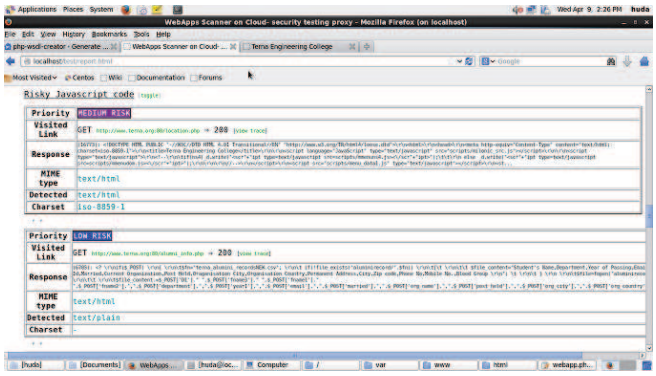


page displayed to user when nuSoap client-server connection is successful and web service is executed inside the running instance

In last report is generated which describe priority of vulnerability, name of the vulnerability, show response number of server i.e. 200, 304, 404 etc, also show visited links etc. Report has one link "TAKE HELP" which help to understand the vulnerability in detail.



Final WebApps Scanner Generated OUTPUT screen



Final WebApps Scanner Generated OUTPUT screen

Conclusion:

Web apps Scanner offers traditional scanning of entire web apps. In my report scanner scanned WebApps 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. Another advantage of "*WebApps Scanner on the cloud*" is that it lends itself to subscription-based software, which doesn't require complex licensing or distribution points, which not only cuts cost, but also ensures no piracy. Clients will have an access to service on pay as peruse basis. Clients don't have to invest in any local hardware and can access their information and services from any Internet access. This type of application moves away from the requirements of having big applications on client's systems to processing everything on the servers, which means clients need less money to get into application.

Future scope:

"*WebApps scanner on cloud* "has number of future scope out of which few are: First, we can add different web scanner and provide the service of that scanner through cloud. Second, we are providing report in standard format which can be more elaborate to understand easily(Multiple reporting formats can be supported).Third, in future scope we can automate web crawling because of that we can achieve report in single click. Fourth, the scan reports can be saved in the volume by creating a volume for each incoming user request and the volume can then be attached before booting up the instance in the cloud. Therefore all the reports will be saved in the created volume and every time the user requests for new scan can also have access to previous reports. Thus helping to collect wide information which further helps network administrators in preparing better mitigation plans. Fifth, the scope of this project was for pc hence the web application is developed for PC use only in future scope can be widened on commercial basis which can be extended to other communication devices as well for e.g. Android cell phones, iphone.

References

1. Acunetix_Analytics_Analysis_of_Blackbox_Web_Vulnerability_Scanners_rus.pdf
2. *Adenegan, Kehinde Emmanuel, Balogun, Folorunso Ojo,* Discussion on the Convergence/Divergence of Series and Sequences; Mathematical Sciences International Research Journal ISSN 2278 – 8697 Vol 2 Issue 2 (2013), Pg 127-130
3. An overview of vulnerability scanners in February 2008http://www.infosec.gov.hk/english/technical/files/vulnerability.pdf
4. Arian J. Evans, "Software Security Quality: Testing Taxonomy and Testing ToolsClassificafi-cation" Presentation viewgraph for OWASP APPSec DC, October 2005.
5. D. Litchfield. SQL Injection and Data Security Breaches. [Online]. Available:http://www.davidlitchfield.com/blog/ archives/00000001.htm.
6. http://sectooladdict.blogspot.in/2011/08/commercial-web-application-scanner.html
7. Open Web Application Security Project (OWASP): OWASP Top Ten Project. http://www.owasp.org/index.php/Top_10 (2010)
8. *V.T.Padmasani,* A Matheamtical Model for Bioconvection in Suspensions of; Mathematical Sciences International Research Journal ISSN 2278 – 8697 Vol 2 Issue 2 (2013), Pg 420-427
9. RSnake: Sql injection cheat sheet. http://ha.ckers.org/sqlinjection/ RSnake: XSS (Cross Site Scripting) Cheat Sheet. http://ha.ckers.org/xss.html
10. 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_oakland10.pdf
11. Web application vulnerability detection using Dynamic analysis With peneteration testing http://www.ijecbs.com/January2012/28.pdf
12. *A.Jayalakshmi, Ananth K. Atre,* on A Semigroup Whose Factorisable Elements Form A Band; Mathematical Sciences international Research Journal ISSN 2278 – 8697 Vol 3 Spl Issue (2014), Pg 951-955
13. Web Application Scanners: Definitions and Functions by Elizabeth Fong and Vadim Okunhttp://samate.nist.gov/docs/wa_paper.pdf
14. Web Application Security Consortium Glossary, http://www.webappsec.org
15. Web Security Threat Classification. Web Application Security
16. *S.Sudha, G. Anbarasi,* A Study on the Domination Parameters of P-Regular ...; Mathematical Sciences International Research Journal ISSN 2278 – 8697 Vol 2 Issue 1 (2013), Pg 16-18
17. Consortium. [Online]. Available: http://www.webappsec.org/
18. projects/threat/

* * *

Huda Khan/401, Swagat Unique, Naya Nagar, Mira Road, Thane 401107/H.O.D/KVMIT Polytechnic/hudskha@gmail.com