Performance Analysis of Cloud Based Penetration Testing Tools

Aruna Pavate
Dept. of Computer Engineering
St. Francis Institute of Technology
Mumbai University

Pranav Nerurkar
Dept. of Computer Engineering
Thadomal Shahani Engineering College
Mumbai University.

***Abstract*— Penetration testing is used for testing the systems for vulnerabilities and plugging the loopholes if any. All major enterprises have web applications or provide services online, so it is important to secure these systems. Most organizations have dedicated security analysts; some even hire third party consultants for the job. However due to improvements in technology it is no longer necessary to have trained personnel for testing. Using online (cloud) testing tools any ordinary person can test the system for vulnerabilities. So it's important to know which tools are available in the market and their advantages. This paper provides detailed analysis of tools and their performance and accuracy, the result can be used to understand which tool is best for the job.**

***Keywords*— penetration testing, black box, cloud, ethical hacking,network security.**

I. Introduction

The infrastructure of an IT company consists of thousands of components connected together as a high level architecture. These components are critical for the working of the company and even if there is an attack on any subsystem of this architecture it can result in decrease in performance of the entire enterprise. Modern industries recognize this problem and so have a dedicated team of security analysts to handle the security issues that are faced by the enterprise. Sometimes even third party analysts are used for monitoring the network and resolving issues related to it.

The field of testing related to identifying of security issues in a network and resolving them before external hackers can exploit it, is called penetration testing. The main objective of penetration testing is to correctly assess the real security risks associated with a seemingly endless stream of vulnerability and patch reports. However IT professionals understand that despite their best efforts, vulnerabilities may still present significant security risks for their companies. The need for penetration testing is to intelligently manage vulnerabilities.

Penetration testing provides detailed information on actual, exploitable security threats. By performing a penetration test, an organization can identify which vulnerabilities are critical, which are insignificant, and which are false positives. Avoid the cost of network downtime.

Recovering from a security breach can cost millions due to IT remediation efforts, lost employee productivity, and lost revenue. Penetration testing allows an organization to prevent this financial drain by identifying and addressing risks before security breaches occur.

Preserve corporate image and customer loyalty. Even a single incident of compromised customer data can be costly. Penetration testing helps an organization avoid data incidents that put its goodwill and reputation at risk.

Justify security investments. Penetration testing can both evaluate the effectiveness of existing security products and build the case for proposed investments.

The experts make use of sophisticated techniques to identify the problems and resolve them. The entire process consists of phases like requirement gathering, analysis of information and classifying it according to the type of vulnerability, then attempt is made to exploit every flaw and then the response of the system is recorded.There have been various attempts to simplify penetration tests by automating various steps of the penetration test. The simplest attempt is Autopwn in Metasploit framework[7]. The security expert gathers information about target systems using Nmap or Nessus. This information is imported to a database using database module in Metasploit. Autopwn query the database for open ports and services. Then it loads the exploits in Metasploit that matches these services and launch them against the target systems.

The people involved in penetration testing activity are called pen testers or ethical hackers. Their role is to attack the system and try to crash it. However as this is done manually there is a chance of overlooking vulnerabilities. To counter this flaw software tools were designed to assist the programmers in their works. These tools required manual intervention in limited manner but they did the task in more precise manner. However the main flaw with these tools was that they were not freely available and required human help in some steps of operation. The results provided by these tools were at times vague, susceptible to misinterpretation [10]. These tools also gave false positives and ignored critical errors.

Online testing tools are a category of automated tools that don't need manual intervention. These tools are platform independent and are not installed on the client systems. They work on the principle of "Software as a service" and hence the use of these tools for testing is called "Security testing from the cloud". These tools allow the tester the advantage of not downloading every tool, all he has to do is go the website that provides this facility and he can use it. Most cloud testing sites provide the user to select tools of his needs viz. specific to operating system, service pack, language, and version numbers [2].

Testing the security of web applications with automated penetration testing tools produces relatively quick and easy results. However there are a lot of such tools, both commercial and free. Unlike traditional black box testing, in which an ethical hacker tries to attack the web application, penetration testing tools can be used by a person with little or no knowledge about security. Only the analysis of the result has to be done by a person with knowledge about security. However even online testing tools had problems like non- detection of critical bugs which could be abused by hackers. Also, some had false positives and false negatives which need manual verification.

The contribution of this paper is:

- To list various online tools given in the market for penetration testing.
- Analysis of their performance.
- Limitations of each tool.

II. Related Research

A. *Inner Working of Cloud-Testing Tools.*

Most of the penetration testing tools use a technique that is called fuzz testing, fuzzing or fault injection. Fuzzing has been defined by as: A highly automated testing technique that covers numerous boundary cases using invalid data (from network protocols, API calls and other targets) as application input to better ensure the absence of exploitable vulnerabilities[8] . From modem applications' tendency to fail due to random input caused by line noise on "fuzzy" telephone lines. The part of the program that does this is called a fuzzier or a fault injector[1].

The usual steps that are performed by penetration testing tools to discover vulnerabilities are explained in section IV [A].

1. Identify the target
2. Identify inputs
3. Generate fuzzed data
4. Execute fuzzed data
5. Monitor for exceptions
6. Determine exploitability

An easier way to divide the steps a penetration testing tool performs is:

- Crawling: The phase that crawls the web application to find the pages the web application consists of and vulnerable inputs.
- Fuzzing: This phase sends the data to test the web application to the application.
- Analyzing: In this phase the result of the fuzzing phase is analyzed to check if the web application is vulnerable.

B. *Comman terms.*

1. Fuzzy logic: Fuzzy logic is a form of many-valued logic; it deals with reasoning that is approximate rather than fixed and exact. Compared to traditional binary sets (where variables may take on true or false values) fuzzy logic variables may have a truth value that ranges in degree between 0 and 1.

2. Cloud: A new branch of Distributed computing, where applications are stored on a remote location and accessed from a browser.

3. Black box: A testing technique where hacker doesn't know anything about internal configuration of target system.

Bugs: Fault in the system that can be exploited.

III. Online Testing Tools

A. *Online Penetration testing tools*

In this section we concentrate on different penetration testing tools for cloud platforms. Basically these tools are useful a) when you need the availability/ functionality of a service from a different IP address. b) company firewall does not allow you to access restricted port c) The target service blocked your IP[5].

Though there are many sites that offer cloud testing services. The prominent 3 are:

1. https://pentest-tools.com/
2. http://www.livehacking.com/
3. http://www.penetration-testing.com/

These sites offer the following facilities for penetrative testing [3][4].

1. Email checker: Verifies if an email address is valid or not by interrogating the mail server of the target domain.
2. XSS server:This tool collects data from target users when exploiting XSS vulnerabilities in web applications.
3. Google Hacking: used to conduct a passive reconnaissance phase.
4. Find sub domains: Used to find sub domains of your target domain.
5. DNS Lookup: Allows you to perform any type of DNS lookups online
6. DNS Zone Transfer: Tries to perform DNS zone transfers automatically by querying all domain name servers of the requested domain
7. Who is Lookup: Find information about the owners of your target domain / IP address by querying the internet registrars.
8. ICMP ping: Discover which hosts are up within a range of IP addresses.
9. TCP port scan: Scan a host using nmap to see if a service is listening on the specified TCP ports
10. UDP port scan: Scan a host using nmap to see if a service is listening on the specified UDP ports

11. Xchg Rate Improver: This tool computes the most advantageous rate that you can have when making currency exchange transactions in your internet banking application. .

B. Test Setup

Testing the tools consists of several tests. A small test created was created on PHP BB, web goat these are WebPages specially built for testing for security flaws. All tests were performed with the tools set to scan for all vulnerabilities and otherwise the tools' default settings.

1. Web Goat: is a very extensive web application consisting of approximately69 vulnerable web pages, divided into 19 categories. This test case was chosen because of its extensiveness in number and type of vulnerabilities[5].

2. PHP BB: It was chosen because of the fact that it is a "real" web application instead of one with vulnerabilities implemented on purpose. Advantage of this was that it's difficult to identify vulnerabilities and disadvantage is that not all vulnerabilities are known.

The online testing tools were used to detect vulnerabilities in each of the above two sites, their reports were then recorded and the summary of the findings is given section IV. It is to be noted that PHP BB needs a web server and different servers might yield different results, in this Apache Tomcat was used.

IV. Test and Evaluation

The tools given in section II are open source and freely available. However each of them has its own set of flaws. A penetration test can only identify those problems that it is designed to look for. If a service is not tested then there will be no information about its security or insecurity[6]. A penetration test is unlikely to provide information about new vulnerabilities, especially those discovered after the test is carried out. Hence a common set of metrics have to be designed to evaluate the performance of each tool.

A. Comman Vulnerabilities

1. SQL Injection: this is the oldest attack possible on Servers. This attack is used to recover protected records from the database which are not available to users. Hacker might also gai unauthorized entry to a server.

2. XPath injection: this is similar to sql injection but works on XML files only.

3. XSS: attacker uses this to steal cookies and impersonate a user on a website.

4. Cross site tracing: Attacker can use this to get the headers of a web server.

5. CSRF: an attacker tricks a user's browser into loading a request that performs an action on a web application that user is currently authenticated to.

6. Local file inclusion: the attacker might be able to load a file that he should not be able to see.

7. Remote file inclusion: Remote file inclusion is equal to local file inclusion, except for that the file that is included is a file from a different server than the one the web application is running on.

8. HTTP response splitting: the attacker can control the data that is used in an HTTP response header and enters a new line in this data.

9. Command injection: the attacker can execute a command on the server.

10. SSI injection: attacker can enter SSI directives (e.g. <!#include file="file.txt" > or <!#exec cmd="ls -l">) that are then executed by the web server.

11. LDAP injection: the attacker inputs LDAP statements that are executed by the server.

12. Buffer overflow: buffer overflow occurs when an application tries to store more data in a buffer than the buffer can hold.

13. Session management: Session management vulnerabilities can mean several things: session prediction, session fixation or session hijacking.

B. Performance metrics

Performance of the tools was divided into two categories:

1. Non-Functional
   a. Reliability.
   b. Speed
   c. Number of false positives.
   d. Ease of use.
2. Functional
   a. User-friendliness
   b. Report-generation.
   c. Customization.
   d. Community/Discussion forums.

C. Parameters for evaluation.

The sites were evaluated on how they were able to detect the following vulnerabilities.

1. SQL injection: to find these vulnerabilities the sites must have modules that must be able to give input values and analyze the response.
2. XST flaw: this flaw is present if the server provides support for TRACE method.
3. XSS flaw: this is present if it's possible to enter HTML code into vulnerable spots.

   Apart from these the sites were tested against their ability to detect all other vulnerabilities mentioned in section IV-A.

D. Result

1. www.pentest-tools.com

Threat detected	11
Threat not detected	Remote file inclusion, session management
Time taken	58min 28 sec
accuracy	78%
False positives	17

2. www.livehacking.com

Threat detected	8
Threat not detected	Remote file inclusion,LDAP,XPATH, session management, XSS
Time taken	40min 28 sec
accuracy	74%
False positives	10

3. www.penetration-testing.com

Threat detected	9
Threat not detected	XSS, XPATH, session management, HTTP response splitting
Time taken	70min 28 sec
accuracy	62%
False positives	13

V. CONCLUSION

The important part of any pen-testing tool is crawling. A crawler should find all webpage's but not include doubles. None of the above tools could crawl Web Goat on default settings successfully. Fuzzing of all sites generated many false positives which had to be manually verified. Analyzing the response was above average for all sites.

VI. ALTERNATIVE TO PENETRATION TESTING

Continuous Penetration Monitoring

This approach can be useful for small scale enterprises which have high probability high impact risk profile. This strategy proves to be in someways more cost effective and efficient than penetration testing. The main idea behind CPM is: It is a security strategy that utilizes two firewalls: a hardware firewall at the Internet's point of entry, and, a software firewall on 6-8 PCs[12].

The logic behind CPM is that two firewalls from different vendors, utilizing different detection strategies, are unlikely to be penetrated by a single Hacking procedure. Accordingly, if the first firewall is penetrated, the second one will capture the attack and immediately issue a warning message.

The reason for installing the software firewall on 6-8 PCs is that it enables multiple people to monitor for a "break-in message". However it must be noted that this strategy may not work if the hacker knows the system installation.

REFERENCES

[1] "Penetration Testing with Improved Input Vector Identification"
W. Halfond, S. Roy Choudhary, and A. Orso
International Conference on Software Testing (ICST 2009).

[2] "Comparison of penetration testing tools for web applications"
Frank van der LooI.S. Jacobs and C.P. Bean

[3] "Windows Tools For Penetration Testing   Penetration Testing Lab"
K. Elissa R. Nicole,.

[4] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Penetration Testing Tools"

[5] "Online Penetration Testing Tools   Ethical Hacking Tools.htm"-http://www.pentest.com/.

[6] Finding bugs in web applications using dynamic test generation and explicit state model checking, Shay Artzi, Adam Kieżun, Julian Dolby, Frank Tip, Danny Dig, Amit Paradkar, and Michael D. Ernst. IEEE Transactions on Software Engineering, vol. 36, no. 4, July/August 2010, pp. 474-494..

[7] State of the Art: Automated Black-Box Web Application Vulnerability Testing, J. Bau, E. Bursztein, D. Gupta, J.C. Mitchell. IEEE Security & Privacy 2010.

[8] Fuzzing with Code Fragments, Christian Holler, Kim Herzig, and Andreas Zeller. Usenix Security 2012.

[9] Enemy of the State: A State-Aware Black-Box Web Vulnerability Scanner, Adam Doupé, Ludovico Cavedon, Christopher Kruegel, and Giovanni Vigna. Usenix Security 2012.

[10] Toward Automated Detection of Logic Vulnerabilities in Web Applications, V. Felmetsger, L. Cavedon, C. Kruegel, G. Vigna, Usenix Security 2010.

[11] Why Johnny Can't Pentest: An Analysis of Black-box Web Vulnerability Scanners, A. Doupe, M. Cova, G. Vigna, DIMVA 2010.

[12] "An Alternative to Penetration Testing"-http://www.bankersonline.com