VULNERABILITY SCANNERS: A PROACTIVE APPROACH TO ASSESS WEB APPLICATION SECURITY

Sheetal Bairwa[1], Bhawna Mewara[2]and Jyoti Gajrani[3]

[1,2,3]Department of Information Technology, Government Engineering College, Ajmer

ABSTRACT

With the increasing concern for security in the network, many approaches are laid out that try to protect the network from unauthorised access. New methods have been adopted in order to find the potential discrepancies that may damage the network. Most commonly used approach is the vulnerability assessment. By vulnerability, we mean, the potential flaws in the system that make it prone to the attack. Assessment of these system vulnerabilities provide a means to identify and develop new strategies so as to protect the system from the risk of being damaged. This paper focuses on the usage of various vulnerability scanners and their related methodology to detect the various vulnerabilities available in the web applications or the remote host across the network and tries to identify new mechanisms that can be deployed to secure the network.

KEYWORDS

Vulnerability, Static analysis, Attack graph, Scanners, Test–Bed

1. INTRODUCTION

With the emergence of information technology, the security aspect of the users has become a more concerned factor. Since most of the software developers are not aware of various security measures to be introduced into the system as their motive is just to make the software application run in a desired state without taking into consideration the flaws that the programming language might have introduced into the system; to protect the users from the risk of being attacked by any unauthorised access, it becomes significantly more important to devise new strategies and methodologies that will consider the security breaches to which the user is prone to. Not only the software developed with flaws makes the user vulnerable to attacks, most often network also becomes a key factor by compromising the security aspect of the users.

Assessing and eliminating the vulnerabilities requires the knowledge and deep understanding of these vulnerabilities. It becomes necessary enough to know the basic idea that works behind these vulnerabilities such as what makes them to appear in the system, what flaws need to be corrected to make the system free from these vulnerabilities, what alternatives can be further devised for these vulnerabilities so that in future, their risk can be reduced and many more.
Various methods have been deployed to identify these vulnerabilities and appropriate steps are taken. Strategies such as static analysis, attack graph generation and its analysis, usage of vulnerability scanners are some of them. However, the use of vulnerability scanners to detect the vulnerabilities is quite prominent today. They play a significant role in the generation of attack graphs.

DOI:10.5121/ijcsa.2014.4111

Our work involves study of various port scanners and vulnerability scanners, scanning of various online web applications and remote host using these scanners. We analysed various vulnerabilities and make a comparison of various scanners based on their capability to identify these vulnerabilities.

Section 2 explains various techniques developed before the usage of vulnerability scanners. Section 3 describes various vulnerability scanners in detail with the results, when applied on various websites. Comparative study of various scanners is given in Section 4.

2. TECHNIQUES FOR VULNERABILITY SCANNING

2.1 STATIC ANALYSIS

Static analysis is a fast and reliable technique. It has been considered as an efficient method in detecting the vulnerabilities [3].This technique focuses on the analysis of program structure using various means. It emphasizes on the analysis of the code of the program in order to detect the flaws present in it.

Some of the techniques included in static analysis are lexical analysis, type inference, constraint analysis and many more. Lexical analysis focuses on the semantics of the program structure; the program structure is divided into modules and then each module is compared with the loophole library in order to detect any flaws present in the system. Type inference is related to the data type rules for the variable. It determines whether the variables used in the program are in sync with the type to which they relate. Constraint analysis is a two-step process. It involves- constraint generation and constraint solution [1].

Many tools based on the techniques mentioned above are developed. The first tool developed was FlexeLint. It uses pattern matching algorithm to detect flaws. Other tools developed are ITS4, SPLINT, UNO, FindBugs, Checkstyle, ESC/Java, and PMD. ITS4, Checkstyle and PMD are based on lexical analysis; SPLINT is based on rule checking; UNO is based on model checking; ESC/Java is based on theorem proving and FindBugs is based on both lexical and dataflow analysis [1].

These tools have been evaluated by analysing their performance in terms of false positives and false negatives. Many of them have low false positives, some produce accurate results and many witnessed high false negatives. Hence, static analysis techniques have many demerits associated with them. For instance, a loophole library or database is maintained which is used to validate the vulnerabilities found in the program; however if an unknown vulnerability is detected, then it is not possible to compare it with the predefined loophole library for its validation [1].

Thus, to resolve the deficiencies associated with the static analysis, an approach was suggested that involved combining the dynamic detection strategy with static analysis.

2.2 ATTACKGRAPH ANALYSIS

Attack graph is defined as the succinct representation of all the paths followed by an attacker in a network to achieve its desired state. The desired state may involve damaging the network, stealing the network packets or gaining a complete access over it to determine what is going in the network.

Network security is a key aspect of security concern and many ways have been identified to protect it. The recent approach that has been included is the use of attack graphs. Attack graph has become the most widely used approach with reference to network security.

Attack graphs help to determine the security weaknesses that lie in the network. System administrators use it to analyze the network for its weaknesses that may allow an attacker to exploit it and gain control over the network [2]. Attack graphs are usually large enough as they represent the complete network with its underlying weaknesses, hence they are quite complex to understand and analyse. Both the generation and analysis of attack graph are significant for protecting the network from security breaches.

The most common approach to generate an attack graph requires the analysis of vulnerabilities that lie in the network and then using an attack graph generator, attack graphs can be generated [4]. The vulnerabilities could be identified with the help of various vulnerability scanners that are designed for this purpose only. Specifically, Nessus is extensively used for the identification of the underlying vulnerabilities.

Various other techniques have already been proposed for generating an attack graph as well as for their analysis. For instance, adjacency matrix clustering algorithm makes the complex attack graph simpler enough. It combines the blocks having similar attack graph pattern. The matrix represents the attack reachability within one step. For multiple steps, matrix is raised to a higher power level [13].

Ranking algorithm is another approach, based on the rank of the attack graphs. The rank decides the priority of an attack graph that is more applicable to attacker [14]. Another approach is a game theoretic approach where the attacker and network administrator are considered as two players and a Nash equilibrium is applied that gives the administrator an idea of attacker's strategy and helps him to plan to do something in order to protect the network [12].

Table 1 above compares the various attack graph generation and analysis techniques and illustrates the advantages and disadvantages of each [2].

Technique	Author	Merits	Demerits
Clustered adjacency matrix	Steven Noel Sushi Jajodia	Automatic, parameter-free, and scales linearly with problem size	Need to calculate highest level of adjacency matrix for multistep reachability
Hierarchical aggregation	Steven Noel Sushi Jajodia	Framework useful for both computational and cognitive scalability	The process of interactive de-aggregation is potentially tedious to determine low level details
Minimization analysis	S. Jha O. Sheyner J. Wing	Identifies the smallest set of countermeasures required to prevent all possible attack paths	Approach is limited to Directed Acyclic Graph
Ranking graph	Vaibhav Mehta C. Bartzis Haifeng Zhu Edmund Clarke J. Wing	Ease and flexibility of modelling	Difficult for security manager to make decision on actions to protect network
Game theoretic	K.W. Lye Jeannette Wing	Allows to know more about attacker's attack strategies	Full state space is extremely large.

Table 1: Comparison of the attack graph techniques

3. VULNERABILITY SCANNERS

A large number of applications are becoming online, but how secure are these products is a matter of concern as it is related to the user's security who will be ultimately using the application. Thus, it becomes necessary to find out vulnerabilities present in the software application that may cause a severe risk to the user's security [5].

Vulnerability assessment means identifying the vulnerabilities in the system before they could be used by anyone else with bad intentions of harming the network. This is a proactive approach where the vulnerability is found and is dealt with accordingly before anyone comes to know about it. More emphasis has always been laid on the firewall protection but the internal functionality does matter. Vulnerability assessment is not only performed on a particular application but it even correlates the platform on which the application is being run, middleware, operating system being used etc. It takes into consideration all the factors that can provide the correct answer for the assessment of the vulnerability and security of the system. Therefore, vulnerability scanners are used to scan the network system and/or the software applications.

Scanning can be of two types:

a) **Passive Scanning**: In passive scanning, it is determined whether a tool can enlist the vulnerabilities by considering the existing network.
b) **Active Scanning**: In active scanning, it is determined whether the queries can be made to the network for the vulnerability.

Different categories of scanner are:

a) **Port Scanners**: Port scanners are used to scan the ports for determining the open and closed ports, operating system, services offered.
b) **Application Scanners**: Application scanners are used to assess a specific application on the network in order to track its weaknesses that can be further used to cause the risk to the system.
c) **Vulnerability Scanners**: Vulnerability scanners are the ones that find out the vulnerabilities in the system which if accessed by a malicious user or hacker can put the whole network system at risk.

Penetration testing is the other concept that follows the vulnerability assessment. With penetration testing, it is possible to make use of the loopholes or vulnerabilities to gain an unauthorised access. It validates how effectively the system can respond to the real life attacks.

OWASP (Open Web Application security Project) focuses on providing the better security of the software. It has enlisted commonly critical vulnerabilities that the application may be prone to. These vulnerabilities when exploited provide the risk of losing security and confidentiality. For instance, Injection vulnerability occurs due to the execution of a command or query for an untrusted data; Broken Authentication and Session Management, due to improper implementation of an application risks the user's confidentiality. Cross Site Scripting, commonly referred as XSS is another flaw in which attacker injects malicious script into web pages viewed by users and also to bypass access controls. Insecure Direct Object References, in which developers unknowingly leave some holes which give a chance to attackers to access and manipulate directory, database key. Cross Site Request Forgery or CSRF, is an attack where user is forged to click on a link that is intuitively designed to steal the cookies and other private details of the user. Sensitive data exposure is another area of vulnerability where the sensitive data such as credit card details, authentication credentials etc. are not secured which helps an attacker to conduct the fraud [15].

Next subsections discuss various scanners and the results obtained by scanning various web applications using these scanners.

3.1 NMAP

Nmap is a port scanner that is used to scan the ports. It takes an IP address or the host name and then finds the basic information related to it. If an IP address is provided, it then finds the host to which it belongs to. It also finds the number of ports that are running on that particular host, number of ports that are opened, number of closed ports, services provided by those ports, for instance, whether services are TCP-oriented or FTP-oriented [10]. It even predicts the type of operating system being used on that particular host. The topology of the scanned host is recorded in the graphical format which shows the various gateways through which the local machine accesses that particular remote host.

Considering the ports that are opened, an attack can be designed in order to have an unauthorised and a legitimate access to the host with a goal set in mind. Moreover, if the opened ports are providing the services which are TCP-oriented or FTP-oriented, it becomes easy to gain access to the host.

A number of various sites have been scanned using NMAP. The figure below depicts the results obtained after scanning RTU website.

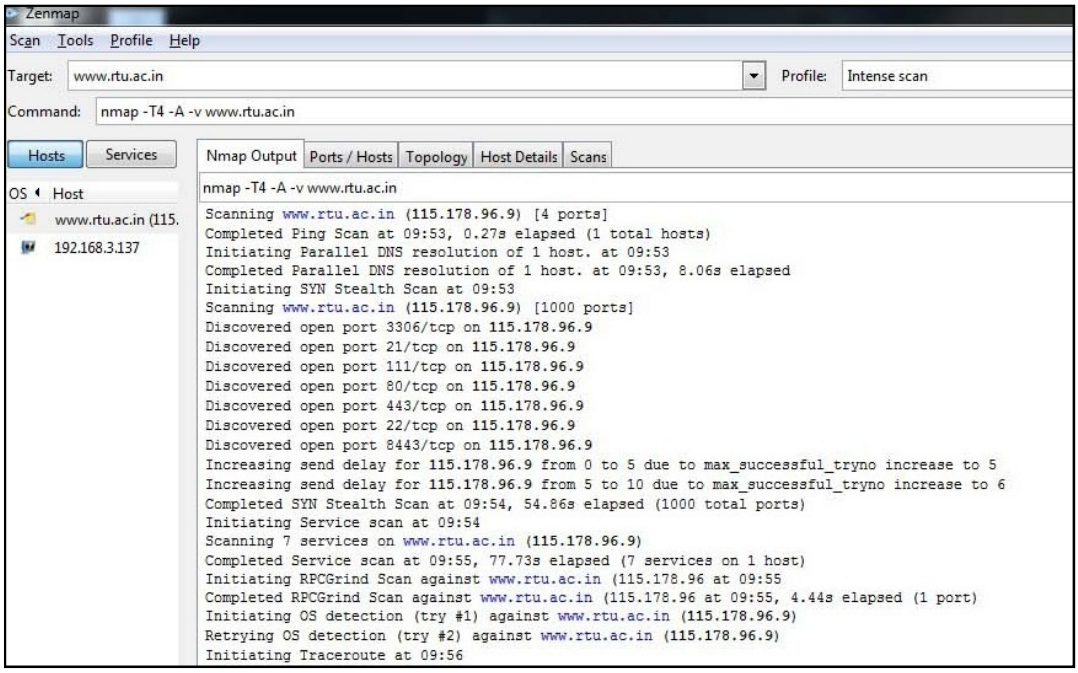

Figure1. Nmap basic output for RTU website

Figure 1 shows the basic details of RTU website including the IP address, number of total ports available, number of open ports discovered, performing RPCGrind scan and much more other relevant details.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.