Analysis of Penetration Testing Tools

Palak Aar, Aman Kumar Sharma
Computer Science Department, Himachal Pradesh University, Shimla, Himachal Pradesh, India

Abstract - Penetration testing is defined as the procedure of imposing as an attacker to find out the vulnerabilities in a system that can be used to gain access to system for malicious use. This paper provides an overview of penetration testing and list out the criteria used to select the best tools for the given purpose. It also provides a brief description of the selected tools and furthermore we compare those tools. The results of the comparison are shown in terms of graphs and tables.

Keywords - Dmitry, hping3, nmap, penetration testing, unicornscan.

I. INTRODUCTION

Information Technology (IT) systems have become an indispensable part of many modern organizations. An efficiently implemented system may not only enable smooth operations, but also improve management processes significantly. Unfortunately, despite the remarkable benefits brought by these systems, companies might suffer catastrophic consequences and losses if the systems were hacked by cyber criminals. As a result, a large variety of defensive mechanisms are needed to prevent intruders.

A penetration test is a security-oriented systematic probing of system from "inside" or "outside" to seek out vulnerabilities that an attacker could exploit. A system could be any combination of application, host or networks [1].

Penetration test provides a bird-eye perspective on current security posture of an organization’s IT infrastructure. The intent of a penetration test is to determine the feasibility of an attack and its impact of a successful exploit if discovered. The process involves an active analysis of the system for any potential vulnerabilities that may result from poor or improper system configuration, known and/or unknown hardware or software flaws, or operational weaknesses in process or technical countermeasures. It helps to narrow down security risk and confirm whether the current security measures implemented are effective, or not [2]. It is not merely the serial execution of automated tools and generation of technical reports as it is frequently viewed. It should provide a clear and concise direction on how to secure an organization’s information and information systems from real world attacks.

One critical factor in the success of penetration testing is its underlying methodology. A systematic and scientific approach should be used to successfully document a test and create reports that are aimed at different levels of management within an organization. It should not be restrictive to enable the tester to fully explore his intuitions [9].

A vulnerability assessment normally forms the first part of a penetration test. The additional step in a penetration test is the exploitation of any detected vulnerabilities, to confirm their existence, and to determine the damage that might result due to the vulnerability being exploited and the resulting impact on the organization [10].

Before a penetration test can be conducted there are certain requirements that need to be met. Stephen Fried in his paper “Introduction To Penetration Testing”, defines three requirements for a penetration test –

1. The test must have a defined goal and that goal should be clearly documented. The more specific the goal, the easier it will be to recognize the success or failure of the test.
2. The test should have a limited time period in which it is to be performed. The methodology in most penetration testing is to simulate the types of attacks that will be experienced in the real world. It is reasonable to assume that an attacker will expend a finite amount of time and energy trying to penetrate a site.
3. The test should have the approval of the management of the organization that is the subject of the test. This is extremely important, as only the organization’s management has the authority to permit this type of activity on its network and information systems[3].

Arpita Tewai and Arun Misra in the paper “Evaluation and Taxonomy of Penetration Testing” classified penetration testing into three types-

1. Black box — Testers do not have the full information about the internal environment of the organization. They have the data which is publically available. This type of test is used to perform a real pen test since attacker
starts with no knowledge about the system. It is also known as zero knowledge tests. Black box testing is best conducted from outside.

2. White box — Testers have full knowledge and information about the target system. Unlike black box in which experts are required to go through the code step by step and identify the faults due to which there may a possibility to attack. This is also called source code analysis because tester has full information about the application and source code.

3. Gray box — It is known as partial knowledge test and tester has partial knowledge about the organization [4].

II. PENETRATION TESTING TOOLS

Penetration testing tools are used as a part of a penetration test to automate certain tasks, improve efficiency and discover issues that might be difficult to find using manual analysis technique alone. Different approaches can be taken when choosing what toolsets can be used for performing individual phases of the penetration test. There are plenty of tools and toolsets for penetration testing and can be used for testing various types of products and conducting diverse types of attacks [11]. These tools are grouped under various categories based on the function they perform - Port scanning- It is a method in which a machine is scanned for open Transmission Control Protocol (TCP)/Universal Datagram Protocol (UDP) ports [5].

Vulnerability analysis- it is the process of 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 [12].

Vulnerability Exploitation- It requires special skills and techniques to launch attack on target system. Experienced penetration testers can use their skills to launch attack on the system [15].

Generally, machines are connected to a network and run many services that use TCP or UDP ports for communication with each other. There are 65536 defined ports on a computer. They can be categorized into three large ranges:

(i) well known ports (0-1023) 
(ii) registered ports (1024-49151)
(iii) dynamic and/or private ports (49152-65535).

Normally, a port scan does not directly damage the system, but potentially a port scan helps the attacker in finding those ports that are available to launch attacks. Essentially, a port scan consists of sending a message to each port, one at a time and listening for an answer. The kind of response received indicates whether the port is being used and can therefore be probed further for weakness to launch future attacks. Port scanning usually happens on TCP ports, i.e., ports that use a connection-oriented protocol; such ports return good feedback to the attacker. It also happens on UDP ports, but they provide connectionless services that may not readily give relevant information to attackers. Also, a UDP port may be easily blocked by network defenders [6].

The result of a scan on a port is usually generalized into one of the six categories-

Open- An application is actively accepting tcp connections, udp datagrams or sctp associations on this port. Finding these is often the primary goal of port scanning. Security-minded people know that each open port is an avenue for attack. Attackers and pen-testers want to exploit the open ports, while administrators try to close or protect them with firewalls without thwarting legitimate users. Open ports are also interesting for non-security scans because they show services available for use on the network [7].

Closed- A closed port is accessible (it receives and responds to Nmap probe packets), but there is no application listening on it. They can be helpful in showing that a host is up on an IP address (host discovery, or ping scanning), and as part of OS detection. Because closed ports are reachable, it may be worth scanning later in case some open up. Administrators may want to consider blocking such ports with a firewall. Then they would appear in the filtered state, discussed next [7].

Filtered- Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port. The filtering could be from a dedicated firewall device, router rules, or host-based firewall software. These ports frustrate attackers because they provide so little information. Sometimes they respond with ICMP error messages such as type 3 code 13 (destination unreachable: communication administratively prohibited), but filters that simply drop probes without responding are far more common. This forces Nmap to retry several times just in case the probe was dropped due to network congestion rather than filtering. This slows down the scan dramatically [7].

Unfiltered- The unfiltered state means that a port is accessible, but Nmap is unable to determine whether it is open or closed. Only the ACK scan, which is used to map firewall rule sets, classifies ports into this state. Scanning
unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open[7].

Open Filtered- Nmap places ports in this state when it is unable to determine whether a port is open or filtered. This occurs for scan types in which open ports give no response. The lack of response could also mean that a packet filter dropped the probe or any response it elicited. So Nmap does not know for sure whether the port is open or being filtered. The UDP, IP protocol, FIN, NULL, and Xmas scans classify ports this way [7].

Closed Filtered- This state is used when Nmap is unable to determine whether a port is closed or filtered. It is only used for the IP ID idle scan [4].

III. PORT SCANNING TOOLS OVERVIEW

Nmap- ("Network Mapper") is a free and open source utility for network exploration and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and both console and graphical versions are available [8].

Unicornscan- It is an open source (GPL) tool designed to assist with information gathering and security auditing. It is an attempt at a User-end Distributed TCP/IP stack for information gathering and their interrelation. It provides a superior interface for introducing a stimulus into and measuring a response from a TCP/IP enabled devices. The various features of this scanner includes asynchronous stateless TCP scanning with all variations of TCP flags, asynchronous stateless banner grabbing, and active/passive remote OS and component identification by analysing responses [13]. It provides Scalable, Accurate, and Efficient system scan. It is released for the community to use under the terms of the GPL license [14].

Dmitry (Deepmagic Information Gathering Tool) is a UNIX/(GNU)Linux Command Line Application coded in C. Dmitry has the ability to gather as much information as possible about a host. Base functionality is able to gather possible sub domains, email addresses, uptime information, tcp port scan, whois lookups, and more.

Hping3 is a command-line oriented TCP/IP packet assembler/analyzer. The interface is inspired to the ping(8) unix command, but hping isn’t only able to send ICMP echo requests. It supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features. It is fully scriptable using the TCL language, and packets can be received and sent via a binary or string representation describing the packets. In practice this means that a few lines of code can perform things that usually take many lines of C code.

IV. EVALUATION OF PORT SCANNING TOOLS

The four tools were assessed using eight criteria. In section A, the criteria are stated. Section B provides the actual assessment using tables and graphs.

A. Evaluation Criteria

- Number of ports scanned- This parameter lists the number of ports scanned from the total 65535 ports by the tool.
- Number of open ports found- This parameter lists the number of open ports found by the tool.
- Types of scan available- This parameter list the type of scanning mechanisms available-
  a) TCP SYN scan
  b) UDP scan
- Scan time- This parameter lists the total time taken by the tool to perform the scan in seconds.
- OS and service version detection- This parameter lists if the tool can detect the os version of the host(s) being scanned.
- Type of interface- This parameter lists the types of interface provided by the tool.
- Platform support- This parameter lists the platforms the tools can run on.

B. Evaluation Procedure

The above criteria are used to compare the four tools in question. The same approach will be used when new tools are added. The criteria were distributed among two tables, with four criteria per table. Depending on the criteria used, some cells contain yes/no, and others contain values. Table I – II illustrates the outcomes of the evaluation.
Table I Port scanning results part-1

<table>
<thead>
<tr>
<th>Ports scanned</th>
<th>Ports found open</th>
<th>Scan time (in sec)</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nmap</td>
<td>65536</td>
<td>9</td>
<td>105.77, Command line, GUI</td>
</tr>
<tr>
<td>Dmitry</td>
<td>150</td>
<td>2</td>
<td>3, Command line, GUI</td>
</tr>
<tr>
<td>Hping3</td>
<td>65536</td>
<td>9</td>
<td>110.13, Command line</td>
</tr>
<tr>
<td>Unicornsan</td>
<td>65536</td>
<td>4</td>
<td>10, Command line</td>
</tr>
</tbody>
</table>

Table II shows the result of port scanning performed on the victim system by the attacker system using four different tools. It can be seen that, Nmap scanned 65536 ports in 105.77 seconds and found nine open ports. The interface provided by Nmap is both command line as well as GUI. Similarly, the interpretation of Table III for the other tools can be inferred.

Table IVV Port scanning results part-2

<table>
<thead>
<tr>
<th>TCP SYN scan</th>
<th>UDP Scan</th>
<th>Platform</th>
<th>OS and Service version detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nmap</td>
<td>Yes</td>
<td>Yes</td>
<td>Linux, windows, Mac os x</td>
</tr>
<tr>
<td>Dmitry</td>
<td>Yes</td>
<td>No</td>
<td>Unix, Linux</td>
</tr>
<tr>
<td>Hping3</td>
<td>Yes</td>
<td>Yes</td>
<td>Unix, Linux</td>
</tr>
<tr>
<td>Unicornsan</td>
<td>Yes</td>
<td>Yes</td>
<td>Unix, Linux</td>
</tr>
</tbody>
</table>

Table VI also shows the result of port scanning performed on the victim system by the attacker system using four different tools. It can be seen that, Dmitry is capable of performing both TCP SYN scan and UDP scan. Also, Dmitry provides OS and service detection and can be used on unix and linux platforms. Similarly, the interpretation of Table VIII for the other tools can be inferred.

The above bar graph represent the time taken by each tool to perform port scan on the victim machine. It can be observed that, Hping3 took the maximum time to perform the scan while Dmitry took the least.
The above bar graph represents the number of open ports found by each tool after performing port scan on the victim machine. It can be observed that, Nmap and Hping3 found the most number of open ports while dmitry found the least.

V. OUTCOMES DISCUSSION

A number of interesting observations can be spotted in the above tables. Table I shows that nmap and hping3 were able to find the most number of open ports whereas Dmitry found the least. Also, Dmitry scanned the least number of ports while the other three were able to scan all the 65535 ports. With regard to interface available, only nmap provides an additional option for a GUI interface. The least amount of time taken to perform the scan was by Dmitry and hping3 took the maximum time.

Table II indicates that nmap, Dmitry, hping3 and unicornscan are all capable of performing TCP SYN scan. All tools except Dmitry can perform a UDP scan and have additional functionality of OS and service version detection.

Also, it can be observed that nmap is available for most platforms including windows.

The assessment exhibits that nmap is the superior tool, hping3 and unicornscan follow. Dmitry satisfy fewer criterions than rest.

VI. CONCLUSIONS AND FUTURE SCOPE

Penetration testing is one of the most efficient approaches for the process of security assessment. It can accurately examine the effectiveness of security measures implemented on the system being inspected. With a wide variety of supporting tools available in the market, it is confusing for practitioners to make proper informed decisions when looking for suitable tools. The research aims to provide the community more reliable references regarding the tools’ effectiveness by carrying out an evaluation on the performance of some particular tools. The experiment results have indicated that nmap is more powerful than the other selected tools owing to broad coverage, easy to use interface, fairly fast response time and highest number of detected open ports.

Future studies may continue the topic of penetration testing tools evaluation with more tools from different categories such as vulnerability assessment tools and vulnerability exploitation tools. Additionally, more representative characteristics of the tools like accuracy of port scanners in terms of false positive/negative rate, or usability, should be put under investigation. This will increasingly support the empirical evidence, as well as provide the security community more reliable references regarding the performance of penetration testing tools.

REFERENCES


