A Distributed Network Security Assessment Tool with Vulnerability Scan and Penetration Test

Abstract — As the Internet grows in scale almost every year, security measures are expected to become all the more important on the Internet. Vulnerability assessments provide us with a way to detect and resolve security problems before someone can exploit them.

The need for automated network assessment vulnerability tools combining with both penetration test and intelligent reporting has motivated this research. In this research, an automated tool was developed which features network mapping, vulnerability scanning, penetration testing, and intelligent reports.

Keywords: Vulnerability, penetration, hacker, exploit, migration

1 Introduction

Software security is an important property of any software system and is a hot topic within the media. Organizations that fail to integrate security throughout the development process will find their software suffers from faults both at the design level and in the implementation [7, 10]. While it is difficult to determine the exact cost of issuing a security fix, the Microsoft Security Response Center believes a security bug that requires a security bulletin costs in the neighborhood of US$100,000 [6].

Network administrators have to constantly defend against security holes inadvertently opened by users and managers. Network vulnerability [2, 3] assessment tools, such as SATAN [5] or Nessus [4, 8], locate live systems on a network and then analyze network services in use which are potential vulnerable. Vulnerability scanners are automated tools designed to scan hosts for known vulnerabilities and weaknesses. They look for such holes as misconfigured file permissions, open services, and other operating system problems. By using the information from the scanner in conjunction with the rest of the testing data, network administrator could gain an excellent picture of the network and systems.

However, one of the most common problems with these vulnerability assessment tools is that the false positive, the tools report a vulnerability which does not actually exist on the target systems. Vulnerability scanning usually result in many false positive scores, or it may not detect certain types of problems that are beyond the detection capabilities of the tools. Network administrator must be able to review and analyze the output to determine whether the vulnerability really applies or is a false positive.

Penetration test [1, 7, 10] is an effective complement to vulnerability scan, aimed at uncovering hidden vulnerabilities. The findings or results of the penetration test are aimed at improving the security posture of a network by presenting countermeasures for the vulnerabilities identified.

In this research, we developed a user-friendly implementation of automated network vulnerability assessment tool which is capable of distributively scanning for vulnerability and perform penetration test. Our software was developed mainly in Java language and provides an in-depth solution for system-wide network security assessment. It also generates credible reports for network mapping, vulnerability scan, and penetration test.

2 System Architecture

The main goal of our tool is to automate the security scan and testing procedures in order to reliably verify network security and detect network vulnerabilities. Our tool has two main components:

(1) Controller: This is a Windows application which controls assessment flow and commands Agents to perform assessment task. It provides a graphical interface for editing scanning policies. The scanning policies include all settings in network policy.
mapping, vulnerability scan, and penetration test. Controller also shows the performing progress of assessment talk and generates final reports when agents finished.

(2) Agent: This is a front-end application which provides network mapping, vulnerability scan, and penetration test. It received commands from the Controller and sends the results back to the Controller. Network administrator could considerably reduces overall completion time by setting up several Agents work independently at different physical locations so that the overall security assessment task could be shared by these Agents.

Fig. 1 shows system architecture of the developed system. Our tool uses the Controller/Agents architecture for security assessment. The main advantage of this architecture is the assessment convenience. User can deploy multiple agents across different crucial locations and performs multiple assessment tasks simultaneously. These agents then send the results back to the controller for generating reports on all target systems.

Another important advantage of our Controller/Agents architecture is that it uses result from penetration test to verify the result of vulnerability scan. Vulnerability scan usually have erroneous judgments. In the traditional vulnerability scan, human work to check the status of target system one after another is needed to verify the result. Our tool replaces human work by automatically invoking penetration test after vulnerability scan. If the vulnerability is a false alarm, penetration test will fail while attempting this vulnerability to attached related target.

3 Assessment Policies

Assessment policies are the configurations and rules which user wants the tool to perform. Policies consist of which networks user wants to scan, what network information user wants to retrieve, what kind of vulnerabilities user would like to check, and which exploit user want to test for penetration.

Our assessment policies consist of two main parts by the Controller software, Agent management and policy management. The Agent Management, illustrated on Fig. 2, configure Agent’s IP address, port, and target network. User can setup one or more Agents and control them simultaneously by the Controller; all Agents can perform networking mapping, vulnerability scan, and penetration test. User also can divide Agents into different groups so that one group performs network mapping and the others perform vulnerability scan or penetration test.

Fig. 2. Agent Management

On the other hand, the Policy Management consists of Network Mapping Policy, Vulnerability Scan Policy, and Penetration Policy. Network Mapping Policy, shown in Fig. 3, allow user to define various scanning options, such as network topology drawing, OS type detection, SNMP enumeration, port-scan range, and scan timing specification, etc.

Fig. 3. Network Mapping Policy

Vulnerability Scan Policy, shown in Fig. 4, let user select predefined profiles or select from plug-ins in categories for vulnerability scan. Predefined profiles including Microsoft Windows-related vulnerabilities, SANS [9] top 20 vulnerabilities, UNIX-related vulnerabilities, web server related vulnerabilities, etc. User can also select plug-in
4 Test Reports

After Agent completes assessment tasks, it will send the results of each task to Controller. Controller then shows the runtime results. The results of network mapping show the information about all network status. It consists of live hosts and network topology in target network.

When the scan tasks are completed, a report page will show the result with some charts and text report. In this page, user can also input asset value to do risk evaluation. The Report Page in our tool has three main functions: Risk Evaluation, Statistics Charts and Report Management.

User can set asset value for each target host depends on the host’s importance to evaluate the risk. The result of risk evaluation will be given at Risk Evaluation report. Fig. 6 gives an example of such report.

Pie chart and bar chart are provided for operating system (OS) statistic, vulnerability statistic, and service statistic. Pie chart of all OS statistic shows the situation of OS distribution. An example is shown in Fig. 7.

User can get an image of how serious the security defects exist in whole target network from Vulnerability Charts report, shown in Fig. 7.

Finally, Service Chart shows the situation of services distribution in whole target network, as illustrated in Fig. 8.
5 Conclusions

Securing today’s complex systems is challenging and demanding. Organization should conduct vulnerability scan to validate that operating systems and major applications are up to date on security patches and software version. Penetration test is an effective complement to vulnerability scan, aimed at uncovering hidden vulnerabilities.

This paper has presented a new implementation for automating both vulnerability scan and penetration test. The developed tool uses a distributed Controller/Agents architecture to improve overall performance. Test results can be reported in pie chart or bar chart.

Acknowledgement

This work was partially supported by the iCAST project sponsored by the National Science Council, Taiwan, under the Grants No. NSC95-3114-P-001-002-Y02 and NSC95-3114-P-307-003-Y. Also supported by NSC 95-2221-E-017-007 and NSC 95-3113-P-017-001.

References


