AN OVERVIEW OF VULNERABILITY SCANNERS

February 2008

TABLE OF CONTENTS

SUMMARY

A vulnerability scanner is software application that assesses security vulnerabilities in networks or host systems and produces a set of scan results. However, because both administrators and attackers can use the same tool for fixing or exploiting a system, administrators need to conduct a scan and fix problems before an attacker can do the same scan and exploit any vulnerabilities found. This article provides a general overview of vulnerability scanners.

I. BASICS OF VULNERABILITY SCANNERS

WHAT IS A VULNERABILITY SCANNER?

A vulnerability scanner can assess a variety of vulnerabilities across information systems (including computers, network systems, operating systems, and software applications) that may have originated from a vendor, system administration activities, or general day-to-day user activities:

1. Vendor-originated: this includes software bugs, missing operating system patches, vulnerable services, insecure default configurations, and web application vulnerabilities.
2. System administration-originated: this includes incorrect or unauthorised system configuration changes, lack of password protection policies, and so on.
3. User-originated: this includes sharing directories to unauthorised parties, failure to run virus scanning software, and malicious activities, such as deliberately introducing system backdoors.

THE BENEFITS OF VULERABILITY SCANNERS

Firstly, a vulnerability scanner allows early detection and handling of known security problems. By employing ongoing security assessments using vulnerability scanners, it is easy to identify security vulnerabilities that may be present in the network, from both the internal and external perspective.

Secondly, a new device or even a new system may be connected to the network without authorisation. A vulnerability scanner can help identify rogue machines, which might endanger overall system and network security.

Thirdly, a vulnerability scanner helps to verify the inventory of all devices on the network. The inventory includes the device type, operating system version and patch level, hardware configurations and other relevant system information. This information is useful in security management and tracking.

THE LIMITATIONS OF VULERABILITY SCANNERS

The drawbacks of vulnerability scanners are:

1. Snapshot only: a vulnerability scanner can only assess a "snapshot of time" in terms of a system or network's security status. Therefore, scanning needs to be conducted regularly, as new vulnerabilities can emerge, or system configuration changes can introduce new security holes.
2. Human judgement is needed: Vulnerability scanners can only report vulnerabilities according to the plug-ins installed in the scan database. They cannot determine whether the response is a false negative or a false positive[1]. Human judgement is always needed in analysing the data after the scanning process.

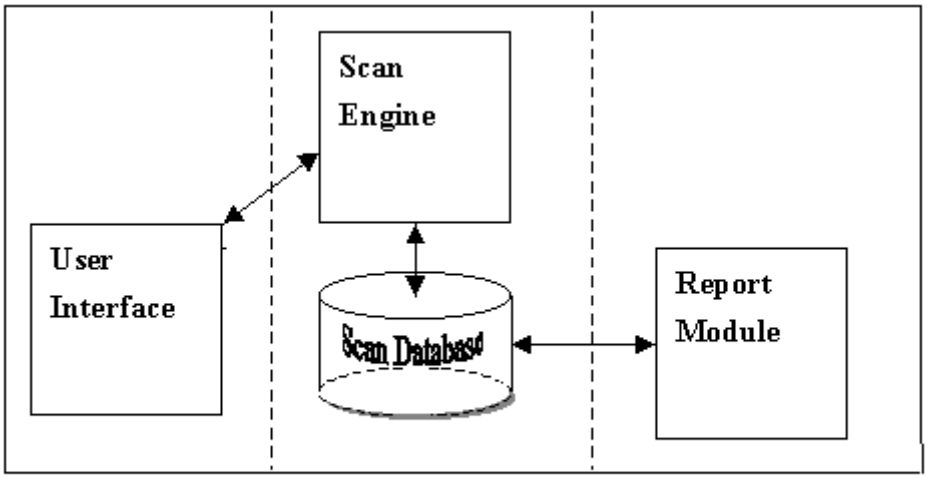
[1] Regarding vulnerability scanning, "false negative" is the failure to recognise an existence of a flaw in the system or the network under assessment, whereas "false positive" is the incorrect determination of the presence of vulnerability. The former might be due to missing plug-ins in a scanner database while the latter requires human judgement to confirm.

3. Others: a vulnerability scanner is designed to discover known vulnerabilities only. It cannot identify other security threats, such as those related to physical, operational or procedural issues.

In addition, many vulnerability scanners rely on "plug-ins" to determine potential vulnerabilities. Plug-ins are part of the knowledge database (or scan database) of the vulnerabilities that the scanner is capable of detecting. These databases may be named differently (such as "Scanning Profile") in different scanner products, but the term "plug-ins" will be used in this article. The finite number of plug-ins can be another drawback with vulnerability scanners. A scanner can only check for those vulnerabilities that it "knows", by cross checking with the presence of its corresponding installed plug-in set. It cannot identify those vulnerabilities that don't have a plug-in. Not all scanners need plug-ins. For example, port scanners do not need any plug-ins as they just scan a target range of ports.

II. THE ARCHITECTURE OF VULNERABILITY SCANNERS

In general, a vulnerability scanner is made up of four main modules, namely, a Scan Engine, a Scan Database, a Report Module and a User Interface.



Components of Scanner

1. The Scan Engine executes security checks according to its installed plug-ins, identifying system information and vulnerabilities. It can scan more than one host at a time and compares the results against known vulnerabilities.
2. The Scan Database stores vulnerability information, scan results, and other data used by scanner. The number of available plug-ins, and the updating frequency of plug-ins will vary depending on the corresponding vendor. Each plug-in might contain not only the test case itself, but also a vulnerability description, a

Common Vulnerabilities and Exposures (CVE)[2] identifier; and even fixing instructions for a detected vulnerability. Scanners with an "auto-update" feature can download and install the latest set of plug-ins to the database automatically.

3. The Report Module provides different levels of reports on the scan results, such as detailed technical reports with suggested remedies for system administrators, summary reports for security managers, and high-level graph and trend reports for executives.
4. The User Interface allows the administrator to operate the scanner. It may be either a Graphical User Interface (GUI), or just a command line interface.

Most vulnerability scanners are characterised by their modular structure as explained above. However, there are also primitive scanners that are basically sets of scripts or C-code exploits producing simple plain-text files as scanning results. Updates to these primitive scanners are infrequent and require manual intervention.

On the other hand, there are now a number of Distributed Network Scanners with more complex architecture. When enterprise networks are widely distributed, Distributed Network Scanners are used. They are composed of remote scanning agents, a plug-in update mechanism for those agents, and a centralised management point. Such scanners are capable of assessing vulnerabilities across multiple, geographically dispersed networks from one centralised management console, which can then distribute updates to scanning agents, modify settings in all scan engines, and schedule testing activities across the whole enterprise. Scan results are in turn collected from all remote scanning agents into the central database for analysis and reporting.

[2] http://cve.mitre.org/

III. TYPES OF VULNERABILITY SCANNER

Vulnerability scanners can be divided broadly into two groups: network-based scanners that run over the network, and host-based scanners that run on the target host itself.

NETWORK-BASED SCANNERS

A network-based scanner is usually installed on a single machine that scans a number of other hosts on the network. It helps detect critical vulnerabilities such as mis-configured firewalls, vulnerable web servers, risks associated with vendor-supplied software, and risks associated with network and systems administration.

Different types of network-based scanners include:

1. Port Scanners that determine the list of open network ports in remote systems;
2. Web Server Scanners that assess the possible vulnerabilities (e.g. potentially dangerous files or CGIs) in remote web servers;
3. Web Application Scanners that assess the security aspects of web applications (such as cross site scripting and SQL injection) running on web servers. It should be noted that web application scanners cannot provide comprehensive security checks on every aspect of a target web application. Additional manual checking (such as whether a login account is locked after a number of invalid login attempts) might be needed in order to supplement the testing of web applications.

HOST-BASED SCANNERS

A host-based scanner is installed in the host to be scanned, and has direct access to low-level data, such as specific services and configuration details of the host's operating system. It can therefore provide insight into risky user activities such as using easily guessed passwords or even no password. It can also detect signs that an attacker has already compromised a system, including looking for suspicious file names, unexpected new system files or device files, and unexpected privileged programs. Host-based scanners can also perform baseline (or file system) checks. Network-based scanners cannot perform this level of security check because they do not have direct access to the file system on the target host.

A database scanner is an example of a host-based vulnerability scanner. It performs detailed security analysis of the authorisation, authentication, and integrity of database systems, and can identify any potential security exposures in database systems, ranging from weak passwords and security mis-configurations to Trojan horses.

IV. CONSIDERATIONS

CHOOSING A VULNERABILITY SCANNER

The following factors should be considered when selecting a vulnerability scanner:

1. Updating Frequency and Method of Plug-in Updates

   Usually, a vulnerability scanner cannot identify a vulnerability if its corresponding "plug-in" is not available. As a result, the faster a vendor can produce updated and new plug-ins, the more capable a scanner is in spotting new flaws. Also, scanners with an "auto-update" feature can automatically download and install the latest plug-ins on a regular basis. This should be considered when choosing a vulnerability scanner.

2. Quality versus Quantity of Vulnerabilities Detected

   The accuracy with which critical vulnerabilities are identified is more important than the number of vulnerability checks, because the same vulnerability may be counted more than once by the scanner. The effective number of vulnerabilities in terms of Common Vulnerabilities and Exposures (CVE)[3] can be compared in a list of standardised names for vulnerabilities and other information security exposures. The content of a CVE is a result of a collaborative effort by the CVE Editorial Board.

3. Quality of Scanning Reports

   Apart from the details of detected vulnerabilities, a useful scanning report should give clear and concise information about fixing the problems uncovered. When administrators need to perform subsequent scans after initial scanning or

[3] http://cve.mitre.org/

configuration changes, or make comparison between the results of previous scans, a scanner with a back-end database that can keep an archive scanning results for trend analysis is preferable.

OPERATIONAL ISSUES

The following are issues that need to be considered before conducting vulnerability scanning:

Deployment Practices

Location of the Scanner (applicable to a network-based scanner)

Whether a scanner is located in front of or behind the firewall will have an effect on the scan result. Scanning an internal network from outside the firewall will only detect services that are available to the outside, but not vulnerabilities within the internal network that cannot be seen due to the protections provided by the firewall. On the other hand, scanning DMZ hosts from the inside may not provide a complete picture of the security position. Therefore, both external and internal scanning should be conducted in order to build a more complete picture.

Scanning Port Range (applicable to a network-based scanner)

Port scanning detects which ports are available (i.e., being listened to by a service). Because open ports may imply security weaknesses, port scanning is one of the basic reconnaissance techniques used by attackers. Therefore, security scanning should always include port scanning. However, some vulnerability scanners have a pre-defined default

port range set, such as only from port 0 to 15000. System administrators should be aware of these default settings and ensure all necessary ports are scanned.

Baseline Setup

The general cycle of a vulnerability scanning includes an initial assessment, implementation of recommended remedies, followed by a re-assessment. To verify the effectiveness of remedies, it is a good practice to keep archived logs of all scans (i.e. develop a working baseline), and compare the latest results with the baseline for trend analysis over time.

After-scan and Ongoing Practices

The scanning process itself is only part of a good assessment exercise. It is important to correctly interpret the scanning results so that valid vulnerabilities can be identified and subsequently fixed. The priority of necessary follow-up action should also be worked out and agreed upon.

In order to achieve this, vulnerability scanning and afterscan follow-up must be supported by solid security policies, so that vulnerabilities that are discovered will actually get fixed. In addition, a practice of continual scanning is essential. Systems should always be re-scanned after patches have been applied, any configuration changes, any installation of new software, or just on a fixed-time regular basis.

Precautions

Potential threats caused by the scan process

A scan itself can pose risks to IT systems by, for instance, crashing an already vulnerable server if all “plug-ins”, including high-risk ones (such as a DoS scan) are enabled.

Therefore, risk assessment and careful planning are necessary before scanning. Usually, for a pre-production system, it might be acceptable to enable all plug-ins including high-risk ones. However, for ongoing continual scans on a production system, administrators should consider disabling certain high-risk plug-ins.

In addition, when conducting scanning using a network-based scanner, a large amount of system requests and network traffic will be generated. The administrator should note any deterioration in the system and network performance of the target groups during scanning.

Handling of the scanning results

Leakage of scanning results, which contain system vulnerability information, may facilitate attackers in exploiting those loopholes directly. It is therefore important to safeguard this information by keeping it in a safe place, or keeping it encrypted to prevent unauthorised access. If an external party is employed for the assessment process, the organisation should ensure that any party involved is trustworthy, and that both findings and proprietary information will be kept secure.

Policies and procedures for the Scanning Process

Malicious or improper use of scanning tools could pose an enormous risk and cause tremendous harm to information systems. Therefore, policies and procedures should be in place to specify whom, how and when vulnerability assessment tools are to be used. Such policies may include the kind of prior arrangement or notifications, management approval and/or legal clearances that are required before a scanning takes place. No one should be allowed to conduct any vulnerability scanning without prior permission.

EXAMPLES OF COMMON VULNERABILITY SCANNERS

A number of open source freeware or commercial vulnerability scanners are available for download or trial. The following are examples:

1. Network-based scanners
    a. Port scanners
        - Nmap : http://insecure.org/nmap/
        - Superscan:

          http://www.foundstone.com/us/resources/proddesc/superscan4.htm
    b. Network vulnerability scanners
        - Nessus : http://www.nessus.org/nessus/
        - GFI LANguard Network Security Scanner (N.S.S.) (commercial) : http://www.gfi.com/languard/
    c. Web server scanners
        - Nikto : http://www.cirt.net/code/nikto.shtml
        - Wikto : http://www.sensepost.com/research/wikto/
    d. Web application vulnerability scanners
        - Paros : http://parosproxy.org/index.shtml
        - Acunetix Web Vulnerability Scanner (commercial) :

          http://www.acunetix.com/
2. Host-based scanners
    a. Host vulnerability scanners
        - Microsoft Baseline Security Analyser (MBSA)

          http://www.microsoft.com/technet/security/tools/mbsahome.mspx
        - Altiris SecurityExpressions (commercial) :

          http://www.altiris.com/Products/SecurityExpressions.aspx

b. Database scanners

- Scuba by Imperva Database Vulnerability Scanner:

  http://www.imperva.com/application_defense_center/scuba/default.asp

- Shadow Database Scanner

  http://www.safety-lab.com/en/products/6.htm

Important Note: please carefully review the relevant terms and conditions before registering on any website, as well as downloading and installing any software. In addition, please note that running a scanner tool can carry its own inherent risks (e.g. in the case of denial of service scans, you may crash a vulnerable server). It is necessary to plan and perform the scanning carefully. Prior arrangement or notification, such as management approval and/or legal clearance has to be obtained. For obvious reasons, never scan any network that is not your own.