CPNI
Centre for the Protection
of National Infrastructure

CPNI TECHNICAL NOTE 04/2008

VULNERABILITY ASSESSMENT TOOLS

DECEMBER 2008

CPNI would like to acknowledge and thank NCC for their help in the preparation of this report.

Disclaimer:

Reference to any specific commercial product, process or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favouring by CPNI. The views and opinions of authors expressed within this document shall not be used for advertising or product endorsement purposes.

To the fullest extent permitted by law, CPNI accepts no liability for any loss or damage (whether direct, indirect or consequential and including, but not limited to, loss of profits or anticipated profits, loss of data, business or goodwill) incurred by any person and howsoever caused arising from or connected with any error or omission in this document or from any person acting, omitting to act or refraining from acting upon, or otherwise using, the information contained in this document or its references. You should make your own judgement as regards use of this document and seek independent professional advice on your particular circumstances.

Contents

1. Introduction

This CPNI technical note is aimed at organisations in the UK’s national infrastructure that are looking to assess the extent of their vulnerability to common attacks. It is vendor-independent and does not promote the use of any specific technologies. However, some of the web references in this paper contain advice from particular vendors or from third-party companies. The tools in this technical note are not a substitute for professional vulnerability testing, such as that provided by CREST or TIGERScheme members.

The intention is to increase technical awareness about vulnerability assessment tools and promote their use within the Critical National Infrastructure (CNI). This paper does not provide guidance on specific tools or how to use them.

The note is intended to serve managers, system administrators and security officers within organisations who are interested in or maintain an IT security strategy.

The motivation for this paper is to promote the use of automated tools that can aid an organisation’s system administrator in the identification, analysis and management of vulnerabilities that may be present on their computer systems or networks.

2. Legal use of vulnerability assessment tools

Vulnerability assessment tools should only be used on networks and computer systems with the permission of the owner of those networks and systems. Unauthorised access to, or unauthorised modification of, computer systems is a criminal offence in the UK.

3. Vulnerability defined

A vulnerability in IT security is a group of conditions which, taken together, can leave a system open to unwanted access or unauthorised use by an intruder, denies availability of the system or otherwise represents a violation of the system security policy. A vulnerability can relate to the system security policy, system design, implementation or configuration. Hence, a vulnerability could be one of the following:

- An error or bug in software or hardware, allowing the system to be used in a way not intended (for example, buffer overflows allowing execution of arbitrary code).
- A design flaw which can allow security to be subverted or bypassed (for example, being able to change a password without needing to know the previous password).
- A misconfiguration of software or hardware, allowing the system to be used in a way not intended (for example, default accounts and passwords enabled allowing unauthorised access).

With the increase in software designed for both internal and external use, there are a large number of attack vectors available to somebody who is attempting to compromise a host/service. Attackers are often opportunistic with much of their success being attributed to poorly designed and maintained networks together with the successful exploitation of known vulnerabilities.

4. Vulnerability assessment tools defined

A vulnerability assessment tool (or scanner) can be defined as a utility that can be used to test the capability of a system's or network's security and discover points of weakness. These tools do not provide direct protection or security for a system or network but instead they gather and report information, such that some other mechanism, policy or tool can be put in place to provide protection against any vulnerability found.

Vulnerability assessment tools typically can be classified as follows:

- Host
- Service
- Application
- Active/passive

Host

Host based tools scan and report on the computer where they reside; they have no interaction with other systems. The advantage of this architecture is that the scanner has complete access to all the system's resources such as logs and file systems. The main disadvantages are that the scanner may also take up a large amount of the host's resources and running these scanners individually across a large network can be especially resource intensive.

Service

Network vulnerability scanners cover tools which are designed to scan a range of hosts and the network services they offer to identify vulnerabilities. These tools range from simple port scanners which will identify open ports through to automated scanners which will detect live hosts and attempt to extract data and vulnerability information from them.

The more advanced tools usually allow the user to specify the types of vulnerabilities they are looking for and generate a report once the scan has completed. If you are after a specific type of service there are a large number of tools designed for just this purpose (for example database vulnerability scanners).

Application

Application scanners break down into two main areas: web based and host based. Both types of applications can either be scanned at source code or runtime level.

There are a number of generic types of tool available. These can be simple mapping tools which help you locate pages which shouldn't be there or shouldn't be accessible, or they can be more complex and attempt to manipulate the application in such a way it provides additional information or access. Examples of manipulation of the application may be trying to inject Structured Query Language (SQL) queries in an attempt to bypass security or gain access to the information stored within the application.

As well as the generic tools there are a number of frameworks available which will allow you to perform more in-depth tests. An example of these is "fuzzers" which provide random input to a system interface, and robustness test suites, and may use knowledge of the application or protocol structure and likely points of failure.

Vulnerability scanners can also be active or passive (or use both techniques) depending on their method of scanning. Active and passive scanners can be host or application scanners as well as network or service scanners.

Active scanners

Active scanners attempt to compromise (or evaluate the possibility of compromise) a network, specific system or service by using attack strategies that may be used in real attacks. Many active scanners identify missing system patches or configuration vulnerabilities. The reason for calling these scanners "active" is that they will actually perform tests that consume resources on the network.

An advantage of active scanners is that the administrator has a great deal of control of the timing and the extent of vulnerability scans. Another advantage is that, in some cases, the presence of a vulnerability can be determined by the success of a test (for example, if the scanner guesses an account name and password).

The major disadvantage of active scanners is that they can cause systems to hang or crash. Active scanners should not be used on critical operational systems without careful planning. Active scanners also, by definition, use system resources which may affect the processing of other tasks.

Passive scanners

Passive scanners on the other hand tend not to affect system resources significantly as they only monitor data on the system and tend to perform any data processing on a separate analysis machine. Passive scanners behave similarly to intrusion detection systems in that they intercept or receive system data and evaluate it against a set of rules. The analysis provides information about what processes are running on the system.

An advantage of this type of scanner is that passive scanners can provide an “always-on” capability because they do not consume a large system overhead. A disadvantage is that passive scanners will only reveal information derived from readily available data, some of which is usually inferred and may therefore be inaccurate.

The types of tests typically used by a passive scanner are: determination of the version of a program to check for the presence of the vulnerability (for example whether it has been patched or not), and checks for the presence of a program that has not been seen before on the system. An example of a passive vulnerability detection signature is the analysis of an email server Simple Mail Transfer Protocol (SMTP) banner when a connection is made to a server from an existing e-mail client to see if that servers running a vulnerable version of SMTP.

5. Current tool availability

There are a number of commercial and non-commercial vulnerability scanners available. Some of the scanners that are either open source or made available free of charge are:

- Nikto, open source web server scanner, http://www.cirt.net/code/nikto.shtml
- Microsoft Baseline Security Analyser, Microsoft's host based scanner for Microsoft Windows, http://www.microsoft.com/technet/security/tools/mbsahome.mspx
- Winfingerprint, open source network based vulnerability scanner for Microsoft Windows, http://winfingerprint.com/index.php
- WinVulnScan, free host based vulnerability scanned for Microsoft Windows, http://www.pspl.com/download/winvulscan.htm
- CIS Benchmarks/Security Tools, free host based UNIX, Microsoft Windows, Sun Solaris and Oracle vulnerability scanner, http://www.cisecurity.com
- OpenVAS, Open Vulnerability Assessment System, http://wald.intevation.org/projects/openvas/
- Paros, web testing framework, http://parosproxy.org/

The following web pages also provide details of vulnerability assessment tools that are currently available:

http://sectools.org/
http://www.securitywizardry.com/scanners.htm
http://chiht.dfn-cert.de/functions/proactive_tools.html

The majority of vulnerability assessment tools will be able to find well-known vulnerabilities within a system and these are the types of vulnerability that attackers will easily exploit and will be most widely exploited. It is important that any vulnerability scanner of appropriate type can identify such vulnerabilities and provide a score or rating based on the risk presented. However, vulnerability tools are generally reactive and will only find known vulnerabilities. Some tools (such as robustness testing tools) may have the ability to detect new vulnerabilities using heuristics.

It is important to note that some vulnerability assessment tools can reproduce real attacks which could compromise your system or network if they are not protected. Clearly such tools should not be used on an operational system or in any environment where the effect of the attacks is not known, but as these tools are available to attackers a conscientious system administrator should ensure that their system and network can not be compromised by such means. It would be prudent to use active scanners on a test or reference version of the system.

6. Types of vulnerability

Vulnerabilities can exist in network service software or software that is able to receive data from the network. Internet worms and remote system compromises exploit network service vulnerabilities. Other vulnerabilities are local to a particular application running on the host or require unmediated access to the operating system or file system. All of the types of vulnerability below apply to network service vulnerabilities and to local vulnerabilities. Network based scanners are used to detect network service vulnerabilities, while host based scanners detect local vulnerabilities. The types of vulnerabilities are typically:

Unpatched software - The majority of worms, viruses and remote attacks on computer systems exploit known vulnerabilities in computer software. In order to protect against these attacks it is therefore important that automated scanning tools are available that can identify systems on the network that are not completely patched.

Misconfiguration - Organisations choose differing configurations due to the requirements of their IT usage, but it is important that the configuration chosen is secure and that any possible vulnerability in this configuration is highlighted (for example, firewalls with incomplete rule sets, weak passwords on user accounts or files with incorrect access permissions).

Default configuration - A number of products, both hardware and software, have insecure default ‘out of the box’ configurations which can include well-known default user accounts and passwords, the running of unnecessary services and limited logging.

Unsupported applications - There are two types of unsupported applications: software that is out of date and no longer supported by the vendor (and hence patches will no longer be provided for any vulnerabilities in that software), and software that is not permitted by the user’s organisation. Some types of vulnerability are much harder to detect unless they have a known signature or affect particular versions of a software product and not others.

Programming input validation errors - This type of vulnerability includes buffer overflows, format strings and SQL injection attacks amongst others. It is by far the most common type of vulnerability used in remote system compromises.

Kernel vulnerabilities - Operating system kernels may have vulnerabilities that can be exploited through programming interfaces. Current vulnerability scanners are unlikely to identify these vulnerabilities at all unless the kernel version is known to be vulnerable. The only complete method for checking for these types of vulnerability is source code review - if the source code is available - which can be supported by automated code analysis tools. Otherwise the use of disassemblers or run time analysers (whether recording network flows or debuggers attached to a process) may be required. Source code review, behavioural analysis and reverse engineering are, to various degrees, resource intensive and require specialist skills.

7. Criteria for evaluating vulnerability assessment tools

When using or purchasing vulnerability assessment tools the following criteria provide a guide to product selection and effectiveness:

Types of vulnerabilities being assessed - The most important step in evaluating any vulnerability assessment tool is to determine the types of vulnerabilities (see above for a description) that it will discover. The tool itself should provide this information but you would not expect, for example, a network based scanner to detect file permission misconfigurations.

Reporting abilities - As vulnerability assessment tools are information gathering tools their reporting capabilities are very important. Reports should be clear and provide steps to mitigate or remediate the vulnerability. A problem that these tools face is the need to present clear information about what vulnerabilities exist in the system scanned and the criticality of those vulnerabilities. Vulnerability scanners should lay down criteria for criticality of vulnerabilities and ideally should allow the user to change the criteria. A common method of rating vulnerabilities is Common Vulnerability Scoring System (CVSS). CVSS is a vendor agnostic, industry open standard. It is designed to convey vulnerability severity, help determine urgency and priority of response.

It should be stressed that the overall business risk posed by any of the issues found in any test is outside a tool's remit. This means that some risk may be reported by a tool as high from a technical perspective but may, as a result of other controls, be considered acceptable to the business. The opposite is also true.

CVSS is described in detail at http://nvd.nist.gov/cvss.cfm.

Configuration options - The ability to select which types of vulnerabilities you wish to scan for. This allows you to target specific software and turn off vulnerabilities that are known to crash or disrupt a system.

Completeness of scan - Vulnerability assessment tools should execute all tests that have been selected and, where applicable, should scan all of the systems selected to be scanned.

Accuracy of scan - When evaluating the accuracy of a scan the total of false positives and false negatives is a simple measure (the larger the number the less accurate the scan). A "false positive" is a vulnerability that is reported that does not actually affect the version of the product tested. An example of a false positive is where a vulnerability only relevant to one application is reported against another. Reports of the vulnerability behind the "Code Red" worm affecting an Apache web server on a UNIX based operating system illustrate this ("Code Red" only exploiting the Microsoft Internet Information Services (IIS) web server).

Determining that a report is a false positive requires expertise and access to other sources of evidence: the system logs in the case of a host based scan and network traffic capture in the case of a network based scan. A “false negative” is a vulnerability that does affect the version of the product tested and is not reported by the scanning tool. This could occur because the tool does have an appropriate test or the test is incorrectly coded. False negatives significantly undermine confidence in the tool because the tool has not made it apparent to the user that a possible weakness in security exists. The way that false negatives can be detected include comparing the results of several scanners, a manual assessment by a penetration tester, or, if you are unfortunate, a successful intrusion.

Performance (Efficiency & Effect on scan) - There are two aspects to performance of vulnerability scanning tools. Firstly there is the performance of the scan itself such as how long it takes per host scanned and what resources are required specifically for the scanning engine. Secondly there is the performance degradation or loss of service that may be seen on the system being scanned. Many scanners offer differing intensity and potential destructiveness of scanning. As discussed above, the importance of degradation caused by a scanner is associated with active vulnerability scanners because passive vulnerability scanners monitor the system and analyse the results on a separate host.

Ability to perform intelligent searches - A feature that a few vulnerability scanners are now offering is the idea of searching for vulnerabilities by means other than test signatures. This is where the scanner does some kind of intelligent attack on the system until it compromises the system. An example of this is Common Hacking Attack Methods (CHAM), a proprietary technology used in eEye’s commercial RETINA scanner, which claims to employ common techniques such as buffer overflows and SQL injection that have no specific signatures but do have a methodology that can be followed to attempt to exploit these types of vulnerabilities. (As noted above, this type of test can be dangerous to your system).

Updates - It is important that a new test that detects a vulnerability is produced as soon as possible after the vulnerability is published and that the test is updated to reflect any exploits of the vulnerability that become available. In terms of making these tests available to users automatic download to the users’ site is the quickest way of providing the tests, but availability on a web site or via an update CD-ROM is not uncommon.

Functionality for writing own tests - When robust signatures are not yet available for a newly discovered vulnerability it is useful if the vulnerability scanner allows for user-developed tests to be used, at least until a definitive test becomes available. This functionality also allows for an administrator to write bespoke scanning signatures that may not be required by other organisations. Some scanners, such as Tenable Nessus (commercial), have a large community of test authors which helps to keep the tests current.

Scheduling - As these tools perform automated scans the opportunity to schedule the test runs is important, in order that scans can be run during quiet periods on the network such as the early morning without needing operator control. It is worth recalling that while active vulnerability scanners take a snap shot at a specific test-time, a passive scanner can continuously monitor a network.

Compliance with standards - Information sharing within the IT security community is an important objective. Being able to conform to standards makes information sharing easier. Two standards that are related to the vulnerability area are Common Vulnerabilities and Exposures (CVE) and Open Vulnerability Assessment Language (OVAL).

CVE is a standard for vulnerability content (see http://cve.mitre.org/cve). Vulnerabilities that are in CVE have been validated and are judged as significant by a group of experts. A number of vulnerability assessment tools provide CVE identifiers, where available, for the vulnerabilities they identify.

OVAL is a vulnerability description language that can be used to specify the conditions for detecting the presence of a vulnerability in a product (see http://oval.mitre.org). Vendor-approved implementations of OVAL are available for Microsoft Windows NT 4.0/2000/XP and 2003 Server and for Red Hat Linux and Sun Solaris.

8. Best practices for using vulnerability assessment tools

Vulnerability assessment tools are detection tools and it is only through understanding, analysis and action by a system administrator that they become vulnerability protection tools. These tools all require a high degree of technical expertise and time to understand the alert, establish that it is not a false positive and undertake the required action. The use of vulnerability scanning tools is not a substitute for using qualified penetration test teams, for example those accredited by CREST or by TIGERScheme.

Vulnerability scanners are used to help protect the system or network so be sure that running these tools does not have the opposite effect, damaging your system or network or degrading performance. It is recommended that before deploying a new vulnerability tool or tests for that tool that they are run against a non-production system or network. It is also recommended that destructive tests are disabled when running scans on an operational network. Note that these recommendations do not apply to purely passive vulnerability scanners. Some commercial tools offer independent test reports of their performance in terms of false negatives and positives. Be sure that systems you wish to use the tool against are covered by the report.

Before using any of these tools it is essential to understand how they work and establish what information that you intend to gather by using them. The information in this technical note should help achieve this end.

The location source of the scan needs to be considered because the security mechanisms are likely to be different for access from outside the network boundary than for access from within. The decision where to locate the source of the scan depends on what information you are trying to gather. An externally-based network scan will to some extent emulate an external attacker. An internally-based scan will more accurately reflect the vulnerabilities present in the system that could be exploited by an insider or an attacker who had gained entry to a host on the internal network.

When performing scans, logging should be enabled on all the computers being scanned. If the scan is a network scan, traffic from the network should be collected. As noted above, these are mechanisms for validating the results of a scan to determine if a vulnerability report was correct or if it was a false positive. When undertaking a vulnerability scan ensure that all results and methodologies are annotated as this will allow you to quickly disregard false positives that have been seen before, as well as provide you with information of historical vulnerabilities you discovered and how they were dealt with.

It is useful to have the ability to be able to write your own tests for a vulnerability scanner, as this will allow you to write temporary tests for new vulnerabilities to use while the vendor is creating a definitive test. Writing your own tests also allows for scanning systems that may be bespoke to your organisation. Being able to write your own signatures is also a way to deal with any false negatives that you discover.

Users of vulnerability scanners should regularly scan their systems for vulnerabilities. This will enable them to provide a baseline for vulnerabilities. Furthermore, users of vulnerability scanners should proactively monitor for vulnerabilities and exploits (see information sources below) and should apply tests for those vulnerabilities as soon as they are available. Changes made to systems as a result of vulnerability assessments should be implemented using the organisation’s change and configuration management processes.

Alternatives

One alternative to using vulnerability assessment tools is to have your system and network tested by a specialist penetration test team. There are many available penetration test teams across the world, including a number in the UK who belong to the UK government’s CHECK scheme, the commercial CREST scheme or to TIGERScheme.

http://www.cesg.gov.uk/products_services/iacs/check/index.shtml
http://www.crest-approved.org/
http://tigerscheme.org/

Another alternative is to use managed vulnerability assessment services, which provide remote scanning of systems' perimeters or networks. The advantages of these services are that they administer the vulnerability scanning and are specialists in this providing this kind of service. The problem with vulnerability assessment services is that they will not have the same understanding of your system and business requirements as your system administrator.

9. Further vulnerability information

There are numerous resources on the Internet for monitoring new vulnerabilities, the following offers a sample:

- CERT/CC (http://www.cert.org)
- US CERT (http://www.uscert.gov)
- CVE (http://cve.mitre.org/cve and http://icat.nist.gov)
- SANS Internet Storm Centre (http://isc.sans.org/)
- OSVDB (Open Source Vulnerability Database) (http://www.osvdb.org)
- Bugtraq (available via http://www.securityfocus.com)
- Full Disclosure mailing list (http://seclists.org/#fulldisclosure)
- VulnWatch (http://www.vulnwatch.org)
- Secunia (http://secunia.com/)
- Milw0rm (http://www.milw0rm.com)
- Metasploit (http://www.metasploit.com)

10. Summary

- The first step to protect against a particular vulnerability is to assess whether it affects your system or network. The task of monitoring for new vulnerabilities is something that system administrators should be doing on a day-to-day basis, so they may find it helpful to use automatic tools to identify these vulnerabilities on their systems or networks.
- Vulnerability scanning is not a complete network security solution but it is necessary to ensure that the security mechanisms in place protect your systems from the majority of attacks such as computer viruses and email trojans. Only the most sophisticated vulnerability assessment tools are likely to discover new vulnerabilities. Vulnerability scanning should be regarded as complementary to manual assessments performed by independent vulnerability testers such as those accredited under the CHECK or CREST schemes.
- There are a number of different types and architectures of vulnerability assessment tools available. It is important to establish the information you require from the tool used and the impact it will have on the network or system being scanned. It may be necessary to put a variety of these technologies in place to perform different levels of vulnerability scan.
- Vulnerability assessment tools allow system administrators to take a proactive attitude towards IT security in that they will attempt to elicit or deduce details of the current security mechanisms, whereas firewalls and intrusion detection systems take a more reactive view towards security. It is important to have active security strategies as well as passive strategies, so the system administrator can be assured of the security mechanisms in place.
- Vulnerability testing is also part of ISO 27001 compliance see (para. A6.1.8) The frequency will be business specific, and vulnerability tests should be performed at least annually. The use of professional vulnerability testers is strongly recommended.