9

ADDRESSING INTERNET SECURITY VULNERABILITIES

A Benchmarking Study

A.ALAYED, S.M.FURNELL and I.M.BARLOW
Network Research Group,
Department of Communication and Electronic Engineering,
University of Plymouth,
Plymouth,
United Kingdom,
Tel: +44 1752 233521,
Fax: +44 1752 233520,
Email : nrg@jack.see.plym.ac.uk

Keywords: Vulnerability, Exploits, Advisories.

Abstract: The exploitation of vulnerabilities in operating systems and applications has become a frequent and increasing problem in IT environments. This paper assesses the extent of the problem by examining the scale of vulnerability reports issued by a number of popular advisory sources. It then proceeds to determine the workload implications that this introduces from system administrators, benchmarking the number of vulnerabilities that would need to be addressed and patched within a reference environment over a 12-month period. It is concluded that further advances are required in order to facilitate more targeted vulnerability notification, and where possible, the automated rectification of the problems themselves.

1. INTRODUCTION

Security vulnerabilities are a known problem that affect operating systems, Internet servers and application programs, and may take many forms (e.g. old or unpatched software, poorly configured servers, disabled controls and badly chosen passwords [1]). Their exploitation can enable various breaches, including attacks such as denial of service and defacement of web sites, and malware incidents such as viruses and worms [2].

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35586-3_46

M. A. Ghonaimy et al. (eds.), *Security in the Information Society*

A security vulnerability is defined as "a flaw in a product that makes it infeasible – even when using the product properly – to prevent an attacker from usurping privileges on the system, regulating its operation, compromising data on it, or assuming ungranted trust" [3]. There are three types of technical vulnerabilities, namely [4]:

- Flaws in software or protocol design.
  A fundamental design flaw, in which security is left out of the initial description and is later "added on" to the system.
- Weaknesses in how protocols and software are implemented.
  Even when a protocol is well designed, it can be vulnerable because of the way it is implemented. Software may be vulnerable because of flaws that were not identified before the software was released.
- Weaknesses in system and network configuration.
  Vulnerabilities in this category are not caused by problems inherent in protocols or software programs, but as a result of the way in which these components are set up and used.

As an example of vulnerability exploitation, the summer of 2001 witnessed the appearance of Code Red, a worm program that exploited a vulnerability in Microsoft's Internet Information Server (IIS). Security company eEye Digital Security discovered the flaw in IIS on 18 June 2001, and warned that an exploit could soon be created to take advantages of the vulnerability. The Code Red worm was such an exploit and reportedly infected around 300,000 computers during the following month. A notable point was that an appropriate patch to eradicate the IIS vulnerability was made available and widely publicized. The victims of infections were, therefore, the individuals or organizations that had still not heard of, or managed to heed, the warnings.

It has been identified by previous wok [8] that a vulnerability will generally pass through several distinct stages, as part of an overall lifecycle, namely birth, discovery, disclosure, release of a fix, publication, and automated exploitation. Intrusions based upon the vulnerability will increase once the problem is discovered, and the rate will continue to increase until an appropriate solution (e.g. a software patch) is released. Figure 1 depicts this lifecycle.

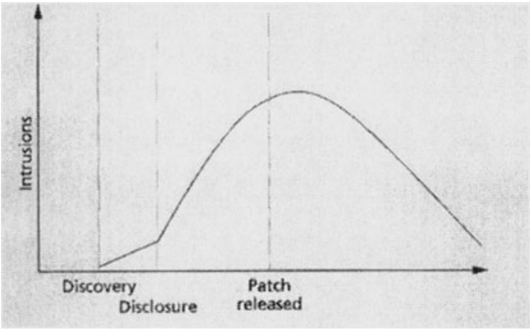

Figure 1 : Lifecycle of a system-security vulnerability [8]

The problem of dealing with vulnerabilities is compounded by a number of factors, as identified by the SANS Institute [6]:

- 1.2 million new computers are added to the Internet every month;
- there is a lack of security experts to address the problems;
- vulnerabilities are increasing, without a priority list for dealing with them.

As a result, system administrators face a significant challenge in combating a growing problem, particularly considering that for many of them security maintenance will be only one of a range of responsibilities. Although vulnerability scanning tools are available to analyze the systems and software configurations for known vulnerabilities, this only helps to address part of the problem, and still relies upon administrators being aware of and utilizing the tools – otherwise their availability may simply represent an open invitation to hackers [7]. The issues that the scanners do not address are, firstly, that administrators must still maintain awareness of newly emerging vulnerabilities (e.g. to determine when they might need to update their scanners or take exceptional action to protect their systems), and secondly, that finding a vulnerability leads to the obvious requirement to fix it, thus requiring an additional activities. Although there are solutions to these aspects, they typically relate to manual actions by the administrator.

This paper aims to investigate the scale of the vulnerability problem and the associated workload implications for administrators. The main discussion begins by considering statistics from a number of well-known vulnerability lists, enabling the annual totals and monthly averages to be traced over recent years. This is followed by the discussion of findings from a benchmarking study, which aimed to determine how such statistics would influence administrator workloads within a small network environment. The implications of the findings are then considered, leading to the recommendations of key system administration tasks that could usefully be automated.

2. A COMPARISON OF VULNERABILITY DATABASES AND STATISTICS

In this section, the scale of vulnerabilities from recent years will be considered, by examining a selection of publicly accessible sources that maintain repositories of the associated warnings and advisory reports. There are many such lists and databases, most of which are made available for online public access. At the top level, these sources can be categorized according to whether they are provided by a specific product vendors, such as Microsoft or Sun, or independent security advisory groups.

In many cases anyone can report a vulnerability to the operator of the vulnerability database, who in turn will do some preliminary analysis and forward a resulting report to the affected vendors as soon as is practical. In some cases, the database staff work with vendors on understanding the cause of the vulnerability and facilitate the development of solution. They also send vulnerability information to others who can contribute to the solution [3]. For example, in the case of vulnerabilities reports issued by the Computer Emergency Response Team Coordination Center (CERT/CC), details are usually disclosed to the public after 45 days of the initial report. The schedule might be shortened if a threat is especially serious or they have evidence of exploitation already having taken place. Conversely, the schedule might be lengthened if a threat requires "hard" changes, such as modifications to core operating system components [8].

Three well-known vulnerability advisory sources are analyzed in this paper, namely those from CVE, CERT/CC, and BugTraq, all of which provide vendor-independent services. These are considered in the subsections that follow.

2.1 Common Vulnerabilities and Exposures (CVE)

CVE is a list of information security vulnerabilities that aims to provide common names for publicly known problems [9]. The goal of CVE is not to provide a database in its own right, but rather to make it easier to share data across separate vulnerability databases and security tools by providing a common enumeration. After a vulnerability is discovered and reported, it is assigned a CVE candidate number (CAN) and proposed to the CVE Editorial Board for consideration. The board then discusses the new vulnerability and votes on whether it should become a full CVE entry. If the candidate is rejected, the reason for rejection is noted in the Editorial Board Archives posted on the CVE Web site. If the candidate is accepted, it is entered into CVE and is published via the site, along with a description, and the candidate number is converted into a CVE name [10].

2.2 CERT/CC

The CERT/CC is a major reporting centre for Internet security problems. It analyzes product vulnerabilities and maintains a searchable database of problems. CERT/CC is part of the Networked System Survivability (NSS) program at the Software Engineering Institute (SEI), a federally funded US research and development centre operated by Carnegie Mellon University. CERT/CC was established in 1988 (the catalyst for the formation having been the infamous 'Internet Worm' incident of the same year [11]) and the website maintains advisories dating back to this point [12]. The information released by CERT/CC can be divided into three categories: Advisories, Incident notes and vulnerability notes. CERT Advisories are limited to vulnerabilities that meet a certain severity threshold, Incident notes contain information that does not meet their criteria for alerts, but that might be useful to the Internet community, and finally vulnerability notes are very similar to advisories, but may have incomplete information. In particular, solutions may not be available for all vulnerabilities in the database.

2.3 BugTraq

BugTraq describes itself as "a full disclosure moderated mailing list for the detailed discussion and announcement of computer security vulnerabilities: What they are, how to exploit them, and how to fix them" [13]. Since its original inception in 1993, the list has grown to encompass over 27,000 subscribers, and includes information relating to vulnerabilities, exploits and fixes for a wide variety of operating systems and applications.

As with the CERT offering, the database is completely searchable by vendor, title (product name, technology, etc), keyword, and CVE ID number, allowing users to easily find the information they need. The database is hosted by SecurityFocus.com, but is also licensed to security product and service vendors to allow them to create information resources for their employees and customers.

A significant point to note in the case of BugTraq, in comparison to the other two sources discussed, is that vulnerability or patch information is not verified or validated before inclusion in the list. As a result, potential users are warned not to assume that the information is correct, and are advised to wait until it is verified by other subscribers if they are not in a position to do so themselves.

2.4 Vulnerability Statistics

Having introduced a number of the key information repositories, it is now relevant to examine the number of incident reports or advisories that they make available for security-conscious system administrators to consider. Figure 2 presents statistics relating to the total number of vulnerabilities reported each year, in the period from 1995 to 2001 (in the case of BugTraq the 2001 figures cover the first quarter only, whereas for CVE and CERT they encompass the second quarter as well). The statistics are based upon the three databases described above, although it should be noted that the CVE archives did not commence until 1999, and BugTraq figures prior to 1997 could not be located. The CVE figures are for reported vulnerabilities and hence include candidates that were not accepted as full CVE entries.

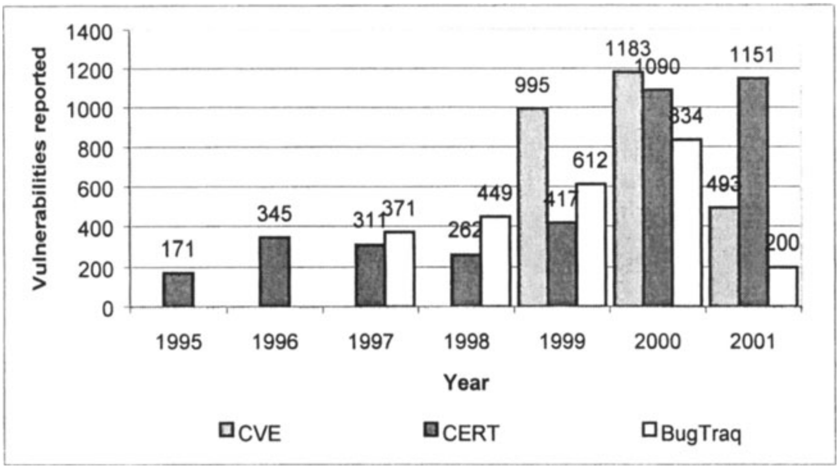

Figure 2 : CVE, CERT and BugTraq total vulnerabilities reported (1995 - mid 2001)

As can be seen from the figure, there is a clear upward trend in the number of vulnerabilities reported by each source. Cross comparison of this against information from other sources, such as the annual CSI/FBI computer crime surveys [2, 14], reveals that there has been a significant increase in the number of security incidents during the same period. This provides possible evidence to support the pattern illustrated in figure 1, where incidents increase after the disclosure of a vulnerability.

It is also interesting to consider how the totals from Figure 2 break down into monthly averages, as this has a more direct relationship to the likely workload of any administrator (or admin team) interested in addressing the problem. Table 1 presents a summary of the above figures in this context, and illustrates clearly that there has been a significant increase in the problem during the last two years. Indeed, looking at the CERT figures reveals that the average number of vulnerabilities reported in the first and second quarters of 2001 represents an almost tenfold increase on the average figure for the whole of 1998, and more than double the total for the previous year.

Table 1 : CVE, CERT and BugTraq average vulnerability reported per month (1995 - mid 2001)

Year		1995	1996	1997	1998	1999	2000	2001*
Average vulnerability	CVE	-	-	-	-	82.9	98.6	82.2
	CERT	14	29	26	22	35	91	191.8
	BugTraq	-	-	31	37.5	51	69.5	66.7

*CVE and CERT figures are for Q1 and Q2, BugTrag figures are for Q1 only

Comparing the vulnerability statistics of CVE, CERT and BugTraq one can notice that there is a difference between them in terms of the total number and the consequent averages. One reason for this is that the editorial boards of the databases do not use the same standard reference in assessing the vulnerabilities that are reported to them, and the procedures when they study any new vulnerability is slightly different. This has knock-on effects in terms of the vulnerabilities they then choose to pass on as reports to the public. A second reason is that each database has different sources of reported vulnerability. These differences make it difficult for administrators to rely on only one database in order to maintain vulnerability awareness, the impacts of which are examined later in the discussion.

3. BENCHMARKING THE IMPACTS

In order to obtain a clearer impression of what the figures mean in practice, it is useful to consider the impacts of addressing vulnerabilities in practical context. To this end, this section presents the results of a benchmarking exercise, in which the total number of patches required within a reference environment were traced over a 12-month period (1 July 2000 to 30 June 2001). The environment in question was an independently managed subnetwork, comprising ten end user systems and incorporating a range of operating systems and applications that it would not be unusual to find together in a modern IT installation. Table 2 shows the full list operating systems and applications considered, alongside the total number of vulnerabilities and patches required within the reference environment.

It was determined that there were a total of 175 distinct patches during the benchmarking period that were applicable to the systems used within the reference environment. This resulted in a monthly average of 40 patches to be applied across the ten end systems.

Table 2 : Total vulnerabilities for software used and requiring patches

Software	Vulnerabilities reported (1/7/00 – 30/6/01)	Clients affected	Total patches required
Windows NT Workstation 4	13 (a)	10	130
Windows 2000	27 (a)	1	27
Office 2000 SR-1	6 (a)	10	60
Office 97	6 (a)	2	12
Windows 98	11 (a)	2	22
Netmeeting	1	3	3
Media player 6.4	2 (a)	10	20
Adobe Acrobat 4.0	1 (b), (c), (d)	9	9
Macromedia Flash 5	1 (b), (c), (d)	1	1
Red Hat 7.1	19(d) (19/4/2001-26/6/2001)	1	19
Senmail 8.11.2	1 (e), (f), (c) (1/3/2001-30/6/2001)	3	3
SuSE 6.4	44 (g)	3	132
SuSE 6.1	37 (g)	1	37
Netscape 4.76	1 (e), (f)	3	3
Linux Kernel 2.2.14	2 (e), (f), (c)	2	4
Linux Kernel 2.4.2	2 (e), (f), (c)	1	2
Linux Kernel 2.2.18	1 (e), (f), (c)	1	1
APACHE 2.0	1 (e), (f), (c)	1	1
Total	**175**		**486**

(a) Microsoft Security Bulletin, http://www.microsoft.com/technet/itsolutions/security/current.asp
(b) BugTraq, http://www.securityfocus.com
(c) http://icat.nist.gov/icat.cfm
(d) http://www.linuxsecurity.com/advisories/index.html
(e) BugTraq database, http://www.securityfocus.com
(f) http://www.cert.org/
(g) http://www.suse.com/us/support/security/index.html

An immediate observation is that such a figure could have significant workload implications for such a small environment, where the administration duties (including security) would typically be handled by a single person (often as only a part-time activity). Furthermore, if the network grew, then the burden of patches would increase correspondingly.

The time taken to apply patches can be extremely variable, and while some may be applied fairly quickly on the fly, others demand more involved activity and may, for example, necessitate the restart of a system. These factors have a further influence on the administrator's workload, as well as potential implications for the availability of the system to other users.

As the footnotes to Table 2 suggest, the information above was collated from a number of different sources, including the three lists discussed in the previous section, as well as product/vendor-specific sources. Ideally, it would be desirable for an administrator to simply be able to rely upon a single source of advisory information, such as one of the generic lists already discussed. However, looking at the number of vulnerabilities reported in each source in relation to the same product reveals another complicating factor, in that each database records a different number of reports. This is illustrated in Table 3, in relation to the Windows NT 4 and SuSE Linux 6.1 operating systems. In the case of Windows NT the product-specific source was Microsoft's Security Bulletins, whereas for SuSE the reference was Linux security advisories, and these are contrasted with the number of reports issued by CVE, CERT and BugTraq in the same period (July 2000 to June 2001).

Table 3 : Comparison of vulnerabilities advisories from product-specific and generic sources

Application	**Vendor bulletins / advisories**	**CVE**	**CERT**	**BugTraq**
Windows NT Workstation 4	13	11 (6 CVE + 5 CAN)	1	2
SuSE 6.1	37	6 (CAN)	2	16

As one might well expect, the vendor/product-specific sources provide the most comprehensive number of reports, but this is only of practical benefit if an organization happens to source all of its operating system and application software from a single vendor. In any other situation, an administrator is still forced to monitor multiple sources to guarantee being fully informed.

4. IMPLICATIONS OF FINDINGS

The findings from this investigation enable some clear conclusions to be drawn regarding the problem of vulnerabilities as currently experienced. Firstly, the statistics collected from the vulnerability databases show that the problem is increasing, largely independently of any specific operating system or application environment. In addition, unlike some other security issues such as viruses, it is not generally the case that old vulnerabilities can be relied upon to disappear in order to make way for new ones. With viruses, the problems posed by older strains are largely eradicated if recent anti-virus software is used that includes the appropriate signatures to enable detection, and older viruses effectively die off in the wild. The same cannot be said of vulnerabilities. Even when an operating system vulnerability may have been known for some time, the original version of that software may still be being shipped – and it continues to rely upon administrators to apply the appropriate patches or service packs after installing it.

Maintaining awareness of the vulnerabilities of relevance is far easier said than done. In order to do this effectively, administrators are currently obliged to monitor multiple sources of information rather than being able to rely solely upon one reliable source. For example, whilst Microsoft Security Bulletins may represent a suitably timely and comprehensive means of monitoring issues relating to products such as Windows and Office, they do not help at all if the organization runs other operating systems or applications from other vendors – so it becomes necessary to monitor and assess these as well, which equates to additional administrative burden. Of course, the majority of vulnerability advisory sources enable interested parties to subscribe to an email list, and thereby receive the advisory messages automatically, as opposed to the administrator having to actively go and search for them. Whilst this certainly helps to reduce part of the burden, it does not alleviate it altogether. For example, the administrator must still read each bulletin or advisory message that arrives in order to determine whether or not it requires action. In many cases it may be determined that this may not be the case – bulletins will very often be received that relate to a software product or version that an organization does not use. However, in order to establish this, irrelevant material must firstly be read, and potentially investigated, which serves only to waste time.

The benchmarking study has shown that the implications go beyond simply monitoring and maintaining awareness of the problems, and that actually addressing them in practice has major workload implications for an administrator - even within the context of a relatively small environment such as that studied. In some cases, the number of systems may run into the thousands, whereas the administration team may number less than ten. Relating this to the number of patches released per month, this could lead to the administrator having to patch more than 100 machines per day.

Of course, the problem of vulnerabilities cannot simply be ignored - this is too risky and the evidence already shows that the problem is not just a theoretical one. The existence of vulnerabilities is well understood in the hacker community, and such weaknesses are frequently exploited in practical assaults upon systems. For example, according to Attrition.org, 99% of the 5823 web site defacements that occurred during 2000 were facilitated as a result of failures to address known vulnerabilities, for which the patches were already available [15].

All of these observations lead to the natural question of what can be done to overcome, or at least reduce, the problems in order to make the situation more manageable. In this respect, two issues can be considered, both relating to the automation of what are currently manual processes. The first issue concerns reducing the problem of information overload for administrators, and requires a means by which administrators can be notified of new vulnerabilities in a manner that flags only the advisories that are likely to be of relevance to them. Such filtration and prioritisation of available advisories would enable administrators to direct their efforts more effectively, reducing the amount of time lost following up irrelevant material and enabling genuine problems to be addressed more quickly. This, however, does not overcome the problem that, having obtained the relevant information, they then have to act upon it. Finding the time to do this can be an equally, if not more, demanding task, and a further valuable element of desirable automation would therefore be an active vulnerability resolver, capable of acting upon the notification data above in place of the administrator. This is by no means a trivial undertaking and it is recognized that allowing autonomous control over the update of a system could itself represent a security vulnerability and, as such, the mechanisms must be designed so as to rectify weaknesses in a stable and trusted manner, as well as to guard against compromise of the approach (e.g. by malicious parties).

The proposed solutions are not trivial to achieve. Even with the information gathering aspect, there are challenges to address, including ensuring that any automated agent does not inadvertently filter out and discard relevant information. The automated rectification is more problematic, in that it would need to guard against causing inconvenience or indirect denial of service to legitimate users (e.g. any rectification agent must not, for example, take the system down to install a patch whilst users are working). Furthermore, the framework would need to ensure the validity of the patches and corrections that it tries to apply – using, for example, digital certificates to verify the legitimacy of the source. Without this, the rectification agent could be misused by hackers as a means of getting the target system to accept malicious code.

These proposals contrast significantly with current marketplace solutions for vulnerability resolution. Although some current scanners do include auto-update features that enable them to be aware of and detect the latest vulnerabilities [16], these require specific actions on the part of the product vendors, who must release the associated update for their product. Where products also include 'fix-it' technology, allowing administrators to rectify some issues automatically, these often relate merely to configuration details and do not take care of significant software upgrades or patches [17].

5. CONCLUSIONS

This paper has presented evidence to show that the problem of IT security vulnerabilities is significant and growing. The scale of the problem is such that maintaining awareness of new vulnerabilities can be a major undertaking for system administrators, requiring them to monitor information from multiple sources, and filter out the details of relevance, before being in a position to take any remedial action. The issue of eradicating any relevant vulnerabilities then represents a further element of workload that must be accommodated, typically alongside other duties related to the day-to-day operation and maintenance of a system. These other factors may often lead to security administration being postponed or batched until such time as more pressing demands subside. Although this is perfectly understandable from a practical workload perspective, any delay in addressing publicised vulnerabilities can lead to an increased window of opportunity for hackers. These factors highlight the increasing need for automated solutions, and the design of an automated vulnerability advisor-resolver architecture is the focus of ongoing research by the authors.

6. REFERENCES

[1] SANS Institute. 2001. "How To Eliminate The Ten Most Critical Internet Security Threats", Version 1.32, January 18 2001. http://www.sans.org/topten.htm

[2] Computer Security Institute. 2000. "2000 CSI/FBI Computer Crime and Security Survey", *Computer Security Issues & Trends*, vol. VI, no. 1. Spring 2000.

[3] Microsoft Corporation. 2000. "The Definition of a Security Vulnerability", http://www.microsoft.com/technet/secrity/vulnrbl.asp, December 2000.

[4] Longstaff, T.A, Ellis, J.T., Hernan, S.V., Lipson, H.F., McMillan, R.D., Pesante, L.H. and Simmel, D. 1997. "Security of the Internet", The Froehlich/Kent Encyclopedia of Telecommunications, vol. 15. Marcel Dekker, New York: pp.231-255.

[5] Arbaugh, W.A., Fithen, W.L., and McHugh, J. 2000. "Windows of Vulnerability: A Case Study Analysis", *IEEE Computer*, vol. 33, no. 12, pp52 - 59.

[6] Noack, D. 2000. "The Back Door Into Cyber-Terrorism", APBnews.com, 2 June 2000.

[7] Furnell, S.M., Chiliarchaki, P. and Dowland, P.S. 2001. "Security analysers: Administrator Assistants or Hacker Helpers?", *Information Management and Computer Security*, vol. 9, no.2: 93-101.

[8] CERT/CC. 2001. "The CERT Coordination Center FAQ", CERT Coordination Center, http://www.cert.org/faq/cert_faq.html, May 2001.

[9] CVE. 2000. "Introduction to CVE, The Key to Information Sharing", MITRE Corporation. http://cve.mitre.org/docs/docs2000/key_to_info_shar.pdf

[10] CVE. 2001. "CVE (version 20010507)". Mitre Corporation. http://cve.mitre.org/cve/downloads/full-cve.html

[11] Hafner, K and Markoff, J. 1991. *Cyberpunk: Outlaws and Hackers on the Computer Frontier*. Fourth Estate Limited.

[12] CERT/CC. 2001. "CERT/CC Statistics 1988-2001", CERT Coordination Center, http://www.cert.org/stats/cert_stats.html, Jun 2001.

[13] Security Focus. 2001. "BUGTRAQ Vulnerability Database Statistics", http://www.securityfocus.com/vdb/stats.html, Jun 2001.

[14] CSI. 2001. '2001 CSI/FBI Computer Crime and Security Survey', *Computer Security Issues & Trends*, vol. VII, no. 1. Computer Security Institute. Spring 2001.

[15] CNET. 2001. "Patchwork Security - Software "fixes" routinely available but often ignored", CNET News.com report. 24 January 2001. http://news.cnet.com/news/0-1007-201-4578373-0.html

[16] eEye-Digital Security. 2001. "Retina: The Network Security Scanner", http://www.eeye.com/html/assets/pdf/retina_whitepaper.pdf

[17] Forristal, J. and Shipley, G. 2001. "Vulnerability Assessment Scanners: Detection Result", Network Computing, http://www.networkcomputing.com/1201/1201f1b1.html