Critical awareness – The problem of monitoring security vulnerabilities

Steven Furnell, Abdulaziz Alayed, Ian Barlow and Paul Dowland

Network Research Group, University of Plymouth, Plymouth, United Kingdom
nrg@plymouth.ac.uk

Abstract: Security vulnerabilities are known problems that frequently affect operating systems, Internet servers and application programs from numerous vendors. The paper examines the scale of the problem, referencing advisory sources such as CERT/CC, BugTraq and CVE. Although it is relatively easy to obtain advisories, administrators can be overwhelmed by the volume of information – not all of which is relevant. The paper proposes a generic vulnerability report format, which aims to provide a basis for administrators to filter and prioritise incoming information to suit their needs.

Keywords: Vulnerabilities, Exploits, Advisories, Administration.

1. Introduction

Exploitable vulnerabilities in operating systems and applications represent a significant threat to the security of many Internet systems. Recent years have witnessed a variety of security breaches (including hacker attacks, such as denial of service and defacement of web sites, and malware incidents, such as viruses and worms) that have been facilitated by the exploitation of weaknesses in either the system's software or its configuration (SANS Institute 2001). This has prompted an increased awareness of the issue on the part of software vendors, who have become more vocal in stating their commitment to producing secure, reliable software. For example, in the last 12 months, major companies such as Microsoft and Oracle have made specific reference to the issues in their public relations and marketing materials:

- In early 2002, Bill Gates sent an email to all Microsoft employees headed 'Trustworthy Computing", in which he set out his vision of the importance of security and reliability in Microsoft products (Gates 2002).

- Oracle launched a major advertising campaign based around the claim that its database product was 'unbreakable'. The company's web site and advertising for Oracle9i began to include the bold statement "Unbreakable security. Can't break it. Can't break in".

Nonetheless, at the time of writing, the problem of vulnerabilities still persists. For example, Oracle's claims prompted a response from the analyst organisation Giga Information Group, which pointed out that three major flaws had been uncovered in Oracle products since the launch of the 'unbreakable' campaign (CW360 2002) It also pointed out that Oracle's bullish attitude was likely to increase the chances of it being a target for hackers, who would simply view the 'unbreakable' claim as a challenge. As a result, addressing vulnerabilities represents an ongoing task for system administrators, who are effectively engaged in an continuing battle to secure their systems and networks before they fall victim to an attack.

The paper begins by considering how administrators can obtain relevant information about vulnerabilities that may affect their systems, identifying a number of organisations that maintain product and vendor-independent advisory repositories. The discussion then

proceeds to consider the magnitude of the problem, and identifies reasons why this may lead to administrators being overwhelmed by the volume of information they are presented with. The requirement to ease the burden on administrators leads to the proposal of a means for enabling increased automation of the vulnerability notification process.

2. Obtaining relevant vulnerability information

If system administrators wish to maintain awareness of vulnerabilities affecting their systems, then it is necessary for them to have appropriate sources of information. A number of publicly accessible sources are available that maintain repositories of the associated warnings and advisory reports, which can be categorised according to whether they are provided by a specific vendor (e.g. Microsoft or Sun), or a vendor-independent group. Three examples of vulnerability advisory sources falling into the latter category are CVE, CERT/CC, and BugTraq, the activities of which are summarised below:

- **Common Vulnerabilities and Exposures (CVE)**
  CVE is a list of information security vulnerabilities that aims to provide common names for publicly known problems (CVE 2000). 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 (CVE 2001).

- **Computer Emergency Response Team / Coordination Center (CERT/CC)**
  The CERT/CC is a major reporting centre for Internet security problems, which analyses product vulnerabilities and maintains a searchable database of problems (CERT 2001). 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.

- **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” (Security Focus 2001). Since its original inception in 1993, the list has grown to encompass over 27,000 subscribers, and includes information relating to vulnerabilities, exploits and associated fixes for a wide variety of operating systems and application programs. As with CERT, 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.

In addition to these independent sources, there are also vendor-specific sources, which provide information tailored to a specific product or range. A good example is that of Microsoft, which maintains a relevant section on its website, including a series of advisory reports entitled Microsoft Security Bulletins, addressing the company's full range of operating system, server and application programs. Vendor sources will often contain a greater volume of information, as well as downloadable patches that can be installed to rectify problems that have been already been solved.

3. Assessing the scale of the problem

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 1 presents statistics relating to the total number of vulnerabilities reported each year, in the period from 1995 to 2001 (note: the BugTraq figures for 2001 only cover the period up to August). 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.

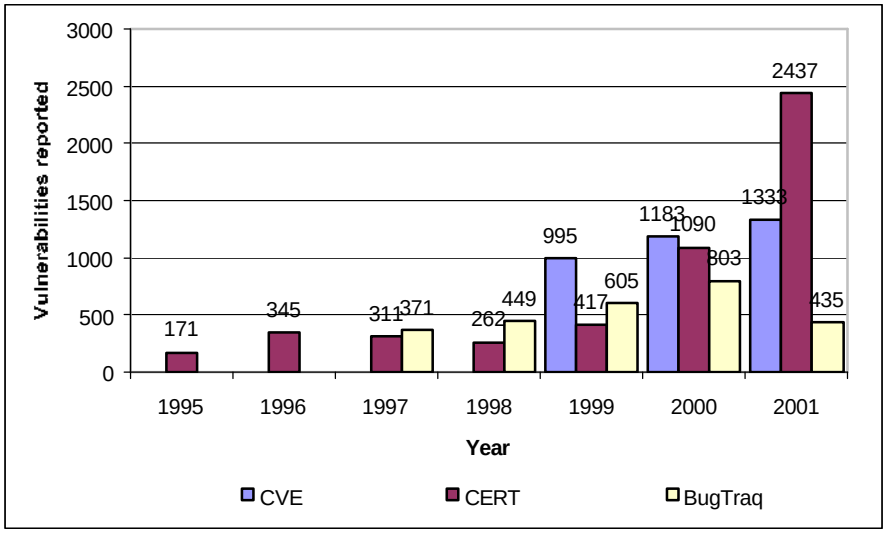

Figure 1 : CVE, CERT and BugTraq total vulnerabilities reported (1995 - 2001)

Figure 1 shows a clear upward trend in the number of vulnerabilities reported by each source. A more detailed investigation of the underlying reports also reveals that the increase in the vulnerability problem is occurring largely independently of any specific operating system or application environment. Previously published work by the authors has used this information as the basis for benchmarking the task that would consequently face system administrators over a one year period if they wished to track relevant vulnerabilities and apply associated fixes. The findings suggested that even for a small network (of ten workstations), using a

relatively standard selection of Microsoft software and an implementation of Linux, the task for administrators could involve installing an average of 40 patches per month across the affected systems (Alayed et al 2002).

Such a volume of advisories requiring attention gives a clear indication of the scale of the task facing system administrators, and indeed a current problem is the sheer number of sources that they may need to consult in order to ensure that they receive relevant information about all vulnerabilities that may pertain to their systems. 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 - each database records a different number of reports. This is illustrated in Table 1, 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 (which, in this part of the study, was from July 2000 to June 2001).

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

Table 1 : Comparison of vulnerabilities advisories from product-specific and generic sources

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 organisation happens to source all of its operating system and application software from a single vendor. In any other situation, an administrator may end up needing to monitor, or subscribe to, multiple sources, each of which may provide a different level of information. Even if the organisation does only use software from one vendor, monitoring alternative sources might still be relevant, because generic security sites may issue an alert before the vendors formally acknowledge a vulnerability or release a patch.

In an attempt to make things a little easier, most sources now make it easy for administrators to obtain the advisory information, and enable them to subscribe to mailing lists rather than manually monitor the information from a website. However, although this is clearly helpful to some extent, it can also lead to administrators receiving large amounts of information unrelated to the systems that they run. For example, subscribing to Microsoft Security Bulletins would not only yield messages relating to Windows NT Workstation 4, but also any other products from Microsoft's portfolio (some of which the recipient organisation might also run, but also many that it would be likely not to). The unfortunate consequence of this is that administrators may quickly become overwhelmed by the volume of incoming information that they need to consider. For example, the administrator must still read each bulletin or advisory message that arrives in order to determine whether or not it requires action. However, in order to establish this, potentially irrelevant material must firstly be read, and potentially investigated, which ultimately serves only to waste time. This overhead may in turn lead to administrators postponing consideration of the advisories until they have time

in their schedule to examine many of them in a batch (which may result in published vulnerabilities remaining unaddressed for a much longer period, leaving a greater window of opportunity for exploitation). Worse still, they may become complacent about the situation – particularly if the majority of advisories that they have to work through turn out not to be relevant to their systems. Of course, the problem of vulnerabilities cannot simply be ignored. The existence of exploitable weaknesses is well understood in the hacker community, and they are frequently utilised 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 (CNET 2001).

4. A generic format for vulnerability advisory reports

In view of the above, a means is required to enable administrators to be more selective in terms of the information that they receive. Ideally, notification should occur in a manner that flags only the advisories that are likely to be of relevance to the software and network configuration in the target organisation, and gives an indication of their relative importance. 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.

It is suggested that the above could be achieved via an automated software agent, which allows administrators to indicate the systems that they run (as well as other characteristics that could be used to more specifically define the information that they are interested in receiving) and then filters and prioritises incoming advisory reports accordingly. Unfortunately, the ability to perform such automatic filtering is currently complicated by the fact that each vulnerability reporting source releases its information in a different format, and they do not provide a consistent set of details (so, the same vulnerability would be described in different ways by different reporters).

The authors have performed a top level analysis of vulnerability reports from the different sources identified earlier, plus a number of vendor-specific sources, and have identified the core elements that a meaningful advisory needs to include. The result is a generic data set for vulnerability advisory reports, which has been abstracted and enhanced from existing advisory formats, as listed and described in Table 2. Although most of this information is commonly found in existing reports, it is often buried within free-text descriptions rather than being represented in distinct fields. Clearly, abstracting the information out into separate fields increases the potential for automated search and manipulation of the resulting information. It should be noted that in the case of the ‘vulnerability type’ and ‘target’ fields, the defined values have been adopted from those used by the ICAT Metabase, a search engine for CVE-listed vulnerabilities (ICAT, 2002).

Main field	Sub-field	Defined values (if applicable)	Description / Comments
Title			Title of the vulnerability or exploit.
Advisory ID			A reference number for the vulnerability, assigned by vendor.
Vendor ID			Denotes the vendor of the affected product.
Affected Product	Package		The affected product / packages in relation to this vulnerability.
	Version		If all versions of the product are vulnerable, then this field could be left blank.
Date	Released		Date of original issue of the advisory.
	Revised		Revision history of the advisory (if applicable).
Severity/risk		Critical/High/Medium/Low	Indicates the level of risk to systems on which the vulnerability could be exploited.
Exploitation side		Local/Remote	Indicates whether the vulnerability can be exploited locally, remotely or from both locations.
Exploitation ratings	Internet	Low/Medium/High	Potential for exploitation via the Internet.
	Intranet	Low/Medium/High	Potential for exploitation if the attacker has access to the local intranet.
	Client	Low/Medium/High	Potential for exploitation if the attacker has access to the local client system on which the vulnerable software is installed.
Vulnerability type		Input error (buffer overflow). Access error. Exceptional condition error. Configuration error. Design error.	Indicates the type of flaw / weakness that is being exploited when targeting the vulnerability.
Target type		Operating system. Network protocol stack. User application. Server application. Hardware. Communication protocol. Other component.	Indicates the type and level of software that contains the vulnerability and hence becomes the target of the exploit.
Impact / consequence	Availability	Low/Medium/High	The impact of exploitation of this vulnerability on the system and its potential to cause issues such as Denial of Service, Information Disclosure, and exposure of the system to hostile code.
	Confidentiality	Low/Medium/High	
	Integrity	Low/Medium/High	
Automated exploit		Yes/No/Unknown	Indicates whether the exploitation can be automated in software, or requires manual intervention by the attacker
Expertise to exploit		Low/Medium/High	Indicates the level of technical expertise that an attacker would require in order to successfully exploit the vulnerability.

Main field	Sub-field	Defined values (if applicable)	Description / Comments
Solution	Work-around available	Yes/No	Information relating to any solutions to avoid this vulnerability (if available). Details of any workaround or patch would then be documented in the 'problem description' field.
	Patch available	Yes/No	
Problem description			A free-text description of the vulnerability, which could present specific technical details and other supplementary information.
References			Links to information about the same vulnerability, which may be provided in other sources.
Cross references			Links to related reports about other vulnerabilities.
Obsoletes			Details of any advisories that the current one supersedes or renders obsolete.

Table 2 : Draft Generic Vulnerability Advisory Format

It is suggested that this could be used as the basis for a common vulnerability reporting format, which could be adopted by multiple reporting sources so as to make their information compatible. If this were to be achieved, the aforementioned automated agent would have a consistent basis from which to work, and would thus be able to allow system administrators to filter the information to suit their needs. For example:

- The administrator could specify the products utilised within the organisation, and the agent would then selectively forward the relevant reports only.

- Fields such as 'severity' and 'impact' could be utilised to help prioritise how urgently the vulnerability report needs to be acted upon.

- Fields such as 'expertise to exploit' and 'automated exploit' could be used to indicate how likely an exploit is to occur, which could again feed into the prioritisation process.

In this way administrators could reduce the volume of information to which they are exposed (i.e. they should receive only reports that pertain to their systems, rather than receiving everything the source has to offer), as well as having a more structured approach regarding what issues to address first.

5. Conclusions

The paper has established the significant problem that system administrators may face in maintaining an appropriate awareness of the security vulnerabilities affecting their systems. The generic vulnerability advisory format that has been proposed will represent a valuable step forward in facilitating automated filtering and prioritisation of incoming reports – thus reducing the potential for information overload and wasted time for administrators. The authors' research will proceed to trial the generic advisory format, including the development

of the accompanying software to allow administrators to express their interests and filter incoming advisory messages accordingly.

Another vital point to note is, of course, that having found the information about a vulnerability that affects your system, it is necessary to do something about it. Merely being aware of a weakness will not stop someone else from being able to exploit it (indeed, many incidents occur as a result of known issues, in which the victim did not act quickly enough to protect their system). Patching the vulnerabilities is another aspect that has significant workload implications for the administrator, and as with keeping up to date with advisories, it often risks being sidelined in favour of what may appear to be more pressing administration duties (e.g. responding to incessant requests from the user community). As such, research is also required to enable increased automation of vulnerability rectification, and this will represent another aspect of the authors' ongoing study.

6. References

Alayed, A., Furnell, S.M. and Barlow, I.M. (2002) "Addressing Internet security vulnerabilities: A benchmarking study", in *Security in the Information Society: Visions and Perspectives*, M. Adeeb Ghonaimy, Mahmoud T. El-Hadidi and Heba K.Aslan (Eds), Kluwer Academic Publishers, Boston. pp121-132.

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

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

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

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

CW360 (2002) "Giga slams Oracle security claims", CW360.com, 23 January 2002.

Gates, B (2002) "Trustworthy Computing", email message to Microsoft employees, 15 January 2002.

ICAT (2002) "ICAT Metabase Documentation", http://icat.nist.gov/icat_documentation.htm.

SANS Institute (2001) "How To Eliminate The Ten Most Critical Internet Security Threats: The Experts' Consensus", Version 1.32, January 18 2001. http://www.sans.org/topten.htm.

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