Security analysers: Administrator Assistants or Hacker Helpers?

ABSTRACT

Security analyser tools provide a means of automatically identifying, and potentially exploiting, vulnerabilities within computer systems and networks. Although such tools are useful to system administrators, in order to highlight and overcome weaknesses in protection, they are also of assistance to hackers looking for ways to break in. The paper highlights the range of tools that are currently available (and of potential use to both audiences) and considers the extent to which each group is likely to benefit from them in practice. It is considered that the ease of use of tools such as Back Orifice 2000 provide a means by which even the relatively unskilled hacker may inflict damage upon a system. Although it can be argued that the tools are generally equally available to hackers and administrators, the hacker community is likely to be more aware of the opportunities available. Even where they are aware of the existence of particular tools, survey results presented in the paper indicate that system administrators make relatively limited use of them. Factors that may account for this include their overall workload and lack of security awareness. Appropriate countermeasures can be identified to combat the individual categories of tool, but the problem of ensuring that these safeguards are implemented still remains.

KEYWORDS

Security analysers, Hackers, Administrators, Vulnerabilities.

INTRODUCTION

One of the frequent benefits of information technology is that it can help to make complex tasks easier to perform by automating certain elements. It is possible to identify numerous and wide-ranging examples of where this is the case, from manufacturing to warfare to the office environment. A classic example in the latter case is the use of a spreadsheet, which has a fundamental effect on the ease and speed of making calculations. In all of these scenarios, the automation is generally seen to be beneficial as it has a positive effect upon aspects such as productivity and reliability. Unfortunately, however, it is also possible to identify scenarios in which the automating properties of technology can be used to undermine the technology itself. An example here can be cited in terms of the identification and exploitation of computer security vulnerabilities. Whilst this was once the sole province of individuals with the appropriate technical skills, the ability to assess the protection of a system and, potentially, take advantage of any weaknesses identified, it is now frequently encapsulated within software tools that are publicly available on the Internet. Although the motivation of these tools is often to provide system administrators with an automated means of checking their systems, the public availability of the tools makes them an attractive facility for hackers, with less

benevolent intentions. Furthermore, a number of tools and pre-written attacks (often referred to as exploit programs) have been released by the hacker community itself, providing not only the means to automatically identify a vulnerability, but also to take advantage of it, to the detriment of the target system.

This paper considers the threat posed by such automated analysis and attack programs. It begins by summarising the evolution of such tools and then presents an overview of the different categories that can now be identified. The potential threat is then analysed by considering the use of the tools by the hacker and system administrator communities. The latter aspect is supported by the results of a questionnaire study that attempted to assess administrators awareness and use of the available software. The discussion concludes by considering the different approaches that may be used to control the problem.

THE EVOLUTION OF ANALYSERS AND EXPLOIT PROGRAMS

The concept of automated attacks can be traced back to programs such as password crackers and war diallers (see Table I for overview descriptions), which have been around in some form since the early days of personal computing. The difference then was that the distribution of such programs (if indeed they were distributed at all) was quite limited. If a hacker had the need for such a program, then he would probably have the skill to write one for himself – an upfront investment of skill and effort in order to reduce the level of mundane activity required later. It can also be noted that these tools were used to locate and assist in gaining entry to systems, rather than automating the exploitation of some vulnerability. Having gained access to a system, it would be down to the hacker's own creative talents to determine what happened next. This is in no way meant to applaud the hacker's subsequent actions or to suggest that they were any more legitimate because the hacker performed them without further programmatic assistance. The point is that the overall potential for damage was less because fewer people has the required knowledge and skills at their disposal to discover and exploit obscure vulnerabilities in operating system software and the like. With such knowledge now encapsulated in all manner of scripts and exploit programs, this is no longer the case.

Password crackers and war diallers take advantage of inherent characteristics of the implemented systems (i.e. that an alphanumeric string can be broken by brute force and that it is possible to distinguish between voice and data traffic) rather than exploiting unforeseen vulnerabilities. Some of the programs available today are in marked contrast to this, having been established specifically to identify and/or exploit bugs in software or potential holes in a security configuration for malicious intent. These newer tools provide the ideal platform for opportunity hackers – those who do not have the skill to break into a system themselves and may have no particular target in mind, but will happily attack a system if the vulnerability is there and the means is provided.

One of the first programs of the new breed to attract public attention was the Security Administrator's Tool for Analyzing Networks (SATAN), written by Dan Farmer and Wietse Venema and released in 1995. SATAN is a network-based vulnerability

scanner that has the capability to identify a range of potential security weaknesses, including:

- Password file access from arbitrary hosts
- Remote shell access from arbitrary hosts
- Writable anonymous FTP home directory
- NFS file systems exported to arbitrary hosts or unprivileged programs

The argument behind the public release of the tool, which is common to several others, is that the problems it can identify are not a secret. They have been documented in various sources, including advisories from organisations such as CERT (Computer Emergency Response Team) and CIAC (Computer Incident Advisory Capability), as well as in security handbooks. As such, the information necessary to exploit them is easily available for those with an interest in doing so. The question is whether putting it all together in an automated system is simply making it too easy (whereas obtaining the required information from documented sources and then manually using it to target systems at least required some effort and understanding). Of course, the authors of SATAN were not ignorant of its potential for misuse, as illustrated by the following quote from the Frequently Asked Questions web page (Farmer and Venema, 1995) :

> *"We realize that SATAN is a two-edged sword - like many tools, it can be used for good and for evil purposes. We also realize that intruders (including wannabees) have much more capable (read intrusive) tools than offered with SATAN. We have those tools, too, but giving them away to the world at large is not the goal of the SATAN project".*

It should also be noted that, whilst SATAN detects and reports vulnerabilities, it does not actually exploit them. Furthermore, it offers tutorial explanations of the problems and what can be done to rectify them – reinforcing the point that it is offered from the perspective of assisting the administrator in strengthening the system rather than for use by a hacker to disrupt it. Nonetheless, SATAN's release was met with mixed opinion from the IT and security industry, with many expressing the view that it was inviting trouble (Bicknell, 1995). The choice of name was, of course, hardly helpful in promoting a safe image either.

SATAN falls into the category of vulnerability scanner and there are now various other tools of a similar nature. In addition, there are a range of other categories of tool that may also be of use to both system administrator and hacker communities. These are considered in the next section.

AN OVERVIEW OF AVAILABLE TOOLS

Table I presents a summary of the main categories of security analysis tools, indicating some of the better known examples of available programs in each case.

Category	Description	Examples
Vulnerability scanners	A program that can probe a network and identify some/all systems connected to it. Once identified each system is investigated to assess its susceptibility to attack. Typical vulnerabilities may include known weaknesses or bugs within operating systems, Internet servers or application software.	COPS Titan SATAN SAINT SARA
Remote Administration	A program that allows a user to remotely monitor and control a target system. If used by a hacker, the program may be installed via stealth methods (e.g. as a Trojan Horse). The remote user may be able to capture passwords, download/alter/delete files, access emails and even corrupt system files.	Back Orifice 2000 NetBus BackDoor-G SUB7
War Dialers	A program that dials a list of telephone numbers either in sequence or randomly. Once a modem carrier tone is detected the phone number is logged for further investigation.	Toneloc PhoneSweep
Port scanners	A program that probes specific systems (or a range of network addresses) and identifies available TCP/UDP ports. Programs can also search for specific service ports, such as Telnet, FTP, SMTP etc., which may have known weaknesses. Once a port is identified it may be possible to determine the target operating system, which may reveal further vulnerabilities	Nmap Strobe Ncat
Sniffers	A program that sets a network card into promiscuous mode in order to enable the capture of all network traffic. This can be used to directly grab plain-text passwords (e.g. from FTP, SMTP, POP3, Telnet applications) or to gather encrypted versions (e.g. from Windows NT SMB packets, SSL, etc.). Sensitive data may also be captured from other forms of network traffic, such as email	Analyser Ethereal Supersniffer WinSniff Iris

	messages.	
Password Crackers	A program that attempts to break an encrypted password string. Passwords are gathered in an encrypted form, either from a copied password file or by eavesdropping on the network. Once captured, the program can compare the encrypted string against a dictionary of pre-encrypted words to find a quick match. Failing this, a brute force attack attempting all combinations can be launched.	Crack John The Ripper L0phtCrack

Table I : Categories of security analyser tools

There are, of course, some other categories of exploit program that have no legitimate security analysis purpose whatsoever, for example denial of service tools. These programs have clearly been created with a malicious intent and it is not possible to offer even the shaky defence of legitimacy that can be given for other hacker-originated tools. Their only use to system administrators is in running a self-test to ensure that a system is not vulnerable to such attacks.

With some of the other tools, the problem is not so much that they exist, but that they are so freely available. Why should the average Joe, with no system to administer, be permitted to have the same access to a vulnerability scanner as someone responsible for IT assets worth thousands of pounds? In the same sense as confidential or sensitive information, access should be based upon a need to know. Making scanners freely accessible is analogous to saying that anyone should have the freedom to go around and try the doors and windows of your house, and be free to enter if you have not left them properly secured.

Having established the range of tools available, the discussion will now proceed to consider their relative merits in practice. Venter and Eloff (2000) present a general assessment of the applicability of different categories of tool to different audiences (i.e. administrators, hackers and end users). The following discussion seeks to provide a deeper view by assessing the extent to which hackers and administrator audiences can make use of the tools (the end user audience is not considered in this study, as tools are generally not released with this specific audience in mind).

AN OPEN INVITATION TO HACKERS?

It can be argued that if a technically competent hacker wishes to get into your system, then he/she will be able to do so regardless of assistance from analyser and exploit programs. One of the main concerns, therefore, is how greatly these tools open up the playfield for relatively unskilled novices. Whilst some tools still require a fair level of technical competence to install, run and interpret their output (enough at least to scare away the casual newbie looking for a quick result), others provide a truly automated, point and click approach to assessing and exploiting vulnerabilities. This

can be illustrated by considering the use of the Back Orifice 2000 (BO2K) tool, which provides a means for remote administration of a target system (Back Orifice is chosen here as it has been around for some time and, therefore, to show the process of attack is revealing nothing new. In addition, most security conscious organisations should already be protected against it via standard anti-virus software).

Back Orifice 2000 is a tool consisting of two main elements, a client application and a server application. The client, running on one machine, can be used to monitor and control a second machine running the server. The use of BO2K, therefore, requires that the server program be installed onto each target machine. This could be explicitly installed by an administrator wishing to conduct remote administration duties, but it will more typically arrive as a Trojan Horse, attached to an email message or similar, and rely on installation by an unwary end user. Then, anyone with the other half of the BO2K software (the administrator tool) can control the victim's PC from anywhere on the Internet. The remote user can stealthily do anything to the victim's machine that the victim could do locally. Some of the operations that can be performed remotely include:

- execute any application on the target machine;
- log keystrokes from the target machine;
- restart the target machine;
- lock up the target machine;
- view the contents of any file on the target machine;
- transfer files to and from the target machine;
- display the screen saver password of the current user of the target machine.

Whilst all of the above features could conceivably be of use to a system administrator wishing to remotely monitor and control a machine within his/her network, it can also be seen that the facilities would represent a significant security risk if placed in the wrong hands. The clear problem in the case of BO2K is that it can be distributed using stealth methods by someone other than the legitimate administrator.

There is a detailed tutorial in the site of BO2K, which can guide the user to install step by step the application and configure both the client and the server. Furthermore there is a wizard that simplifies the procedure of the installation to the minimum. As such, the installation of the program is not beyond the capabilities of a novice user. The simplicity is further illustrated by the user interface, which allows commands to be issued to the server via mouse and menu interactions, as shown in Figure 1.

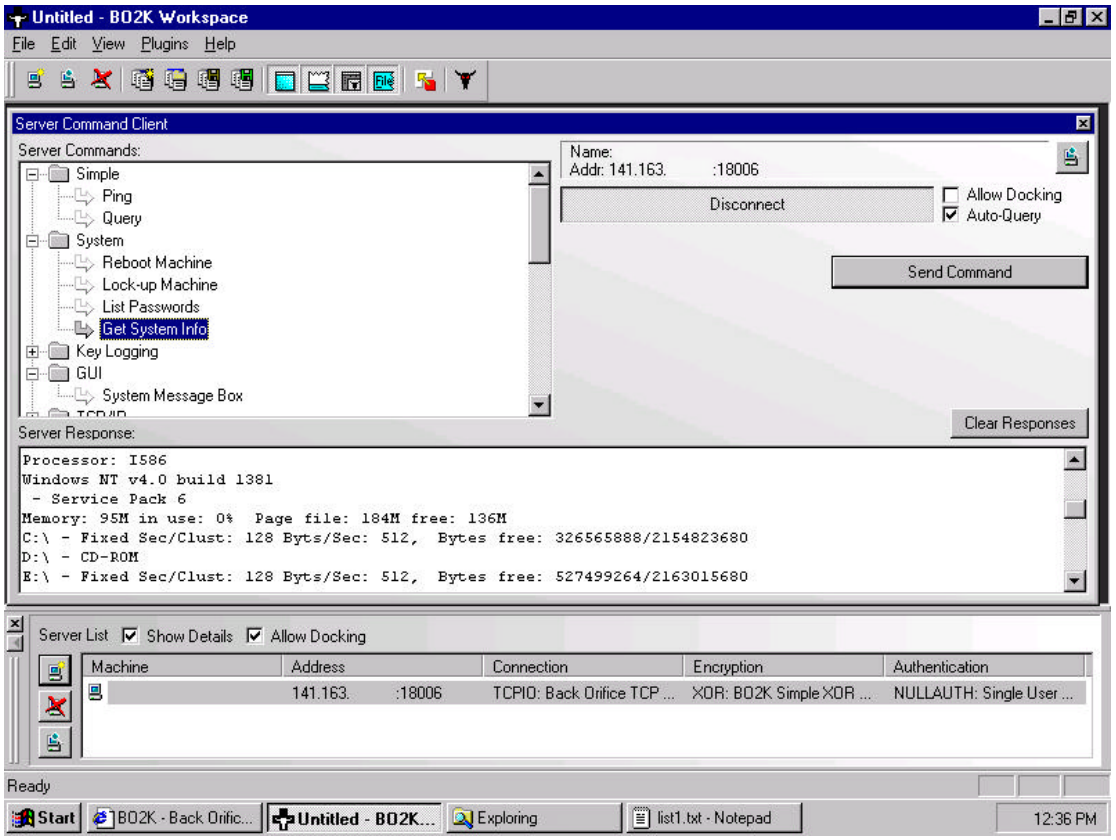

Figure 1 : The Back Orifice 2000 administration tool

Returning to a more general level of discussion, it can be conjectured that a significant barrier to the illicit use of such tools is not only the moral standpoint but also the fear of getting caught. If presented with the necessary means to mount an attack, as well as a 100% guaranteed assurance that they would not be found out, it is likely that many more users would consider taking the opportunity to indulge in some form of mischief. However, in the past, in order to scan or probe a remote machine for vulnerabilities, it was necessary to do so from your own system – introducing the possibility that the attempt would be traced if the target site was vigilant regarding its security. Whilst accomplished hackers could employ means to cover their tracks and hide their location, novices would not have this opportunity. Recently, however, the landscape has changed and there are now web-based tools available that enable the security of remote sites to be scanned via an intermediate web server. The consequence of this is that, from the perspective of the target system, the scan is coming from the web server system and cannot be so easily traced beyond that. Therefore, the attacker is able to remain anonymous, shielded behind the web server that performs the scan on his/her behalf.

An example of such a web-based vulnerability scanner was reported to the Windows NTBugtraq mailing list on 14 August 2000 (Docekal, 2000). The posting was made by Daniel Docekal, the editor of Czech IT newspaper Svet Namodro and concerned vulnerabilities identified in web sites running on Microsoft's Internet Information Server and using Active Server Page (ASP) technologies. The message explained that, as a result of the bugs, people could potentially gain access to password information, source code of scripts, and database files from the affected servers. The possession of such information could then open the way for a more significant breach of security. In order to enable the easy identification of the vulnerabilities, an

automated tool was made publicly available on Svet Namodro's website. Whilst the intention was to allow concerned administrators to test the vulnerability of their servers, the completely open availability of the tool was effectively a security risk in itself, enabling anyone to go to the Svet Namodro site and enter the URL of a server that they were interested in scanning. Whilst the page cautioned users only to use the tool against their own servers and applications, this clearly would not prevent them from doing otherwise. The site also claimed to log IP addresses to enable any misuse to be traced. However, this issue may be clouded where people connect to the server via an ISP, in which IP addresses are normally assigned dynamically for each session. In addition, it assumes that either the target site or the Svet Namodro server can determine that misuse is actually occurring. Reports suggested that Docekal was appalled that thousands of sites were still vulnerable several days after the test had been made available (Delio, 2000). However, this may be a somewhat naive assessment, as it tends to assume that the administrators of the aforementioned sites are all subscribers to the NTBugtraq list or a similar information source that may also have picked up the story. In addition, as later discussion highlights, there are a variety of reasons why system administrators may not be able to respond to security issues as quickly as one may hope or expect.

Other web sites offer what could be considered a more secure scanning service, whereby you can log in and get the server to scan the security of the client machine that your connection originates from. This is still useful from the perspective of an administrator or a security-conscious end-user, in that it allows their own system security to be assessed, whilst at the same time preventing an assessment of any third party's security from being made (thereby removing the potential to assist in attacking someone else). Examples of such services are:

- Shields Up! From Gibson Research Corporation (see grc.com), which enables the Internet connection security of Windows-based systems to be analysed.

- HackerWhacker (see hackerwhacker.com), which enables the scanning of TCP and UDP ports, SMTP email server vulnerabilities and web server CGI weaknesses.

Both services output a report, indicating any potential weaknesses that are identified. Such tools are advantageous in determining the baseline vulnerability of a system, as this information can be gathered without having to breach the security of the target system or spy on it for a prolonged period. The information collected is effectively available to anyone who is able to determine the target IP address and so can provide administrators with an insight into what hackers will also be able to see. The downside is that, with the analysis locked to the machine from which the user is accessing the web site, the administrator would be forced to move from machine to machine in order to check multiple clients on a network. The wider issue of administrator use of security analyser tools is discussed in the next section.

ANALYSERS AS TOOLS FOR SYSTEM ADMINISTRATORS

In order to investigate the extent to which system administrators are aware of and utilise analyser tools, a questionnaire was devised and distributed to 50 IT managers

(the names of the organisations contacted were obtained from databases of the top 100 companies according to a Financial Times survey). Although the responses were anonymous, the response rate was expected to be very low due to the sensitive nature of the subject matter. The survey yielded a total of 12 usable responses (24% of those contacted), whilst several further companies replied that they were 'unable to divulge any information on this subject for security reasons'.

Table II summarises the main sources from which administrators claim to maintain their awareness of security vulnerabilities.

Source	Awareness
Web pages	66%
Microsoft Security Alerts	50%
Bugtraq or other mailing lists	58%
Newsgroups	66%
IRC channels	58%
Black Hat meetings	16%
Phrack, 2600 etc.	41%

Table II : Administrators awareness of security information sources

The respondents were also asked to indicate their awareness and use of different categories of analyser tool, with specific examples being indicated that can be freely obtained from the Internet/WWW. Table III summarises the results in relation to this issue.

Category	Tool	Awareness	Use
Vulnerability scanners	COPS	0%	0%
	SATAN	41%	25%
Remote administration	Back Orifice	66%	8.3%
	NetBus	41%	8.3%
	BackDoor-G	8.3%	0%
Port scanners	Nmap	41%	25%
	Ncat	16%	16%
	Strobe	25%	0%
Sniffers	NTSniff	25%	0%
Password crackers	Crack	33%	0%
	L0phtCrack	41%	33%
	Ntcrack	33%	16%
	John the Ripper	25%	0%

Table III : Administrator awareness and use of specific tools

A significant point that can be immediately observed from the results is that, in all cases bar one, less than half of the respondents are aware of the specific tools listed (and, in many cases, the aware proportion is nearer to a third or less). Even allowing for the fact that security may be only one of their responsibilities, one would

instinctively expect administrator awareness of tools such as password crackers and vulnerability scanners (which have received a fair degree of attention in the computing press) to be higher than that suggested. Comparing the results in Table III with those from Table II, it must be questioned whether the relatively high percentages that claimed to be aware of web and newsgroup information sources are actually using them effectively.

It can be observed that, in the vast majority of cases, the proportion expressing awareness is substantially higher than that in which the tools are actually used. In some cases, this can probably be explained by the fact that administrators are aware of tools that are not appropriate to their own systems (e.g. a Windows NT administrator may be aware of SATAN, but unable to use it as it requires a Unix platform). In other cases, such as Back Orifice and NetBus, one would not routinely expect system administrators to make use of them, other than to possibly evaluate their capabilities and the level of threat that they represent.

The general consensus amongst the respondents was that the hacker community was getting more benefit from the freely obtainable tools than the administrator audience.

When considering the survey percentages it is, of course, important to remember that they are based upon a small sample group and, therefore, they may not be fully generalisable to a larger audience. However, as will be seen from the discussion below, there are a number of reasons that would also suggest that low awareness and usage figures should be expected. Full details of the survey results can be found in Chiliarchaki (2000).

The argument is often offered that making tools with security analysis capabilities publicly available is a valid means of improving security, as system administrators have an equal opportunity to download and utilise them for defence as hackers do for attack. Although this is theoretically true, the practical situation is quite often not as clear-cut as the argument implies. The following factors should be borne in mind:

- Hackers have a greater level of motivation to obtain and utilise the tools, as it will directly assist their cause. For a system administrator, the use of an analysis tool has the potential to increase their workload if problems are exposed that need to be followed up. This is likely to act as a disincentive, particularly if they already have a significant workload.

- For system administrators, security will be only one of their responsibilities and, therefore, will only command a proportion of their available time. Conducting routine maintenance tasks and responding to user-related issues are likely to represent significant jobs in themselves. The most likely security issues to receive attention are password management, data backup and anti-virus measures.

- Hackers may get to learn about the availability of new tools more quickly than system administrators. The nature of hacker communities will mean that word may spread amongst them. An analogous community concept does not normally exist for system administrators and, therefore, they are more likely to find out about new tools via formal sources than word of mouth. Their best

chance of quick notification is likely to be via a security-related email list or discussion group, but administrators are only likely to be members of such lists if they are already reasonably well attuned to security.

With the above points in mind, it appears that system administrators are now playing a constant game of catch-up with the hacker community. Indeed, there are now courses available that administrators can attend in order to learn to think like the enemy (Lemos, 2000). During these courses, they can gain firsthand experience of using the tools and techniques practised by real hackers, as well as obtaining details of relevant countermeasures.

Although such training courses appear to be a logical means to enable administrators to fight abuse, there is also an argument that being an effective hacker is something that cannot simply be taught (i.e. it is more like a state of mind). As such, the lessons will not be effective against all classes of hacker – only those that similarly do things by the book. Therefore, whilst the administrators may then be able to repel the script kiddies, they may still have difficulty in dealing with the more dedicated and creative die-hards who may useless predictable methods of attack.

From a certain point of view, the idea that administrators should need to be educated to think like hackers at all is rather bizarre. The concept of 'know your enemy' is one thing, but if there are well-known vulnerabilities in a particular operating system, service or application then could an organisation not realistically expect its system administrator to keep abreast of the situation? The reality of the situation is, unfortunately, symptomatic of the fact that many system administrators are unaware of the intricacies of IT security and, of those that are, many do not have the time to routinely maintain their knowledge of the latest vulnerabilities and attacks, whilst also dealing with the routine tasks required to keep their systems running. The idea of having a dedicated security administrator (or, indeed, a team) may well be practical for larger organisations, but for administrators in small to medium sized organisations, security is often one consideration amongst many. Hackers, by contrast, can devote themselves whole-heartedly to discovering or learning the weaknesses and then exploiting the knowledge that they have obtained.

TYPES OF CONTROLS

Even though many exploits are based upon vulnerabilities that have been known for some time, the problem is a difficult one to keep on top of. The SANS Institute has identified several reasons why vulnerabilities may remain (Noack, 2000):

- 1.2 million new computers are added to the Internet every month;
- there is lack of security experts to address the problems;
- the number of vulnerabilities continues to grow and there is no priority list for dealing with them.

A number of options can be considered as potential top-level responses to the threats posed by automated analysis and attack tools as a whole. Unfortunately, it is quite easy to identify potential flaws in each case.

- Criminalising the illegal use of security analysers and vulnerability scanners. This is essentially the approach taken as part of the Draft Convention on Cyber-crime, proposed by the Council of Europe (CoE, 2000). Article 6 of the Convention relates to 'Illegal Devices' and prohibits the unauthorised creation, distribution and use of programs that may assist in illegal access, system interference and the like. However, this proposal has caused concern in the security community, as it will also cause difficulties for those wishing to conduct legitimate activities to identify and overcome weaknesses in their own systems (Goodwin, 2000). The key issue that the Convention is seeking to address is clearly the *illegal* use of such tools. However, if the distribution of the tools is restricted to prevent hackers from getting hold of them, how are legitimate administrators meant to do so?

- Payment-based availability. Having to pay to obtain the tool is likely to be a disincentive to the casual hacker. However, given the scale of software piracy, it is very unlikely that this would be a barrier for long and cracked copies of desirable tools would be circulated in warez communities.

- Making the accessibility of the tools registration-based, such that users have to provide personal details before downloading them. This is already the case with some tools, such as L0phtCrack. Unfortunately, of course, there is nothing to stop people providing bogus information.

- Incorporating technical restrictions into the tools. For example, only enabling a vulnerability scanner to target machines in the same network domain – to limit the potential for misusers to scan remote systems.

The flaws identified mean that, while the measures above may help to reduce the problem, none of them can be considered a complete solution. It can also be observed that these methods will only be effective in cases where tools originate from sources who have a positive motivation for releasing them (i.e. to help improve security). Where tools originate from the computing underground, the opposite motive is likely to be true, so any considerations regarding safeguarding the tools capabilities or their distribution would be irrelevant.

The countermeasures with the most chance of success are those that are not directly related to the tool software itself. For example, from a technical perspective, the installation of a firewall will confound vulnerability scanners by intercepting the scan requests and blocking them. At a more procedural level, conscientious system administration will help. It must be accepted that the tools are available and hackers will gain access to them, so responsible system administrators have an obligation to use them as well. It should be regarded as a routine security task rather than a matter of choice. If vulnerabilities are identified, then they should obviously be fixed.

When considering each category of tool separately, the possible countermeasure options are clearer. For example:

- Standard anti-virus software is able to detect the presence of code relating to systems like Back Orifice and NetBus.

- Other software can be used to detect the activity of vulnerability scanners. For example, Courtney and Gabriel are two tools that can be used to detect and monitor SATAN probes (see http://ciac.llnl.gov/ciac/ToolsUnixNetMon.html). Alternatively, attention can be diverted away from sensitive systems via the use of honeypots (special software that is designed to fool hackers by appearing to be legitimate servers/services, whilst enabling their actions to be tracked without damage to real systems).

- Packet sniffers can be foiled by encrypting sensitive network traffic and/or by detecting and disabling network cards operating in promiscuous mode.

- The threat posed by password crackers can be significantly reduced if an organisation follows appropriate password procedures (e.g. ensuring that passwords are based upon non-dictionary words and are at least 8 characters long, using a combination of alphanumeric and special characters if possible). This forces the cracker to revert to brute force attack methods (i.e. laboriously trying each possible character combination in turn).

This suggests that, whilst there is not a universal quick fix for the automated attack problem, it is possible to incorporate effective safeguards against the individual categories.

The discussion above has largely considered controls that relate to the user community. It is also worth noting that a significant proportion of the tools work by making use of known bugs in operating systems, Internet servers and the like. It can, therefore, be argued that if vendors paid sufficient attention to security in the first instance then such opportunities would not exist. Indeed, this is often the defence used by some authors to justify the public release of their tools (often with the accompanying intention of naming and shaming the associated vendor, to reinforce the point that action needs to be taken). For example, this was the claimed motivation with Back Orifice, where even the name was intended as a side swipe at Microsoft's Back Office suite. However, the obvious flaw in this approach was that Microsoft users, rather than the company itself, suffered as a result of the tool being shared.

CONCLUSIONS

Unless dramatic changes are made, it seems probable that the problem of security vulnerabilities will not only remain, but will become worse. The reasons for this are twofold. Firstly, as new software emerges, offering more complex functionality, the potential for unforeseen vulnerabilities is almost inevitable. Secondly, the increasing proliferation of Internet systems means that computers incorporating such vulnerabilities will be more widespread, thereby offering more opportunities for automated analysers to be used.

In terms of the aforementioned dramatic changes that would help to avoid this undesirable scenario, possibly the most fundamental is for software vendors to afford significantly more attention to security during the design, development and testing of their products. From a commercial perspective, security does not appear to be as relevant an investment as activities such as marketing. However, it should be

recognised that attention to the issue in advance could then avoid undesirable bad publicity later, which could otherwise serve to undermine marketing efforts and product image.

It should also be recognised that even with an improved effort to focus on security, some vulnerabilities may still slip through into released products. As such, contingency measures are still needed. From the vendor's perspective, this requires fast response in order to offer a remedy, in the form of patches and upgrades. Current evidence suggests that many vendors are already responsive in this sense and do act quickly to make solutions available. Possibly the more significant aspect at this stage is ensuring a response from the user perspective. The successful misuse of vulnerability scanners and exploit programs is based upon the fact that known security holes have not been addressed – holes that have sometimes been recognised for years. It is, therefore, necessary to make the user community more receptive to the fact that software updates may be necessary and that, in most cases, they are already available.

REFERENCES

Bicknell, D. (1995), "Influence of Satan divides IT industry", *Computer Weekly*, 13 April 1995, p4.

Chiliarchaki, P. (2000), *Security Analysers – Admin Assistants or Hacker Helpers?* M.Sc. thesis. Department of Communication and Electronic Engineering, University of Plymouth, Plymouth, UK. September 2000.

CoE. (2000), *Draft Convention on Cyber-Crime (Draft No 19)*, Council of Europe, PC-CY (2000) Draft No 19. Strasbourg 25 April 2000.

Delio, M. (2000), "MS Server Attack Tool Unleashed", *Wired News.* 16 August 2000. http://www.wired.com/news/technology/0,1282,38259,00.html.

Docekal, D. (2000), "IMPORTANT note to all NT/W2K IIS admins/users", Posting to Windows NTBugtraq Mailing List, 14 August 2000.

Farmer, D. and Venema, W. (1995), *Security Administrator's Tool for Analyzing Networks – SATAN*. General Information. http://www.fish.com/~zen/satan/satan.html

Goodwin, B. (2000), "EU cybercrime treaty puts users at risk, warn experts", *Computer Weekly*, 24 August 2000, p3.

Lemos, R. (2000), "Hack university: Learning from the pros", ZDNet News report, 12 July 2000. http://www.zdnet.com/zdnn/stories/news/0,4586,2601980,00.html.

Noack, D. (2000), "The Back Door Into Cyber-Terrorism", *APBnews.com report*, 2 June 2000.

Venter, H.S. and Eloff, J.H.P. (2000), “Network Security Health Checking”, in *Proceedings of IFIP/SEC 2000: Information Security, 16th World Computer Congress 2000*. Beijing, China, 21-25 August 2000, pp287-290.