Vendor-Supplied Backdoor Passwords - A Continuing Vulnerability

Astrid Todd

Vendor-Supplied Backdoor Passwords
- A Continuing Vulnerability

By Astrid Hoy Todd

GIAC Security Essentials Certification (GSEC)
Practical Assignment Version 1.4b, Option 1

August 29, 2003

Table of Contents

Abstract:

Vendor-supplied passwords embedded in software/hardware continue to be a securing vulnerability. Securing your network against vendor-supplied backdoor passwords is an ongoing process of staying informed through security mailing lists and bulletins, increased scrutiny of software/hardware before purchase, intense review of vendor documentation, application of vendor-supplied patches, and proper handling of default passwords/accounts. Several alternative emergency access schemes have appeared on the horizon that may replace the need for vendor access passwords altogether. In particular, open source software is emerging as an exciting new direction for software applications that effectively solves the problem of vendor-supplied backdoor passwords. This study covers documented and undocumented software/hardware backdoor passwords that are either intentionally or unintentionally placed by the vendor. Management and policy issues such as vendor control, vendor disclosure and design practices are also analyzed.*

Overview of Vendor-Supplied Passwords

VSBPs have always been built into software/hardware since the beginning. A VSBP is a password that the vendor encodes in the software/hardware for the purpose of future access. In the past, users shared access to mainframe security by using usernames and passwords. The system engineers controlled this access and often stored plaintext passwords in files, sure that no user had the knowledge to locate them. They often placed VSBPs in the systems so that they would have access to fix the system if the less knowledgeable user got locked out of the systems, for example.

As computer systems became more accessible and networking emerged, this “security through obscurity” plan became impossible to maintain. The presence of VSBPs forces you to extend your trust to other parties besides the vendor. It is not just the vendor who has access, but it includes any customer who has received the VSBP from the vendor because they experienced an “emergency,” the tech personnel employed by the vendor and the also the obscurity of the password itself.[1] The majority of VSBPs were built-in with vendor remote access in mind, so that when the VSBP is found, the vulnerability is usually in the area of remote access.

Intentionally-Placed VSBPs

VSBPs may be intentionally placed by the vendor for several reasons. Generally, they are used as maintenance trapdoors that the vendor uses for future access when problems occur.

*****For the purpose of brevity, the term “Vendor-Supplied Backdoor Password” will hereafter be referred to as a “VSBP “

They may be documented and well-known by the public or undocumented and unknown by the public. If the VSBP is undocumented and unknown by the public and then is discovered, it could be used as an entryway for malicious attacks. In the case of undocumented VSBPs, concerns have arisen (founded or unfounded) about more nefarious vendor purposes.

<u>Documented Backdoor Passwords-</u>

Documented backdoor passwords are passwords that are widely known through vendor documentation and are generally known by the public. This category includes default passwords and accounts. These passwords/accounts are built into the software/hardware and are managed by the administrator.

<u>Default Passwords-</u> Since they are documented and widely known, one would assume that administrators know to automatically change them upon installation of software/hardware, but this is not the case. Through lack of training (or failure to read the accompanying documentation), an administrator may or may not realize that these accounts even exist and leave them at their default settings.

Even the federal government is vulnerable to this sort of mismanagement, as evidenced by a GAO (General Accounting Office) report. In the report, the GAO describes the security problems discovered in “Pay.gov”, which is the Internet portal sponsored and managed by the Department of the Treasury’s Financial Management Service (FMS). Highlighting the lack of control over user accounts and passwords as a security breach, the report stated that, “A commonly known vendor-supplied password was not removed from one server.”[2] Another issue to consider when dealing with default passwords is that if you upgrade the product, this process can change the account passwords to a new default. One must remember to change the default accounts <u>after</u> applying the updates.[3]

<u>Undocumented Backdoor Passwords-</u>

Undocumented backdoor passwords are ones that are built into the software/hardware and are not widely known by the public. The vendor desires to keep these passwords secret, so that they may use them for various purposes such as maintenance trap doors or maybe for other unknown purposes. These VSBPs are generally kept private by the vendor employees. One might argue that having a vulnerability is okay as long as no one knows about it. While this might be true to an extent, once the VSBP is found, the finder may just choose to exploit the vulnerability instead of announcing it or fixing it. In effect, a hidden

VSBP is basically just a time bomb waiting to explode.

With the advance of programming knowledge, many of these passwords have been discovered either intentionally or unintentionally by users, opening up a security hole in systems. Once these VSBPs are discovered, they are shared on security mailing lists. One of the best known mailing lists is “Bugtraq” and is found on the Security Focus website at http://www.securityfocus.com/archive/1. Another well-known list is Security Tracker at http://www.securitytracker.com/. Participants on these mailing lists discuss the vulnerabilities, test them and decide whether to contact the vendor about the problem. Sometimes users contact the vendor directly first, before they announce the vulnerability to the world. The CERT Coordination Center is the major reporting center that sends out advisories after the vulnerabilities have been found and verified. The vulnerabilities can be viewed at http://www.cert.org/. The CVE (Common Vulnerabilities and Exposures) list is a dictionary that assigns an identifier number for vulnerabilities discovered by others and can be accessed at http://cve.mitre.org. This way there is a common identifier for these vulnerabilities when they are referenced in the future.

Vendors have had mixed reactions to the announcement of undocumented VSBP vulnerabilities. The vendor must ultimately decide whether the discovery warrants their attention based on costs. These costs can be measured in money, time, and public confidence in their product. If the fix means a complete redesign of their software/hardware, they are obviously not inclined to do so. The easier the fix, the more likely the vendor is to provide a fix. Generally, the fix is provided in the form of a patch or workaround. In July 1998, The Telecom Security Group sent out an advisory about backdoor passwords in Nbase’s ND208, NH215, and NH2016 switches. These passwords could not be disabled and they could be applied to both the serial and telnet console ports. The combination of <any> username and the password forgot or debug would grant full access to the switch. After the Telecom Security Group sent a copy of the advisory to Nbase, Nbase discussed the problems “at the highest levels” and made a conscious decision not to correct them. Eventually, Nbase “fixed” the problem by simply changing the former debug password of “debug” to the new debug password of “debug0” and the former lost password recovery password of “forgot” to the new recovery password of “forgotten.”[4]

Some vendors appear to be more proactive when confronted with a vulnerability advisory on one of their products. An example of this is the advisory on Intel’s NetStructure 7180 E-Commerce Director, which is a high-end Internet commerce product that regains the speed lost by servers running secure transactions. It was found to have 2 undocumented accounts called, “servnow” and “root,” each with a password that was generated from the MAC address. When advised of the problem, Intel offered this statement in response, “Intel Corporation takes all comments and publications about the security of our equipment seriously. The solutions offered in the security alert highlighted many of the security recommendations already present in the user documentation.”[5] Even though they appeared to be proactive, their official statement almost seemed to say, “just read the documentation.”

It is also surprising how easily these passwords may be obtained. For example, an undocumented “access level” password was discovered in various versions of 3Com’s “intelligent” and “extended” switching software for Lan Plex/Corebuilder switches. At least one user says that he obtained the VSBP easily from techs who tried to help them get back into a locked out system. Another account of the incident stated that the password was found by a user who was perusing the 3COM web site looking at an upgrade file for 3COM devices. They used a simple UNIX command called “strings” which displays all printable characters in a file and found a list of all the “secret” passwords unencrypted in the file.[6] This account is a “debug” account in addition to various known accounts such as admin, etc. with a password of “synet” on shipped images. This account + password has all the privileges of the admin account plus some “debug” commands that are not available to anyone else. Apparently, one may change all the other access passwords without having to know the old passwords. This means that someone with access to this secret “password” can lock you out of the switch completely. Also, this allows access to the underlying OS shell, where serious problems could occur.[7] Several people involved in this posting suggested that 3Com be informed so that they could release a security advisory. They apparently did just that, because 3Com soon after issued an advisory for a “debug” and “tech” account passwords of “synnet” and “tech” on their Corebuilder and SuperStack II switches which could be accessed via telnet.[8]

Another example of an easily found VSBP was the UTSTarcom case. UTSTarcom placed 2 hidden accounts in its BAS-1000 broadband subscriber management system. These accounts were used by the vendor to give full system access. Also, the customer cannot see these users logged, remove them, or change the access levels or passwords. The only way to do this is for the customer to log in under the password itself.

In a posting on the Security Tracker website, Scott Cameron shows that when the UNIX “strings” command is used, it becomes quite simple to find the passwords as the following command output shows:

```
-- begin --
Development engineer (this option is restricted)
guru
Field engineer (this option is restricted)
field
Management user with full system privileges
manager
Management user with limited write privileges
administrator
Management user with read-only privileges
```

```
operator
-- end --
```

This shows that there are 2 access-levels beyond what the “manager” accounts can see, both “field” and “guru”'.

Going further through the strings, the following is revealed in plaintext:

```
-- begin --
MANAGEMENT_USERS
initializing module %s
initialized module %s
OPER
Failed to create permanent user "%s"
ADMIN
*field
FIELD
*3noguru
GURU
SNMP
DBASE
-- end --
```

This shows that the login name “field” has a password of “*field”. This account is approximately equal to the manager level accounts. One can also now know the login name “guru” has a password of “*3noguru”. This account has higher access to a few more system abilities that the customer would not ordinarily see. “When faced with this vulnerability, the company’s reply was that while they acknowledged that it was possible to encrypt the passwords to stop a strings command, they didn’t have any plans to do so.”[9]

All types of software/hardware can contain a VSBP. In April 2000, the Cerberus Security Team discussed a blatant VSBP in McMurtrey/Whitaker & Associates, Inc.’s Win32 e-Commerce shopping cart. Cart32, as it is called, had a secret hidden password called “wemilo” found at file offset (0x6204h). An attacker could have used this password to go to one of several undocumented URLs, such as http://charon/scripts/cart32.exe/cart32clientlist, and obtain a list of the passwords for each Cart32 client. Although the passwords were hashed, they could be embedded into a specially crafted URL that would set the cart’s properties to spawn a shell, perform a directory listing and pipe the output to a file called file.txt on the root of the C: drive when an order is confirmed. After doing this, the attacker would then create an illegal order and confirm it, thus executing the command.[10]

Basic hardware functions could also be affected by VSBPs. One of the best examples of this type of password is the system BIOS. BIOS settings have been around since the creation of PCs, and amazingly, so have the VSBPs associated with them. The system BIOS is the basic set of instructions that tell the computer how to gain access to the disk drives, keyboards, display, etc. The BIOS is required in order to boot the computer.

There are several types of BIOS, with four major vendors supplying the BIOS instructions including Award BIOS designed by Phoenix Technologies. Many cyber protection products such as FoolProof, and Cyberpatrol recommend that you password protect the BIOS settings. This is done because the BIOS settings load when the operating system loads, before any other program is started. If you forget the password, however, you may not be able to reboot the system or change settings. The solutions suggested for this problem are often involved, including clearing the CMOS using jumpers, removing the CMOS battery, or using password cracking software. However, some manufacturers have VSBPs built-in that allow you to bypass the password protection altogether. You are required to contact them and provide system information in order to obtain the VSBP. It sounds pretty simple; however, through security mailing lists, many of these VSBPs have become publicly known. As the following list of known factory-set passwords for different BIOS manufacturers shows, there are many to try:[11]

AWARDBIOS:
AW
AWARD
AWARD_PS
AWARD_PW
AWARD_HW
AWARD SW
AWARD_SW
Award SW
AWARD PW
_award
awkward
alfaromeo
J64
j256
j262
j322
01322222
589589
589721
HLT
SER
SKY_FOX
Syxz
aLLy
CONCAT
TTPTHA
aPAf
HLT
KDD
ZBAAACA
ZAAADA
ZJAAADC
djonet

AMI BIOS
AMI
ami
bios
setup
cmos
AMIDECODE
A.M.I.
AMI SW
AMI_SW
BIOS
PASSWORD
HEWITT RAND
Oder
A.M.I.
AMI!SW
AMI?SW
HEWITT RAND
alfarome
efmukl

Phoenix BIOS
phoenix

Compaq
compaq

Tinys
Tiny

Other known defaults

LKWPETER
lkwpeter
BIOSTAR
biostar
BIOSSTAR
biosstar
ALFAROME
Syxz
Wodj
PASS
PASSOFF
CONDO
BIOS
SETUP
CMOS
admin
system
J64

Having multiple passwords makes system compromise almost inevitable. Such is the case with Bardon Data Systems WinU version 5.1 and earlier, and their Full Control version 2.6 and earlier. WinU is a program for novice users that includes many tools for systems management, access control, event logging, web-browser oversight, etc. With this VSBP and variations, a remote attacker could easily gain administrative privileges to the product. The main VSBP was “Barry Smiler”; however, at least twenty-nine password variations were discovered by Nu Omega Tau and posted on BugTraq on Oct. 13, 2000.[12]

Some people believe that VSBPs could even be used by vendors for nefarious purposes. For example, an ex-NSA expert states that vendors may work with the NSA in providing backdoors for the NSA to carry out surveillance activities on the customers.[13] The furor generated in 1999 by Intel’s announcement that their Pentium III processors would all contain a PSN (processor serial number) proves that vendor surveillance by vendors is a physical possibility.[14] This built-in serial number would identify customers in online transactions and be used for tracking purposes. Under pressure from consumer protection and privacy groups, Intel decided to include a utility to turn off this feature.[15]

Unintentionally-Placed VSBPs

Often the vendor finds that backdoor passwords have been included in their released software/hardware packages. This could be the result of poor coding practices, forgotten passwords/accounts, or even placed there by unscrupulous programmers employed by the vendor.

Vendor Mistakes-

Vendor mistakes in the software/hardware is a common reason that software/hardware might contain VSBPs. One of the problems with software in particular is that design of applications is poor from the onset. Even companies like Microsoft have been accused of not properly designing or modeling their software before they start writing code. They get an idea, write it down, and then fix it later. A recent study shows that application security flaws are generally introduced early in the design cycle. According to the study, "Nearly two-thirds (62 percent) of applications that were accessed suffered from poor design and implementation choices that allowed access controls to be bypassed."[16]

Poor Coding & Design Practices- The video game Quake had several security flaws associated with it, including a VSBP intended for use by the program vendor Id Software. This VSBP allowed an attacker to remotely send commands to the Quake console. Specifically, the attacker would have to handcraft a udp packet with a header containing the rcon command and the password "tms."[17] The company stated that this was an honest mistake.

Protocols are also not immune to VSBPs. An example of this is the SNMP (Simple Network Management Protocol) which is the primary standard for Internet network management. SNMP services are included in operating systems, routers, switches, cables and DSL modems and firewalls. Many of these devices include hidden backdoor passwords that can be used to bypass the normal security checks. Hidden and undocumented community strings are present in the SNMP subagent, which may allow remote attackers to change most system parameters. A community string is basically a character string that is a cleartext password between an SNMP manager and an SNMP agent. A common value is the 6-character string "public".[18] Attackers can use these to do many things, including indirectly executing arbitrary commands on the system with superuser privileges. This vulnerability is compounded by the fact that these SNMP daemons are configured and executed by default.[19]

Forgotten Passwords/Accounts- Another problem that falls under the category of "vendor mistakes" is that of forgotten passwords or accounts. The vendor sometimes adds in temporary accounts that were designed for testing purposes, and then forgets to remove them.

Vendor mistakes don't always fall under the categories of poor coding and design practices or forgotten passwords/accounts. They fall under a separate category called the "oops" category. Red Hat's distribution of Piranha Linux Virtual Server (version 0.4.12) falls into this category. Piranha is a collection of utilities designed to administer the Linux Virtual Server. It contained a web-based G.U.I. that allowed system administrators to configure and monitor the server cluster.[20] Piranha contained a VSBP that allowed remote execution of the commands on the server. The password was discovered in the G.U.I. portion of Piranha by security vendor ISS (Internet Security Systems), located at http://xforce.iss.net**.**

This vulnerability was considered particularly serious because it affected entire systems and provided an opening into the network.[21] In this incident, the default password for the administrator account was supposed to be blank; however, by error, it was shipped with the administrator password set to "q". The only way to gain access to the system was to discover the secret password or delete the unknown one and assign a new one. So, in effect, before you could even administer Piranha, you had to eliminate this back door.[22]

Backdoor Passwords Placed by Unscrupulous Vendor Programmers- Another area of concern is the possibility of VSBPs being placed into software/hardware by vendor programmers without the vendor's knowledge. These VSBPs could also allow future access for the programmers without the vendor's knowledge. There are only a few actual reported instances of this type of situation because vendors are not anxious to report these types of problems once they are discovered. This types of negative press about their product could cause a loss of sales and possible canceled contracts.

The almost classic example of this situation was announced by the Microsoft Corporation in April 2000. Apparently, Microsoft engineers wrote in a VSBP in some versions of Windows NT. The secret password was "!seineew era sreenigne epacsteN" which is "Netscape engineers are weenies!" spelled backwards. It was reportedly discovered by "Rain Forest Puppy" who notified Microsoft about the vulnerability in an email message.[23]
The threat was at first widely reported as a way that hackers could use the backdoor to access Web-site management files that could lead to customer information on many major Web sites. However, it turned out that it affected only sites using a program called "Visual InterDev 1.0, (this was an old version of this software that was released in 1995), and from sites that installed anything from the 4.0 option kit for the Microsoft NT server software. Deleting the dvwssr.dll file from the affected software abolished the threat.[24] Because of the nature of the password wording, the whole incident seemed almost like a prank, with the password left in bold in the coding. It almost appeared that they wanted the password to be discovered. The engineers were reportedly going to be fired for this.

Concerns over employee-generated vulnerabilities have sometimes bordered on the edge of paranoia. In 1999, Richard A. Clarke of the National Security Council expressed great concern over the possibility of an "electronic Pearl Harbor" produced by software or hardware trapdoors lying dormant in the nation's critical infrastructure. Special concern was placed around the theory that many systems people are foreign born and theoretically could act as “digital moles” planting trapdoors in hardware/software, waiting for the right moment to strike.[25]

There was also great concern around the time of the Y2K scare, when Y2K “fix” companies were hiring foreign-born employees to fill the need for personnel to work on these projects. It was feared afterwards that some of these employees might have implanted VSBPs or trapdoors in the systems of the companies that they were working with to fix the Y2K problem. To this author’s knowledge, none of these “attacks” have occurred.

Securing Your Software/Hardware Against VSBPs That May Already Be Present in Your Software/Hardware

There are four main methods of keeping your software/hardware secure against VSBPs. They are:

- Proper Handling of Default Passwords/Accounts
- Staying Informed
- Use of Workarounds
- Application of Patches

Proper Handling of Default Passwords/Accounts- The obvious cure for default passwords/accounts is to change the default passwords/accounts upon installation of the software/hardware. Original default passwords should also be changed if there is a change to the operating system and, for a higher level of security, after each use of the account.[26] Always read your software/hardware documentation carefully before installation to find out about them.

Staying Informed- It is imperative to keep informed about the discovery of VSBPs. The best way is to be a member of a security mailing list such as the SANS newsbites mailing list. Also make sure you read the CERT advisory information located at http://ww.cert.org to keep informed about a possible vulnerability or workaround. Perusing the online postings at mailing lists like Bugtraq will also help.

Use of Workarounds- One of the ways of bypassing a VSBP security breach is through the use of workarounds. A workaround is a method of fixing the security end of the problem without changing the coding. They are a low-cost yet temporary fix. Many workarounds sacrifice the functionality of the software/hardware in an attempt to bypass the problem. For example, in the case of the backdoor passwords found in the Nbase switches in May 1998 (see p. 5 of this document), the most secure workaround was described as to not define an IP address for the switch, in effect disabling IP-based management. To disable the IP functionality, one would use the set-ip command to a random net 10 address. Because the 10 is not routed on the Internet and probably would not be routed on the LAN, this was considered safe. Patches are probably the most common fix cited, along with simply just changed the VSBP.

Application of Patches- Timely installation of patches when a VSBP vulnerability is found is essential. However, it is also important to realize that most of these vendor supplied patches are just "quick fixes." Often they only temporarily and superficially fix the problem. In the Cart32 example (see p.6 of this document), the patch was just that type of a "quick fix." SecuriTeam stated that, "this patch does in no way make the Cart32 software secure. It merely eliminates the two problems detailed in the Cerberrus Information security advisory CISADV000427. The security problems in this software are at a basic design level and may take several days for the vendor to fix."[27] The key is to apply the patches as soon as possible. Unfortunately, neither workarounds or patches actually "fix" the coding itself.

Securing Software/Hardware Without VSBPs

There are several methods for securing software/hardware without VSBPs. They are:

- Software Audits
- Alternative Emergency Access Schemes
- Legal Front Doors
- Open Source Software

Software Audits- One way to secure your software/hardware without VSBPs is to try to purchase software/hardware that doesn't have them in the first place. You could simply ask the vendor before purchase in writing if there is a VSBP included in their software/hardware. This may put some of the legal liability into the vendor's hands. You could also perform a software audit of the software they are purchasing. This is truly taking matters into your own hands, since most companies may not have the time to do this. In effect, the users on the security mailing lists are performing the audits for you, so it goes back to staying informed.

Alternative Emergency Access Schemes- Many administrators would argue that vendor access is still needed for software/hardware systems in case the administrator gets locked out. However, there are other methods to access software/hardware without the use of VSBPs. An example would be the scheme that the Cisco Corporation devised for its routers and other devices. In essence, the administrator reboots the device into a debugging mode, and then logs in the network. The key point to this procedure is that you need physical access to the device, instead of having one VSBP that could give access to many systems.[28] This practice adds another layer to a "defense in-depth" strategy of securing your network.

Another alternative access scheme is not to have a backdoor password, period. An example of this alternative in practice is shown with the Data Gator software. Data Gator is an application used to encrypt data on a Palm OS. There are no passwords with it, and no allowances for lost passwords. This is the default setting unless you make specific allowances with the vendor. It seems harsh, but very secure.[29] Customer knowledge has advanced far enough to enable them to secure their own software/hardware.

Legal Front Doors- Using a legal front door for the vendor to access software/hardware in case of an emergency is another method. Using this method, the customer creates their own VSBP, and then supplies the vendor with a password to access the software/hardware that the customer remains in control of. The vendor would have to build this capability initially into the software/hardware itself.

Open Source Software- The open source alternative is exploding. The rise of Linux helped create this explosion. Open source is a collaboration design model which includes a huge pool of developers and code that is shareable. Other benefits of open source include: reliability, cost of ownership, lower capital investment, greater flexibility, increased customer control, and application development that is faster and cheaper. Since the source code is available for everyone to see, the chances that an undocumented VSBP would survive the scrutiny of so many eyes is slim. Open source is gaining acceptance among I.T. professionals. In the past, they were much more inclined to choose vendor software, probably because of name recognition. Now with the well-publicized success of Linux and other open source applications, along with the reduced cost of switching to open source, they are more inclined to go with an open source product. For example, in a recent survey of 375 I.T. professionals, the majority of them said that they would choose open source for a new implementation over a proprietary vendor product.[30]

Open source software is the way to go to eliminate the threat of VSBPs. Open source is like a security audit that may not have necessarily been performed by a closed-source vendor. Closed source ultimately leaves control in the hands of the vendor. One might argue that the Microsoft “Netscape engineers are weenies” exploit (see p. 9 of this document) would not have occurred if the software was open source. If it had been open sourced, the coding would have been analyzed by a group of experts before it even hit the market.

Another benefit of open source over closed source is that it is designed to be changeable; however, closed source isn’t really designed with change in mind. If you don’t make the program changeable from the start, you will eventually encounter future problems.

Closed source software/hardware can be converted to open source. When this happens, the underlying security problems with closed source are often revealed.

An interesting example of this is when Borland/Inprise's Interbase Server database package was converted to open source. Interestingly, both the open and closed source versions contained a compiled-in back door account with a known password. The VSBP was fixed, plaintext, and easily located in the source code. This particular VSBP allowed any local user or remote user to be able to access port 3050/tcp[gds_db] and then the user could access any database object. This VSBP was not introduced by malicious programmers, but was added by Borland before the program went open source possibly between 1992 and 1994.[31]

Another major problem with closed source coding is that the vendor doesn't seem to have a great deal of liability on their shoulders. Suing a closed source company doesn't seem to be very productive because almost all of the licenses disclaim warranties, and courts have not held software companies liable in these cases."[32]

This doesn't mean that open source is faultless. As in the Red Hat/Linux/Piranha example discussed previously on p. 9 of this document, there initially appeared to be a VSBP. However, the product was Version 0.4.12, which means that the product was still in development, less than half-finished. The problem was dismissed and corrected immediately after the product was released. At first, Microsoft claimed that the infamous "Netscape engineers are weenies!" VSBP was fixed quickly; then they said that there was never a problem to begin with. Because it is closed source, there is no real way of verifying either examination result. There is also no way of knowing how long the VSBP was present, or who may have exploited it. Herein lies some of the basic problems with closed source code.[33]

Open source is readily gaining acceptance with more than just I.T. professionals. There has even been legislative action introduced at the state level in Oregon to require state agencies to "consider the use of open source software for all new software acquisitions." The bill doesn't mandate the use of open source software as it is in California's Digital Security Act. Open source software is required to be included on the list of approved state products. The bill even requires state agencies to provide justification for using proprietary software.[34] The argument for open source as an alternative to guard against VSBPs can be summed up by visiting the Slashdot website at http://slashdot.org/articles/01/01/11/1318207.shtml and viewing the postings where the users have coined a new formula, "Open source = no backdoor."

Conclusion:

VSBPs should be relegated to cyber history, as there are many alternatives to them. However, as this study shows, discovery of these VSBPs is still occurring, and they remain a security threat. These VSBPs are documented, or undocumented, and it is not known how many of them have been exploited. In the end, it all comes down to how much control you want the vendor to have. As Kevin Savoy wrote in his article, "The best way to trust a vendor is to know that you are not relying on mere trust alone."[35] Control should be in the customer's hands, and there are several alternative access schemes and methods to secure your software/hardware without the use of VSBPs. Customers are entitled to full disclosure on their software/hardware and then should be able to make their own decision on whether or not to purchase the software/hardware that contains a VSBP. Finally, with its full disclosure of coding, open source is the way to go to insure flexible and secure software/hardware.

References

[1] Stutz. Michael. "Software Backdoors Let out Draft." Wired News. 19 May 1998. URL: http://www.wired.com/news/technology/0,1282,12381,00.html (30 July 2003).

[2] General Accounting Office. "Information Security: Computer Controls over Key Treasury Internet Payment System." GAO Website -Report No. GAO-03-837. (30 July 2003).
URL: http://www.gao.gov/atext/d03837.txt (8 August 2003).

[3] CERT Coordination Center. "Unix Configuration Guidelines." DOD-CERT ONLINE.
URL: http://www.cert.mil/techtips/unix_configuration_guidelines.htm (25 July 2003).

[4] The Telecom Security Group. "N-Base Vulnerability." TTSG Vulnerability Advisory. 4 July 1998.
URL: http://packetstormsecurity.nl/new-exploits/nbase.txt (2 August 2003).

[5] Oblivion. “NetStructure 7180 backdoor vulnerability.” SecuriTeam Website 13 May 2000.
URL: http://www.securiteam.com/exploits/5HP0D0U1FS.html (11 August 2003).

[6] Stutz. Michael. “Software Backdoors Let out Draft.” Wired News. 19 May 1998.
URL: http://www.wired.com/news/technology/0,1282,12381,00.html (30 July 2003).

[7] Monti. Eric. “Backdoor passwords in 3com switches, routers, smart hubs.” SecureNow Website. 5 May 1998.
URL: http://securenow.virtualave.net/misc/3com.switches.routers.undocumented.backdoors.html (13 August 2003).

[8] Internet Security Systems. 3Com Advisory 51498. 14 May 1998.
URL: http://www.iss.net/security_center/advice/Exploits/Defaults/Password/Backdoor/default.htm (27 August 2003).

[9] Cameron. Scott T. “UTStarcom BAS-1000 Broadband Subscriber Management System Has Backdoor Accounts With Known Passwords That Give Remote Users Control of the System.” Security Tracker Website. 24 August 2002.
URL: http://www.securitytracker.com/alerts/2002/Aug/1005134.html (31 July 2003).

[10] David and Mark Litchfield. “Cart32 secret password Backdoor.” Cerberus Information Security Advisory – CISADV000427. 27 April 2000.
URL: http://packetstorm.trustica.cz/advisories/cerberus/CISADV000427.txt (25 July 2003).

[11] “Bypassing BIOS passwords.” SecuriTeam Website 24 April 2000.
URL: http://www.securiteam.com/securitynews/5CQ0G000IG.html (11 August 2003).

[12] Nu Omega Tau. “WinU Backdoor passwords!!!” Security Focus Archives. 13 October 2000.
URL: http://www.securityfocus.com/archive/1/139768 (30 July 2003).

[13] Knight. Will. “Ex-NSA expert warns of concealed backdoors.” ZdNet UK News. 25 September 2000.
URL: http://news.zdnet.co.uk/story/0,,s2081591,00.html (25 July 2003).

[14] Sprenger. Polly. “Intel on Privacy: ‘Whoops!’”. Wired News. 25 January 1999.
URL: http://www.wired.com/news/politics/0,1283,17513,00.html (29 August 2003).

[15] “Intel Pentium III Processor Serial Number.” Center for Democracy & Technology. 2001.
URL: http://www.cdt.org/privacy/issues/pentium3/ (29 August 2003).

[16] Jaquith. Andrew. “The Security of Applications: Not All Are Created Equal.” @ stake Website. February 2002.
URL: http://stake.com/research/reports/acrobat/atstake_app_unequal.pdf (27 August 2003).

[17] Zielinski. Mark. “ID games Backdoor in quake.” Insecure Website. 1 May 1998. URL: http://www.insecure.org/sploits/quake.backdoor.html (25 July 2003).

[18] Stevens, W. Richard. TCP/IP Illustrated, Volume 1. Reading: Addison Wesley Longman, Inc., 1994. p. 362.

[19] X-Force. “snmp-backdoor-password (8499).” Internet Security Systems. URL: http://xforce.iss.net/xforce/xfdb/8499 (12 August 2003).

[20] X-Force. “Backdoor Password in Red Hat Linux Virtual Server Package.” SecuriTeam. 25 April 2000. URL: http://www.securiteam.com/unixfocus/5AQ0E000IY.html (5 August 2003).

[21] Middleton. James. “Linux virtual web server hit by “extremely serious” backdoor password.” PC Magazine Online. 10 May 2000. URL: http://www.pcmag.co.uk/News/1105281 (5 August 2003).

[22] Petreley. Nicholas. “THE OPEN SOURCE: Weenies shoot for the backdoor prize in big contest between Microsoft and Red Hat.” Info World. 8 May 2000. URL: http://www.findarticles.com/cf_0/m0IFW/19_22/61948133/p1/article.jhtml?term=vendor-supplied+backdoor+password (15 August 2003).

[23] Geralds. John. “Backdoor found in Microsoft software.” PC Magazine Online. 17 April 2000. URL: http://www.pcmag.co.uk/News/602239 (5 August 2003).

[24] Weil. Nancy. “Microsoft Server Flaw Called Rare.” PC World Online. 14 April 2000. URL: http://www.pcworld.com/resource/printable/article/0,aid,16280,00.asp (8 August 2003).

[25] Ackerman. Robert L. “Hidden Hazards Menace U.S. Information Infrastructure.” Signal Magazine. August 1999. URL: http://www.us.net/signal/Archive/August99/hidden-aug.html (25 July 2003).

[26] Savoy, Kevin. “In Vendor We Trust?” Information Systems Audit and Control Association. 2001. URL: http://www.isaca.org/art16.htm (31 July 2003).

[27] Pond. Weld. “Cart32 contains a secret password backdoor.” SecuriTeam Website. 28 April 2000. URL: http://www.securiteam.com/securitynews/5GP0L2A15S.html (31 July 2003)

[28] Stutz. Michael. “Software Backdoors Let out Draft.” Wired News. 19 May 1998. URL: http://www.wired.com/news/technology/0,1282,12381,00.html (30 July 2003).

[29] “PalmSource: Making PDAs More Secure.” Information Week. 12 December 2000. URL: http://www.informationweek.com/story/showArticle.jhtml?articleID=6510123 (30 July 2003).

[30] Langham. Martin. “Open Source Content Management arrives.” IT-Director Website. 23 May 2003. URL: http://www.it-director.com/article.php?articleid=10868 (25 July 2003).

[31] Diesel. Dave. “Interbase Backdoor Secret for Six Years, Revealed in Source.” Slashdot Website. 11 January 2001.
URL: http://slashdot.org/articles/01/01/11/1318207.shtml (25 July 2003).

[32] Wheeler. David. Secure Programming for Linux and Unix HOWTO. 2001.
URL: http://ftp.redhat.com/pub/redhat/linux/7.1/ja/doc/HOWTOS?Secure-Programs-HOWTO (12 August 2003).

[33] Petreley. Nicholas. “THE OPEN SOURCE: Weenies shoot for the backdoor prize in big contest between Microsoft and Red hat.” Info World. 8 May 2000.
URL:
http://www.findarticles.com/cf_0/m0IFW/19_22/61948133/p1/article.jhtml?term=the+open+source%3A+weenies+shoot+for+the+backdoor+prize+in+big+contest+between+Microsoft+and+red+hat (5 August 2003).

[34] Miller. Robin. “Oregon Considers Open Source Software Legislation.” NewsForge. 6 March 2003.
URL: http://newsforge.com/newsforge/03/03/06/018222.shtml?tid=4
(13 August 2003).

[35] Savoy, Kevin. “In Vendor We Trust?” Information Systems Audit and Control Association InfoBytes. 2001. URL: http://www.isaca.org/art16.htm
(31 July 2003).