Global Information Assurance Certification Paper

Passwords are DEAD! (Long live passwords?)

David Beverstock
GSEC Practical v.1.4b
Submitted June, 2003

1.0 Abstract

Passwords are dead. Computer security professionals presently work in a time period of constant innovations in computer and software technology, astonishing computer power, and numerous and varied threats and attacks on computer systems. In this modern world, there are several viable alternatives to passwords for authentication into computer systems with important functions or containing sensitive data. Passwords are ubiquitous. Removing passwords from all proprietary computer operating systems would be a slow, costly process. Passwords, if used appropriately, provide a low risk, cost effective, and familiar interface to authenticate into systems of low functional importance, or that don't contain sensitive data. The strength of passwords, or an alternate authentication system should be proportional to the value or importance the system that requires protection. Passwords have algebraic, computer implementation, and human behavioral properties that for low value systems, are risks that require mitigation through policies and technical controls. For systems of high importance these same properties are critical flaws which no longer have strong mitigations which render passwords unsuitable for use in this time period. Following a brief history and definition of passwords, this paper will show three properties of passwords that render passwords risky or unsuitable for use. Suggestions for mitigating risk from these properties is covered briefly. Current attacks on passwords, illustrated by a simple experiment, and future trends in computing that will obsolete password use are highlighted. A short description of a risk analysis as applied to authentication is sketched out and pointers are given to alternative forms of authentication.

2.0 Passwords

Passwords have been associated with computers from nearly the beginning of the electronic computer. "Passwords are the most commonly used computer security tool in the world today."[1]. As one may imagine, issues with passwords (and guessing passwords), developed concurrently with the early use of computer systems, most visibly in CTSS and RAX in the early 1960's and 70's[2].

1. Skoudis, p. 279
2. Smith, p. 10-14

2.1 Passwords defined, considerations for passwords

The dictionary definition of a password is: “something that enables one to pass or gain admission”, and “a: a spoken word or phrase required to pass by a guard”, and “b: a sequence of characters required for access to a computer system”. Applied to the business setting, use of a password implies there are two populations, “insiders”, and “outsiders”. A password differentiates between the populations: insiders know the password, outsiders do not. Within a business enterprise comprised of all insiders, there can be layers of insiders and outsiders. For example, a computer workstation end user is an outsider relative to the group of computer system administrators.

A “good” password is easily remembered by insiders, but not easily guessed by outsiders. All passwords should be resistant to password guessing attempts. Password “cracking” is a term that refers to using an automated computer program to guess passwords on a computer system. Passwords are described as “weak” or “strong”. Later in the paper it will be shown that using different kinds of characters in passwords decrease the chance that the password will be guessed. Weak passwords have relatively few characters, perhaps 4 to 7, and don’t have a mix of characters and character case, e.g. “H2TU”, “WAV4”, “ADMIN” or “admin”. Weak passwords are relatively easy to guess. Strong passwords contain different types of characters: numbers, symbols and letters in upper and lower case, and are comprised of 8 to 14 characters (or more, depending on operating system support), e.g. “3w9&t$l^%)6h”, “m!_Wo{}"b69$)d” or “IouR1s!4nc(V”; (note that these last examples are strong passwords, but probably not easily remembered by a person). Strong passwords are not easily guessed.

In the business setting, password policies occasionally seem to have been created with little regard to the value of the resources the passwords were to protect, and/or perhaps with little regard to the environment passwords were being used in. Perhaps with a little thought the reader could think of a few situations they have had personal experience with. Reviewing the password exploit of the 1994 Citibank hack where $400 million was taken, it’s clear that the Citibank authentication system did not offer the level of security commensurate to the value of the computer system the authentication was meant to protect[1]. Organizational and system password policies drive good password selection. All business groups and business systems should have clearly defined and well established password policies. The passwords created by users should be checked for compliance with the policy[2]. The most important password policies are done so at a system level. This recognizes that all systems within a business organization do not carry the same risk and that all systems within an organization don’t carry the same value. If a small business has only 2 computer systems, a human resources/payroll system, and a library management system for the corporate reading room, the payroll system has the greater value, and thus the password policies for the payroll system would require stronger passwords than those required for the library system. Good passwords are easily remem-

1. http://www.pbs.org/wgbh/pages/frontline/shows/hackers/whoare/notable.html
2. http://support.microsoft.com/default.aspx?scid=kb;EN-US;161990

bered by insiders, are not easily guessed by outsiders, and the required password strength is proportional to the value of, or risk to, the systems they are meant to protect.

Having made a functional definition for passwords, and indicated what weak and strong passwords are, the next section highlights the three issues and four common attacks that are important when considering the use of passwords in the business environment.

2.2 Issues

2.2.1 The algebraic issue with passwords

Randomness is defined as “lacking a definite plan, purpose, or pattern”. A high degree of randomness, or a lack of pattern in a password assures that a password won’t easily be guessed. The randomness of a password is represented by the total number of combinations in the set of characters that could create a password. Adapting an example from Smith (p. 63), take a simple 3 digit lock designed to secure travel luggage.

FIGURE 1. Master Lock brand Classic Black Luggage lock

For our example we will think of the combination for the lock as being a password. Each digit in the combination, or place in the password, can take values from 0 to 9, a total of 10 unique values, $(0_1, 1_2, 2_3, 3_4, 4_5, 5_6, 6_7, 7_8, 8_9, 9_{10})$. The password for a 3 digit luggage lock would have 1,000 different combinations. So, a four digit lock allows 10,000 different passwords. Expressing this as an equation:

$$P = 10 \times 10 \times 10 \times 10 = 10,000 \quad \text{(EQ 1)}$$

P represents the possible number of passwords. We can generalize Equation 1 like this:

$$P = C_1 \times C_2 \times \ldots \times C_z \quad \text{(EQ 2)}$$

C represents the total number of unique characters that each place in the password can take and *n* represents the total number of characters in the password. In the 4 digit luggage lock example *n* = 4. Equation 1 can be expressed in this form:

$$P = 10^1 \times 10^1 \times 10^1 \times 10^1 = 10^4 = 10,000 \quad \text{(EQ 3)}$$

Applying the additive law of exponents to Equation 2, we have,

$$P = C^n \text{ so } P = 10^4 \quad \text{(EQ 4)}$$

which we know is 10,000. *P* can be referred to as the "password space". Using the equations and methods illustrated above we can extend this concept to the alphabetical characters that can be used to create passwords. Since there are 26 letters in the English alphabet, we can calculate the password space using lower case letters only in a 6 character password.

$$P = 26^6 \quad \text{(EQ 5)}$$

So, there are 309 million unique, passwords in a 6 character lower case only (or upper case only) password. Using upper and lower case letters in a 6 character password,

$$P = 52^6 \quad \text{(EQ 6)}$$

allows over 19 million unique passwords. Note the exponential increase in the number of unique passwords because we increased the number of unique characters for each place in the password. Using upper and lower case letters and numbers in an 6 character password,

$$P = 62^6 \quad \text{(EQ 7)}$$

yields 56 billion unique passwords. Finally, using the 26 upper and 26 lower case letters, the 10 numbers, the 33 symbols and punctuation marks in a 6 character password,

$$P = 95^6 \quad \text{(EQ 8)}$$

yields over 735 billion possible, unique, passwords.

Increasing the variety of characters and number of characters in a given password greatly increases the password space. A password selected from a small password space, like that illustrated in Equation 5 will be a weak password. A password selected from a large password space like that represented in Equation 8 will be a strong password. Weak passwords are easy to guess. Strong passwords are difficult to guess.

Table 1, "Combinations of ASCII Printing Characters and Resulting Password Space," on page 18 tabulates the various combinations of three types of printing ASCII characters, letters, numbers and symbols, and the resultant password space. The data in this table clearly illustrates how password space increases with increasing types of characters in the password, and increasing password length. A two character password using only a single case of letters has a paltry password space of 676. This password could easily be exhausted by a determined teen-ager using trial and error guessing over a short afternoon. A six character password has at least 309 million, and at most 735 billion password possibilities. An 8 character password has at least 209 billion, and at most 6.6 quadrillion unique passwords. A 14 character password using only the 3 major character groups has a remarkably huge password space of 4,877 septillion ($4{,}877 \times 10^{24}$) unique

passwords. Finally, a chance to read about some meaningful numbers that are larger than our national debt!

In this section it has been shown that there is an easily understandable algebraic relationship between the number of characters in the password (the password length), and the types of characters used in the password. A password of a given length that uses 3 or 4 types of characters, letters, numbers, and symbols will be stronger than the same length password using only 1 or 2 types of characters. A password using a given set of character types of length 8 characters is stronger than a password of 6 characters using the same types of characters. While Equation 4 does produce huge password space numbers that defy comprehension it is shown later, in Section 2.2.4, that a large password space no longer provides a strong mitigation from risk of passwords being guessed.

2.2.2 The computer technology on perspective on passwords

Two important interfaces between the human and the computer are the characters that the human types on the keyboard, and the characters the computer writes onto the computer monitor and prints on paper. Character set encodings have a rich history with roots in the radio and telegraph industry[1] and appear to be a vibrant area with recent development to accommodate the large number of cultures and languages present on our ever-shrinking globe[2]. The American Standard Code for Information Interchange (ASCII) published in 1968 as ANSI X3.4 (American National Standards Institute) define the most common computer character set in use today. For the purposes of this paper the ASCII character set will be referenced, and it is sufficient to note that other character sets are utilized in small numbers of different computer systems.

The ASCII character set contains 128 characters. Ninety-five characters find representation on the monitor and printed on paper, and the 96th, the backspace key, on the keyboard. For reference, Table 4 on page 23 is included to show the ASCII printing characters, and their official description. Thirty-two characters that do not print or display on the monitor, the “control characters”, are left over from the old teletype and teleprinter devices of early computing. These 32 characters have found a new life! Use of any control character combination in a password is miraculously effective, significantly increasing the randomness and size of the password space, creating significantly stronger passwords. Looking back at Equation 8, we see that the *C* value now becomes 127 instead of 95. There is an exponential increase in the number of unique passwords. For an 8 character password using Control-Key combinations, the new password space is 68 quadrillion, over 10 times greater than using 95 characters in the password.

However, the characters that are allowed in an operating system’s passwords reduce the possible password space. Solaris passwords function predictably using any ASCII character combination except: Ctrl-C, Ctrl-Z, Ctrl-U, Ctrl-S, Esc, Tab, and in some cases # and @, yielding a predictable password space of 119 characters[3]. Table 5, “Control-Key

1. http://www.wps.com/projects/codes/index.html#TOP
2. http://czyborra.com/

Combinations (Non-Printing ASCII Characters)," on page 26 is included so that any reader can learn the Control-Key combinations and begin to incorporate them into their Solaris passwords.

The Windows operating system divides the ASCII Printing Characters into 3 groups: letters, numbers, and symbols[1] (all characters not defined as letters or numbers including the space symbol), a total of 95 characters. Table 3, "Windows Allowed Symbols for Passwords," on page 21 is provided for easy reference for Windows operating system support for symbols. Windows doesn't appear to support use of control characters in passwords[2]. The author was not able to find any Microsoft documentation stating control characters were not allowed in passwords. Just the same, the author's Windows 2000 Professional installation would not allow creation of a password using any control characters, although ^H did perform a backspace-delete and ^V appeared to paste from clipboard, and ^Z appeared to undo. "Control-Shift-6" and "Control-Shift-_" gave undetermined results.

Solaris has the potential to create stronger passwords than Windows. In fact, comparing Solaris to Windows in terms of password space, Solaris has 2.5 to 23 times greater password space than does Windows for passwords from 4 to 14 characters in length. For longer passwords Solaris would offer greater yet password spaces, because the exponential nature of the password space equation.

The effective password space for a computer operating system is limited by the support of the computer operating system for Control-Key characters. Solaris has the potential to create stronger passwords than does the Windows operating system.

2.2.3 The human behavior perspective on passwords

The effective password space is not dictated just by algebraic considerations, or operating system support. The password space is also determined by what characters people choose to put into their passwords, and in a broader sense, general human behavior around using passwords, i.e. if people use passwords at all.

People do choose to use weak passwords, or no passwords at all. "The SANS Top 20 - The Twenty Most Critical Internet Security Vulnerabilities..."[3], list as the seventh most critical Windows vulnerability and the tenth most critical Unix vulnerability, "Accounts with No Passwords or Weak Passwords"[4]. A SANS GSEC graduate relates his personal experience, 3 real world situations where no passwords were the usual as opposed to

3. http://docs.sun.com/db/doc/802-5826/6i9iclf5n?a=view
1. http://www.microsoft.com/windows2000/en/professional/help/default.asp?url=/windows2000/en/professional/help/windows_password_tips.htm
2. Windows allowed account name characters are included for completeness, Figure 4 on page 21
3. http://www.sans.org/top20/
4. http://www.sans.org/top20/#index

the exception[1]. For example, a medium sized organization has about 25-30% of users passwords that are found to be weak, on a monthly basis. Also, when reminded by email what a strong password is, the most of these users choose to create yet another weak password. Clearly using weak or no passwords on computer systems is a common issue, and a threat vector that must be taken into account when preforming risk analysis or designing new applications and systems.

People readily remember things that are meaningful to them, for instance their home telephone number, their home address, etc. People don't remember things that aren't meaningful to them. For instance, many people don't remember their personal cell phone number, because they don't ever have a need to call that particular phone number, and thus it has little meaning to them. But these same people easily remember their spouse's or significant other's cell phone number, even though the number is the same length and format. The loved one's cell phone number is easily remembered because the number has a greater meaning to the person and so is easily remembered. People who do easily remember their own cell phone number do so because the number has a larger significance, e.g. it is their business cell phone, and therefore an important tool for making transactions that earn them money or improve their job performance. The person remembers their cell phone number because it aids them to be successful in their career. Children with certain learning disabilities have difficulty correctly spelling certain vocabulary words. The learning disability is created because the child's perception of the vocabulary words is that the words are a meaningless set of characters. When teaching children with these learning disabilities vocabulary words, one method often used is to have the child write the word in the air using their finger as an imaginary chalk and pretending to write on an imaginary blackboard. Another method used is to have the child write the word in a box filled with sand using their finger as a stylus. Both these methods are successful in enabling children to successfully spell their vocabulary words because both of these processes attach an image to an otherwise meaningless sequence of characters. In the first example the image is of the correctly spelled word on the blackboard, in the second, it is the image of the correctly spelled word in the sand.

With regard to password issues, this human behavior reduces the effective password space. The bias that is introduced into the password space is that which is easily remembered by people. For example, let's consider a 6 digit numerical password. This password space algebraically computes to 531,441 different passwords. But, people don't easily remember random numbers. But people are good at remembering dates. Dates can be represented as 6 digit numbers. So people will tend to bias the password space of a 6 digit numerical password to fit the template of calendar dates. What's easier to remember, "386937", or "010104"? The second string is easier to remember, because it's the first day of the New Year! This human bias towards dates dramatically reduces the password space. Just throwing out some numbers, the "date space" is bounded by 12 months and 30 days, 360 unique combinations. Throwing in some combinations for the years portion of the password string, say 20 years, gives only 7200 unique passwords. This compared to 531,441 passwords of random numbers. Lets say the date space is

1. http://www.sans.org/rr/catindex.php?cat_id=6, "Inadequate Password Policies Can Lead to Problems"

bounded by 70 years. This gives 25,200 unique passwords, which is still over 20 times less than the random number set of passwords. The human predilection toward remembering that which has meaning to them significantly reduces the amount of unique passwords that are used in computer systems.

The usability of security measures that are put into place is an important aspect of over all system security. Passwords are hard to remember for many people. This is a reality that needs to be kept in mind. This facet of human nature is as much of a threat vector as is that facet of human nature that wants to deface websites. The fact that people don't easily remember passwords needs to be mitigated, through policy, technical controls, and user education. There are several mnemonic "tricks" that can be used to help people remember passwords. One obvious solution is to eliminate passwords for authentication, and provide a more people friendly solution.

Real life experience supports the human bias contention. "The 2002 NTA Monitor Password Survey found that 84% of computer users consider memorability as the most important attribute in selecting a password and that 81% of users select a common password where possible."[1]. Retired Air Force Gen. Eugene E. Habiger, who was named DOE security "czar" said in a Washington Post article that "Many employees used their last names or initials, and some simply typed "password" when logging onto classified networks..."[2]. It's clear that the password space has been greatly reduced by the human predilection for a personal or substantial meaning to be associated with their passwords. Human behavior negatively affects the effectiveness of passwords in securing computer systems. When performing risk analysis of existing systems using passwords, or designing new systems or applications is it important to take into account that people do not easily remember passwords, long or short. In some cases mitigation can be provided in the form of inducements to use appropriate passwords (Both Windows 2000[3] and Solaris default installs, Figure 5 on page 22, offer technical controls to enforce aspects of password policies). Examples of inducements would be system password policies, and regular password cracking with subsequent follow-up requiring a cracked password be changed to a stronger password.

2.2.4 Password attacks

In the business setting there are 4 attacks most likely to be used to collect passwords. Surprisingly, half of the attacks don't utilize computer technology, rather, these attacks exploit human behavior. The first of these, "social engineering", is a process by which the attacker somehow gains the trust of an insider, and persuades the insider to divulge a password or clues to make guessing a password much easier. The second involves an attacker gaining physical access to the business he wants to attack, and looking for obvi-

1. http://www.nta-monitor.com/Password-survey-press-release_trade_final.doc
2. http://www.washingtonpost.com/ac2/wp-dyn?pagename=article&node=&contentId=A28481-2000Jan25&notFound=true
3. Details in: http://support.microsoft.com/default.aspx?scid=kb;EN-US;161990. How-to in: http://support.microsoft.com/default.aspx?scid=kb;en-us;225230

ous clues to passwords, or to look for passwords written on paper or Post-It notes, perhaps stuck under keyboards or mousepads, or simply stuck to monitors (Smith estimates that as many as 1 in 3 users write their password down and store the information close to the computer[1]). These first two methods can be mitigated by the use of physical security around the perimeter of the business organization, but this only counters the treat vector for outsiders. Insiders looking to gather passwords are already inside the perimeter and have access to written passwords. The third method is to gain access to a computer system, copy the file(s) containing the computer's password information, move this information to the attacker's home site and then guess the passwords contained in the stolen file. The fourth method is to gain access to a site's network, collect network traffic (sniff) with the intent of gathering transmitted passwords. Once some passwords have been gathered, the attacker guesses the passwords.

One Windows password audit and recovery tool, LC4 (the fourth version of L0phtCrack)[2], can read the passwords out of a file from: the local disk, the Windows NT emergency repair disk (not 2000), off remote systems (with privileged access), or sniff NetBIOS traffic on the network and then guess the obtained passwords. John the Ripper is a highly regarded tool for cracking Unix passwords.

Use of passwords can be risky in that 2 major attacks on passwords exploit human behavior, not technical controls. Human behavior is difficult to change, even in a business setting. Policies against writing down passwords can be put in place, and at least will provide evidence of due diligence on the system manager's part, and will certainly deter many "good" people from writing passwords down, if not solely from the educational perspective, then from innate human behavior to conform to local laws and customs. When appropriate technical methods exist that can thwart attack vectors exploiting human behavior, these new methods should be considered for implementation, at least for high value systems.

All modern computer systems store a transformed value of a user's password. This transformation is called a hash. The hash transformation of a password is one way. Once a password has been transformed, it can't be reverse transformed to reveal the original password. These mitigations have been applied early in computing history to the passwording process, to combat attacks against stored plaintext passwords, and to protect the secrecy of a chosen password on the computer (a system admin can't un-hash a users password to discover it). The Unix passwording hashing process has not been found to have flaws. There are two Windows password hashing processes. The two windows hashes are called LANMAN and NTLM. The LANMAN hash is from the older Microsoft LAN Manager network OS, and has two critical flaws which reduces password space. The NTLM hash was introduced in Windows NT 4.0, and is improved over the LANMAN hash[3] (further discussion of password hashing is beyond this paper's scope). All computer based automated password cracking programs guess passwords, trans-

1. Smith, p 161
2. http://www.atstake.com/research/lc/
3. Smith, pp 47 - 58 and 298 - 311

form these passwords using the hash function from the operating system of the attacked computer, then compare these computed hashed values to the hashed values in the attacked computers password file. Cracking passwords or the euphemistic phrase “password recovery”, simply refers to guessing passwords using a computer program specially designed for the purpose. There are two primary approaches to guessing passwords using a cracking program. It’s well known that many people use common words or names in their passwords, so the first approach is to compare hashed dictionary words and common names to the hashed passwords in the password file. This is referred to as a “dictionary attack”. The next approach is to just calculate all the combinations of characters in all possible passwords. This is referred to as a “brute force” attack.

In Section 2.2.1, Equation 4 on page 4 doesn’t just define the password space. More importantly, Equation 4 implies that *all passwords can eventually be guessed*! Passwords are just finite collections of characters. Automated password cracking programs simply create a list of all the possible combinations of characters for passwords, hash them, and then compare each entry in the list to the each of the entries in the password file from a computer system. It is important to stress that there is no “magic” involved with passwords or password cracking, it’s simply creating combinations of characters, and then comparing these to entries in the password file. *All passwords can eventually be guessed*. In the past, passwords were secure because computers could take centuries to guess all the possible combinations of passwords. For instance, Smith relates that in the early 1970’s it would have taken almost 263,000 years to crack all possible Unix passwords using the 95 printing ASCII characters[1]. A modern Pentium III laptop running Windows and LC4 can sustain password cracks of 2 million cracks per second for weak passwords, and around 1 million cracks per second for strong passwords. The immense quantity of cracks per second make a password space of 54 trillion suddenly not seem so large.

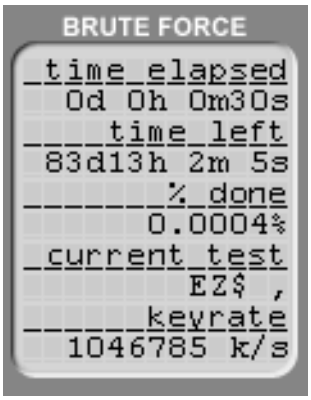
Changing all passwords on a computer system at regular intervals is a well accepted computer security “best practice”. A password change interval is implemented to mitigate the risk of a password being guessed without the knowledge of the administrators of the exploited computer system. If an attacker has a working password, that password would become invalid after a system-wide password change. Once the power of computers became such that passwords could be guessed in minutes to months, and not centuries, the speed at which automated password cracking programs could guess passwords began driving the password change time interval. If all possible passwords for a computer system can be guessed in 90 days, then the password change interval must be less than 90 days to help mitigate risk of system exploit from cracked passwords.

For a simple experiment, 30 randomly generated passwords were loaded into LC4, a highly regarded password cracking tool. The passwords were created using Password Creator Professional (PCP), a very flexible and powerful tool[2]. The passwords generated were special in that a random process was used to “decide” what characters would make

1. Smith, pp. 53
2. http://www.transdig.com/products/pcp/pcp.cfm

up the password. This kind of password is hard to crack, as dictionary attacks aren't effective against passwords that contain no words. The addusers.exe tool from the Windows 2000 Resource kit was also used in the account creation process[1]. The LANMAN hash, the weaker of the 2 hashes used in the Windows world, was used in the test. The laptop used for the experiment was a Pentium-III, 900 MHz, with 348 Mb RAM (MadOnion benchmark CPU:2089, Memory:1225, Hard Drive: 100). LC4 was able to crack 7 4 and 6 character passwords in less than 3 minutes. The remaining 6 character strong passwords were cracked in less than 24 hours. One complex 8 character password was cracked in about 4 days. The remaining passwords were not cracked, after about 6 days of guessing. However, the status monitor in LC4 showed that the entirety of the Windows14 character letters, numbers, and symbol password space would have been guessed by this laptop in about 84 days!

FIGURE 2. LC4 Time to complete crack estimate



The passwords and results from this simple experiment are provided in Table 2 on page 20 and in Figure 3 on page 21. Looking at the table shows that none of these are particularly easy to guess passwords, however short they may be. Several concepts were illustrated with this simple exercise. First, with a limited password space, most notably the 4 character, it is difficult to randomly generate a truly strong password. PCP is configurable such that only upper and lower case, numbers, and symbols are used in the created passwords, but even with multiple runs, it was still difficult to collect enough different, characters to create 5 truly strong 4 character passwords. Also, this experience shows that the concept of password space is useful, and having a small password space truly reduces password strength. The passwords with small password spaces were cracked very quickly, and the passwords with larger passwords spaces took longer to crack, or were not cracked, even after 6 days of guessing attempts. Truly strong passwords do take significant time to crack, at least 6 days. But the it is true that all the passwords, even 14 character ones, would have been cracked in less than 90 days. LC4 has the capability to distribute the brute force cracking task over many machines. The author was able to prepare the distributed crack for 100 different computers. The experiment

1. addusers.exe allows creation of unlimited accounts with "Change password at first logon" set. See http://support.microsoft.com/default.aspx?scid=kb;en-us;199878. Download ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/nt40/i386/addusers_x86.exe

could have been completed in 10 days if the author had access to 8 computers. Clearly, if an attacker wants to crack Windows passwords badly enough, he can do quickly and be confident of a successful exploit.

Given that a modest laptop computer can crack all known windows passwords of 14 characters or less in some cases in days, but in every case, less than 90 days, a security professional might be motivated to slide the password change interval to perhaps, 30 days. Reducing the password change interval is only a sound tactic if there is consequent realization that passwords are in fact unchanged, older technology, and password crack program designers refine their programs consistently. At some point reducing the password change interval will provide diminishing returns solely from the usability standpoint. It's hard for any person to create and remember new strong passwords on 30 day change interval. Efficiency and security may suffer as admins are driven to storing passwords in secure vaults, such as STRIP, or Password Safe[1], which require an additional password login into the vault before the system login can occur (it also should be noted that obtaining the password database from one of these vaults would be a valuable exploit, if the database can be cracked. These two products simply transfer the risk as opposed to mitigating it). However, simply reducing the password change interval is at best an interim solution.

Computers are getting nothing but faster. Intel estimates Moore's law, the doubling of processor speed every 18 months, will continue through this decade[2]. There is little respite in the march for increasing computer processor speed. Quantum computing promises phenomenal increases in computational speed. There are established government funding sources (DARPA, NSF), an on-line scientific journal[3], and university research programs in "Quantum Information Science". In 1998, experts estimated it would take 20 years for a functional quantum computer to be created[4]. A second estimate based on very preliminary data in 2000 thought perhaps 5 years would pass before a functional quantum computer is developed[5]. In any case, any authentication methods planned for implementation on systems with a planned long lifetime should account for a phenomenal increase in the ability to crack passwords, passphrases and cryptographic keys.

A possible argument, or mitigation supporting use of passwords would be that it is possible to harden a computer such that the password file is very difficult to obtain. A similar argument is true for sniffing passwords over the network. Many, if not most networks are switched, which limit effectiveness of sniffing attempts to the local segment. The counter argument to these briefly is this: Many, many networks are hard on the outside, but soft

1. http://www.zetetic.net/products.html and http://www.counterpane.com/passsafe.html and http://sourceforge.net/projects/passwordsafe/
2. http://www.intel.com/research/silicon/mooreslaw.htm (29 June 03)
3. http://www.vjquantuminfo.org/quantuminfo/?jsessionid=2720431057338922023 (29 June 03)
4. http://elib.zib.de/ICM98/TU-Presse/pi184e.htm (29 June 03)
5. http://www.lanl.gov/worldview/news/releases/archive/00-041.shtml (29 June 03)

and chewy on the inside, meaning that in most businesses external firewalls are strong and reliable, but the internal networks are not nearly as tough to exploit. Most business networks do not have multiple internal firewalls[1]. Given that an entire windows password file with very strong passwords can be cracked in as little as 10 days, and since there are few internal controls are in place to stop jumping off or exploits on other machines, and that methods exist to "break" the security offered with switched networks (dsniff, ettercap[2]), can one be comfortable basing a security posture on the fact all computers on the network will be hardened such that the password files will not be retrievable? Certainly it is a valid argument, and the web that must be constructed to create a major exploit is complicated, but since viable alternate authentication methods exist that are reasonable in cost and usability (and in fact probably offer significantly improved usability over strong passwords), the balance tips toward no longer using passwords for authentication.

The material presented in Section 2.0 shows that the use of passwords in the business environment possess negative issues. The algebraic issues that passwords possess are fundamental algebraic properties that cannot be changed, however mitigations exist in that other forms of authentication can be implemented. The computer technology issue boils down to one of choice, if the need to use strong passwords is an over-riding concern, then the system should be required to use the Solaris operating system. The human behavior issue with passwords is able to be mitigated with policies although people's behavior with regard to policies is not completely reliable. Finally, attacks on computer password files are quickly concluded with success, as illustrated by the simple experiment in Section 2.2.4, where all 14 character passwords would have been guessed in less than 84 days by a modestly powerful laptop computer.

3.0 Recommendations

3.1 New perspectives on passwords

Smith raises some interesting points in his discussions of usability and security in various sections of his book.Some of these points have been combined and evaluated against situations the author has experience with. Looking at a modern computer user's workspace in the business setting, the user will probably be found using a computer in a cubicle, or a small office. The computer is likely to be a powerful stand-alone computer, with gigabytes of disk storage, running a Windows, Unix or Unix-like operating system, with applications installed locally. The computer will be networked to a business LAN. The building almost certainly has controlled physical access using some kind of card reader that would disallow entrance to people that aren't part of the business. In the business setting, most users would know many other users first and last names, and would even have access to computer usernames through an employee directory, either hardcopy or freely accessible on the business network. If the business setting resides in one building,

1. http://www.infosecuritymag.com/2003/jun/cover.shtml (29 June 03)
2. http://www.monkey.org/~dugsong/dsniff/ and http://ettercap.sourceforge.net/

it's also likely that the business LAN is wholly owned and managed by the business entity. The LAN may not have an internet connection, but if it does, it is almost certainly firewalled.

In this situation, some normal conventions for password use probably don't create significant benefit from the security perspective. For instance, if everyone using a computer is in a cubicle or office, what reason is there to hide the password being typed at the logon screen? The user would be aware of someone looking over their shoulder since the intruder would be in the computer user's workspace. The computer user could stop typing the password, clear the password, ask the intruder to look the other way, activate a screensaver, or even finish logging in, and at the first private moment, change the password. It seems that failed logons would be reduced if one could see what characters they are typing. Perhaps more importantly, stronger passwords would be easier to use since the user would see what they are typing and know exactly what part of a complicated password was entered incorrectly.

If this business network had no external access, no Internet connection and logging was enabled and reviewed regularly, the password change interval might not need to be set, but only coincide with significant events like receiving a new desktop computer, a major system upgrade, or perhaps even coming back from the Christmas and New Year's holidays. If the network is isolated, all the users know each other, the usernames are public knowledge, and physical access is only allowed for insiders, the only attacks that can come are from insiders, and as infrequent as these would be, they would presumably would be caught during log reviews or other internal security procedures. As security increases around computer systems, thought must be given to the usability of the security measures being put into place.

3.2 Targeted Risk Analysis

We began by stating that passwords are dead. If so, what systems can be used to authenticate users? The vendor market for authentication and directories is rich with offerings. Getting off passwords and onto a new form of authentication is a daunting prospect. If a move from passwords is considered for a system, the first effort in the move can be a targeted risk evaluation of threats specifically against passwords. NIST Special Publication 800-30 can be a helpful guide in this process[1]. The targeted risk analysis should be a written document, stating assumptions and framing the attacks within the three categories of security, Confidentiality, Availability and Integrity. Once these threats are categorized and listed, such as done in 800-30 Section 3, there will be a more clear understanding what properties are needed in the new authentication method to counter the known threats. The strength of a system's authentication method should always be proportional to the value of data, or importance of the function of the system. If a ranking is needed in the threat list, rank by the single or annual expected loss. Ranking in this fashion will show what possible exploits would be most costly, and what can be mitigated or transferred. The severity of an exploit should also be consid-

1. http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf (29 June 03)

ered. 800-30 Section 3.6 gives a good starting point for impact analysis, with 3 categories. Low, corresponding to loss of some assets, or noticeably affect an organization's mission, reputation or interest. Medium corresponding to costly loss, violate harm or impede an organization's mission, reputation or interest, or cause human injury, and High, a highly costly loss, significantly violate harm or impede an organization's mission, reputation or interest, or may result in human death. Given that some systems currently using passwords could be an employee intranet, a central calendar, or other low value systems, the security professional could consider adding 2 more levels to the impact analysis, and numbering them from 1 to 5. The first level would be negligible impact, the second level would correspond to some loss of tangible assets (but no noticeable affect to reputation, mission, or interest), then the three levels as outline in 800-30 following as increasing severity after the new levels. NIST is legislated to provide guidance to the civilian Federal government, so agencies like the National Weather Service, Federal Emergency Management Agency, the Department of Energy and the Nuclear Regulatory commission all have systems that can affect human life and welfare. For a more usual business setting, where the organization doesn't affect human life or welfare these 5 levels could stepped down in severity, or reduced to 3, without significantly affecting the outcome of the analysis. For instance, 5 business systems that could be compromised with a password exploit that would correspond to the 5 levels could be: level 1, Employee Intranet listing softball team scores, and level of charitable contributions by business unit. A level 2 system could list employee benefits, and other closely held but shared information. Level 3 could be policies and procedures and the human resources manual. Level 4, the payroll system, perhaps an exploit giving an insider an un-earned raise or cash disbursement, and level 5 could be the main financial accounting system, allowing transfer of large sums to money to outside accounts, or perhaps a system containing trade secrets or passwords to other systems. The last consideration for the targeted analysis would be taking into account the trust relationships in the network. If system X is compromised, could system Y might then be exploited more easily? Once this targeted risk analysis is complete, the security professional will have an idea of what level of security needs to be implemented for the authentication process, and which method of authentication would work best within the balance of threat, mitigation, acceptance and transference of risk.

3.3 Alternate forms of authentication

The wealth of authentication offerings can be confusing. Add to that the hyperbole of already charged security atmosphere and vendor claims can be positively opaque to logical analysis. An outstanding issue when considering alternate forms of authentication is whether the authentication will be handled within the system, or handled by a directory which then hands off a successful authentication to the system. There a several important considerations with the directory approach that need to be evaluated. Smith covers this aspect in Chapter 11 of his book.

If a current system is low or ineligible value (like the intranet with softball team scores), and authentication is handled by a separate system which requires a strong password, it could be wise to provide an alternate form of authentication to increase system "friendli-

ness" or usability. Can you imagine how popular you would be as a security officer if you made people's passwords easier? When working with authentication for a system always consider the value or importance of the system and scale the strength of the passwords accordingly.

Options for authentication include:

- weak passwords
- strong passwords
- One time passwords (OTP)

  time based OTP- such as RSA[1] (note: RSA and others push the "two-factor" identification concept. Note that the method of authentication is still a time based one time password. However, the OTP has been hybridized with a PIN number. This is a mitigation to counter the threat vector of a stolen OTP generator. Presumably the thief will not know the PIN, and thus will not be able to authenticate.)[2]

  counter based OTP
- RADIUS
- Kerberos - Supported by Windows 2000 (the MS implementation stores the master master key in a cache, which can be dis-allowed if the entire site is Kerberos. This is a risk that should be considered carefully before allowing caching of master keys)
- Challenge/Response - open standard X9.9[3]

  S/Key

  Combined with public key - the server challenge is encrypted by user using his public key. Server decrypts challenge. If challenge received matches challenge sent, user is successfully authenticated[4].

4.0 Conclusion

Passwords come with baggage. The algebraic issues, computer technology and human behavior issues are all inseparable properties, "baggage", that accompany password use. Given the baggage, and that very strong passwords are crackable in less than 90 days, and perhaps even as little as 10 days, and that computers will get dramatically faster in our lifetimes, passwords are dead as a viable alternative for authentication for high value or important systems. System authentication should be reviewed using a risk based methodology which considers system value or functional importance against the expected loss and magnitude of impact of a successful password exploit. Once this risk

1. http://www.rsasecurity.com/products/securid/tokens.html
2. Smith pp. 341 - 368
3. Smith pp. 285 - 293
4. Smith p. 384

analysis has been completed, the system managers can then decide which form of authentication is appropriate form of authentication for their system.

5.0 Appendix

TABLE 1. Combinations of ASCII Printing Characters and Resulting Password Space

Combination of ASCII Printing Characters	Number of Characters in the Combination	Number of characters in a password	Password Space
All Lower Case or All Upper Case	26	2	676
All Lower Case or All Upper Case and Numbers	36	2	1.3 Thousand
Symbols + Punctuation and Numbers	43	2	1.8 Thousand
Upper and Lower Case	52	2	2.7 Thousand
All Lower Case or All Upper Case and Numbers	59	2	3.5 Thousand
Upper and Lower Case and Numbers	62	2	3.8 Thousand
Upper and Lower Case and Numbers and Symbols + Punctuation	95	2	9.0 Thousand
All Lower Case or All Upper Case	26	4	457.0 Thousand
All Lower Case or All Upper Case and Numbers	36	4	1.7 Million
Symbols + Punctuation and Numbers	43	4	3.4 Million
Upper and Lower Case	52	4	7.3 Million
All Lower Case or All Upper Case and Symbols + Punctuation	59	4	12.1 Million
Upper and Lower Case and Numbers	62	4	14.8 Million
Upper and Lower Case and Numbers and Symbols + Punctuation	95	4	81.5 Million
All Lower Case or All Upper Case	26	6	308.9 Million
All Lower Case or All Upper Case and Numbers	36	6	2.2 Billion
Symbols + Punctuation and Numbers	43	6	6.3 Billion
Upper and Lower Case	52	6	19.8 Billion
All Lower Case or All Upper Case and Symbols + Punctuation	59	6	42.2 Billion
Upper and Lower Case and Numbers	62	6	56.8 Billion
All Lower Case or All Upper Case	26	8	208.8 Billion
Upper and Lower Case and Numbers and Symbols + Punctuation	95	6	735.1 Billion
All Lower Case or All Upper Case and Numbers	36	8	2.8 Trillion
Symbols + Punctuation and Numbers	43	8	11.7 Trillion
Upper and Lower Case	52	8	53.5 Trillion

TABLE 1. Combinations of ASCII Printing Characters and Resulting Password Space

Combination of ASCII Printing Characters	Number of Characters in the Combination	Number of characters in a password	Password Space
All Lower Case or All Upper Case	26	10	141.2 Trillion
All Lower Case or All Upper Case and Symbols + Punctuation	59	8	146.8 Trillion
Upper and Lower Case and Numbers	62	8	218.3 Trillion
All Lower Case or All Upper Case and Numbers	36	10	3.7 Quadrillion
Upper and Lower Case and Numbers and Symbols + Punctuation	95	8	6.6 Quadrillion
Symbols + Punctuation and Numbers	43	10	21.6 Quadrillion
All Lower Case or All Upper Case	26	12	95.4 Quadrillion
Upper and Lower Case	52	10	144.6 Quadrillion
All Lower Case or All Upper Case and Symbols + Punctuation	59	10	511.1 Quadrillion
Upper and Lower Case and Numbers	62	10	839.3 Quadrillion
All Lower Case or All Upper Case and Numbers	36	12	4.7 Quintillion
Symbols + Punctuation and Numbers	43	12	40.0 Quintillion
Upper and Lower Case and Numbers and Symbols + Punctuation	95	10	59.9 Quintillion
All Lower Case or All Upper Case	26	14	64.5 Quintillion
Upper and Lower Case	52	12	390.9 Quintillion
All Lower Case or All Upper Case and Symbols + Punctuation	59	12	1.8 Sextillion
Upper and Lower Case and Numbers	62	12	3.2 Sextillion
All Lower Case or All Upper Case and Numbers	36	14	6.1 Sextillion
Symbols + Punctuation and Numbers	43	14	73.9 Sextillion
Upper and Lower Case and Numbers and Symbols + Punctuation	95	12	540.4 Sextillion
Upper and Lower Case	52	14	1.1 Septillion
All Lower Case or All Upper Case and Symbols + Punctuation	59	14	6.2 Septillion
Upper and Lower Case and Numbers	62	14	12.4 Septillion
Upper and Lower Case and Numbers and Symbols + Punctuation	95	14	4,876.8 Septillion

TABLE 2. Password cracking results and account data used in simple experiment

Sample Number	Account Name	Number of Characters	Password	Time to Crack
1	acct1	4	WAv4	0d 0h 1m 44s
2	acct2	4	h2TU	0d 0h 1m 40s
3	acct3	4	3724	0d 0h 1m 44s
4	acct4	4	15ul	0d 0h 1m 39s
5	acct5	4	G6_W	0d 0h 1m 41s
6	acct6	6	Q$MzT{	0d 23h 36m 26s
7	acct7	6	cL9ge!	0d 15h 13m 9s
8	acct8	6	-f$2Ms	0d 3h 19m 17s
9	acct9	6	21'niw	0d 7h 39m 9s
10	acct10	6	-6nC3$	0d 16h 11m 26s
11	acct11	8	Bnv`'SiW	crack stopped
12	acct12	8	{eW98"h5	crack stopped
13	acct13	8	MYs^Ra1n	crack stopped
14	acct14	8	OtmP_(`q	crack stopped
15	acct15	8	uP_"2H0X	crack stopped
16	acct16	10	8P0b_%Xk's	crack stopped
17	acct17	10	`3qm9A7I$D	crack stopped
18	acct18	10	^}$4tCUKe)	crack stopped
19	acct19	10	-V9opj&{}1	crack stopped
20	acct20	10	R43LEgny%)	3d 17h 51m 53s
21	acct21	12	7Jmc8)s1Y'4u	crack stopped
22	acct22	12	^K_"af-%wY(j	crack stopped
23	acct23	12	X"T5`2H6vLPA	crack stopped
24	acct24	12	3w9&t$l^%)6h	crack stopped
25	acct25	12	Dx!4Zkbu12{y	crack stopped
26	acct26	14	%m`q8X2JdHSi@z	crack stopped
27	acct27	14	m!_Wo{}"b69$)d	crack stopped
28	acct28	14	l7$@&8b%4dJFA`	crack stopped
29	acct29	14	T25f6z&SL'^`RO	crack stopped
30	acct30	14	-U2`%16Fj3nHTX	crack stopped

FIGURE 3. Password cracking results screenshot

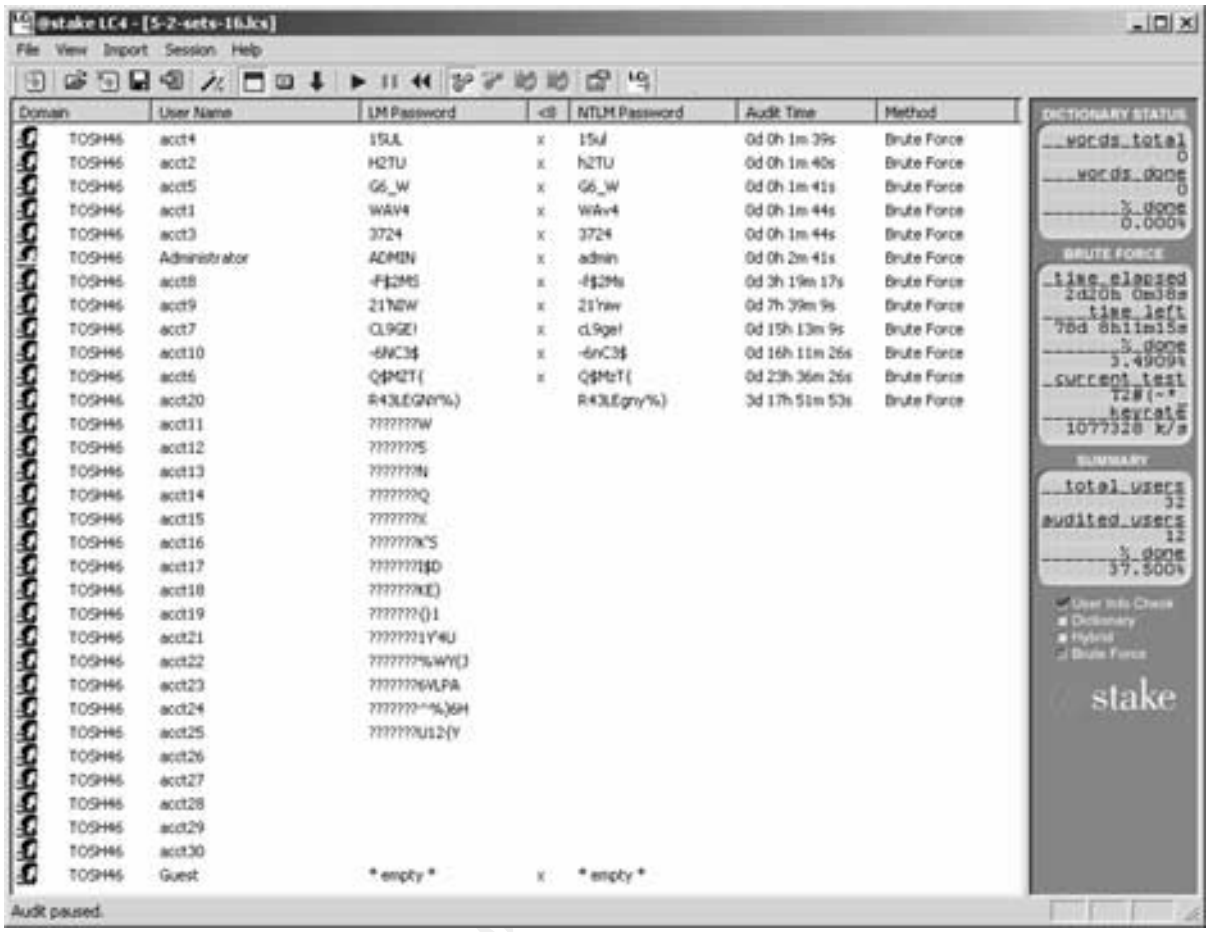


TABLE 3. Windows Allowed Symbols for Passwords[a]

‘	~	!	@	#	$	%	^
&	*	(	)	_	+	-	=
{	}	|	[	]	\	:	“
;	‘	<	>	?	,	.	/

a. Space is allowed, but not shown (it would be a blank square!).

FIGURE 4. Windows Allowed Symbols for Account Names

FIGURE 5. Solaris default install password restrictions

```
raven.foobard.net: fcheswick: 3 /export/home/fcheswick> passwd
passwd: Changing password for fcheswick
Enter existing login password:
New Password: 2a!b#5d
passwd: Password too short - must be at least 8 characters.
Please try again
New Password: 12345678
passwd:
The first 8 characters of the password
must contain at least two alphabetic characters and at least
one numeric or special character.
Please try again
New Password: 1a234567
passwd:
The first 8 characters of the password
must contain at least two alphabetic characters and at least
one numeric or special character.
Permission denied
```

TABLE 4. ASCII Printing Characters

Number	Number of Characters in Group	Character	Description	Octal	Dec	Hex
1	33 symbols	SP	Space	40	32	20
2	33	!	Exclamation mark	41	33	21
3	33	"	Quotation mark (" in HTML)	42	34	22
4	33	#	Cross hatch (number sign)	43	35	23
5	33	$	Dollar sign	44	36	24
6	33	%	Percent sign	45	37	25
7	33	&	Ampersand	46	38	26
8	33	`	Closing single quote (apostrophe)	47	39	27
9	33	(	Opening parentheses	50	40	28
10	33	)	Closing parentheses	51	41	29
11	33	*	Asterisk/star/multiply	52	42	2a
12	33	+	Plus	53	43	2b
13	33	,	Comma	54	44	2c
14	33	-	Hyphen/dash/minus	55	45	2d
15	33	.	Period	56	46	2e
16	33	/	Slant/forward slash/divide	57	47	2f
17	33	:	Colon	72	58	3a
18	33	;	Semicolon	73	59	3b
19	33	<	Less than sign (< in HTML)	74	60	3c
20	33	=	Equals sign	75	61	3d
21	33	>	Greater than sign (> in HTML)	76	62	3e
22	33	?	Question mark	77	63	3f
23	33	@	At-sign	100	64	40
24	33	[	Opening square bracket	133	91	5b
25	33	\	Reverse slant (Backslash)	134	92	5c
26	33	]	Closing square bracket	135	93	5d
27	33	^	Caret (Circumflex)	136	94	5e
28	33	_	Underscore	137	95	5f
29	33	`	Opening single quote	140	96	60
30	33	{	Opening curly brace	173	123	7b
31	33	|	Vertical line	174	124	7c
32	33	}	Cloing curly brace	175	125	7d
33	33	~	Tilde (approximate)	176	126	7e
34	26 upper-case letters	A	Uppercase A	101	65	41
35	26	B	Uppercase B	102	66	42

TABLE 4. ASCII Printing Characters

Number	Number of Characters in Group	Character	Description	Octal	Dec	Hex
36	26	C	Uppercase C	103	67	43
37	26	D	Uppercase D	104	68	44
38	26	E	Uppercase E	105	69	45
39	26	F	Uppercase F	106	70	46
40	26	G	Uppercase G	107	71	47
41	26	H	Uppercase H	110	72	48
42	26	I	Uppercase I	111	73	49
43	26	J	Uppercase J	112	74	4a
44	26	K	Uppercase K	113	75	4b
45	26	L	Uppercase L	114	76	4c
46	26	M	Uppercase M	115	77	4d
47	26	N	Uppercase N	116	78	4e
48	26	O	Uppercase O	117	79	4f
49	26	P	Uppercase P	120	80	50
50	26	Q	Uppercase Q	121	81	51
51	26	R	Uppercase R	122	82	52
52	26	S	Uppercase S	123	83	53
53	26	T	Uppercase T	124	84	54
54	26	U	Uppercase U	125	85	55
55	26	V	Uppercase V	126	86	56
56	26	W	Uppercase W	127	87	57
57	26	X	Uppercase X	130	88	58
58	26	Y	Uppercase Y	131	89	59
59	26	Z	Uppercase Z	132	90	5a
60	26 lower case letters	a	Lowercase a	141	97	61
61	26	b	Lowercase b	142	98	62
62	26	c	Lowercase c	143	99	63
63	26	d	Lowercase d	144	100	64
64	26	e	Lowercase e	145	101	65
65	26	f	Lowercase f	146	102	66
66	26	g	Lowercase g	147	103	67
67	26	h	Lowercase h	150	104	68
68	26	i	Lowercase i	151	105	69
69	26	j	Lowercase j	152	106	6a
70	26	k	Lowercase k	153	107	6b
71	26	l	Lowercase l	154	108	6c

TABLE 4. ASCII Printing Characters

Number	Number of Characters in Group	Character	Description	Octal	Dec	Hex
72	26	m	Lowercase m	155	109	6d
73	26	n	Lowercase n	156	110	6e
74	26	o	Lowercase o	157	111	6f
75	26	p	Lowercase p	160	112	70
76	26	q	Lowercase q	161	113	71
77	26	r	Lowercase r	162	114	72
78	26	s	Lowercase s	163	115	73
79	26	t	Lowercase t	164	116	74
80	26	u	Lowercase u	165	117	75
81	26	v	Lowercase v	166	118	76
82	26	w	Lowercase w	167	119	77
83	26	x	Lowercase x	170	120	78
84	26	y	Lowercase y	171	121	79
85	26	z	Lowercase z	172	122	7a
86	10 numbers	0	Zero	60	48	30
87	10	1	One	61	49	31
88	10	2	Two	62	50	32
89	10	3	Three	63	51	33
90	10	4	Four	64	52	34
91	10	5	Five	65	53	35
92	10	6	Six	66	54	36
93	10	7	Seven	67	55	37
94	10	8	Eight	70	56	38
95	10	9	Nine	71	57	39
96	1 key on the keyboard	DEL	Delete/rubout/cross-hatch box	177	127	7f

TABLE 5. Control-Key Combinations (Non-Printing ASCII Characters)

Control-Key Combination	Control Action	Octal	Decimal	Hex	
^@	Null character	NUL	0	0	
^A	"Start of heading = console interrupt"	SOH	1	1	
^B	"Start of text maintenance mode on HP console"	STX	2	2	
^C	End of text	ETX	3	3	
^D	"End of transmission not the same as ETB"	EOT	4	4	
^E	"Enquiry goes with ACK; old HP flow control"	ENQ	5	5	
^F	"Acknowledge clears ENQ logon hang"	ACK	6	6	
^G	"Bell rings the bell"	BEL	7	7	
^H	"Backspace works on HP terminals/ computers"	BS	10	8	
^I	"Horizontal tab move to next tab stop"	HT	11	9	
^J	Line Feed	LF	12	10	
^K	Vertical tab	VT	13	11	
^L	"Form Feed page eject"	FF	14	12	
^M	Carriage Return	CR	15	13	
^N	"Shift Out alternate character set"	SO	16	14	
^O	"Shift In resume default character set"	SI	17	15	
^P	Data link escape	DLE	20	16	
^Q	"XON with XOFF to pause listings; "okay to send""	DC1	21	17	
^R	"Device control 2 block-mode flow control"	DC2	22	18	
^S	"XOFF with XON is TERM=18 flow control"	DC3	23	19	

TABLE 5. Control-Key Combinations (Non-Printing ASCII Characters)

Control-Key Combination	Control Action	Octal	Decimal	Hex	
^T	Device control 4	DC4	24	20	
^U	Negative acknowledge	NAK	25	21	
^V	Synchronous idle	SYN	26	22	
^W	"End transmission block	not the same as EOT"	ETB	27	23
^X	"Cancel line	MPE echoes !!!"	CAN	30	24
^Y	"End of medium	Control-Y interrupt"	EM	31	25
^Z	Substitute	SUB	32	26	
^[	"Escape	next character is not echoed"	ESC	33	27
^\	File separator	FS	34	28	
^]	Group separator	GS	35	29	
^^	"Record separator	block-mode terminator"	RS	36	30
^_	Unit separator	US	37	31	

6.0 References

"password". "Merriam-Webster Online Dictionary". 2003. http://www.m-w.com/home.htm, (7 March).

Smith, Richard E. Authentication: From Passwords to Public Keys, Boston: Addison-Wesley, 2002.

Skoudis, Ed. Counter Hack. Upper Saddle River: Prentice-Hall, 2002. 279.

MasterLock Corporation. “Master Lock - Travel”. http://www.masterlock.com/cgi-bin/style_search.pl?style_id=A7&sub_style_id=B74, (29 June 2003).

NTA Corporation. “Password-survey-press-release_trade_final.doc”. http://www.nta-monitor.com/Password-survey-press-release_trade_final.doc, (29 June 2003).

Robelle Solutions Technology Inc. “ASCII character set”. http://www.robelle.com/smugbook/ascii.html, (29 Jun 2003).

Hermens, Leonard. “Inadequate Password Policies Can Lead To Problems”, 10 October 2001. http://www.sans.org/rr/catindex.php?cat_id=6 (29 Jun 2003).

Microsoft Corporation. “Microsoft Windows 2000 Documentation”. http://www.microsoft.com/windows2000/en/professional/help/default.asp?url=/windows2000/en/professional/help/windows_password_tips.htm (29 Junl 2003).

Microsoft Corporation. “Microsoft TechNet”. http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/winxppro/proddocs/windows_password_tips.asp, (29 Jun 2003).

Sun Microsystems, Inc. “Solaris Advanced Users Guide”. http://docs.sun.com/db/doc/802-5826/6i9iclf5n?a=view, (29 Jun 2003).

Loeb, Vernon. “Energy Chief Touts Nuclear Lab Security Upgrades”. Page A13. 26 January 2000. http://www.washingtonpost.com/ac2/wp-dyn?pagename=article&node=&contentId=A28481-2000Jan25&notFound=true (29 Jun 03).

Futuremark Corporation. “Benchmarks - PCMark 2002”. http://www.futuremark.com/products/pcmark2002/ (29 Jun 2003).

Jennings, Tom. “Texts: Annotated History of character codes”. 5 Dec 2001. http://www.wps.com/projects/codes/index.html#TOP (29 Jun 2003).

Czyborra, Roman. “ISO 646 (Good old ASCII)”. http://czyborra.com/ (29 Jun 03).

PBS. “frontline: hackers: who are hackers: notable hacks”. http://www.pbs.org/wgbh/pages/frontline/shows/hackers/whoare/notable.html. (29 July 2003).

Microsoft Corporation. “How to Enable Strong Password Security in Windows NT”. KB161990. 11 Jun 2002. http://support.microsoft.com/default.aspx?scid=kb;EN-US;161990 (29 Jun 03).

“randomness”. “Merriam-Webster Online Dictionary”. 2003. http://www.m-w.com/home.htm, (7 March).

Microsoft Corporation. “Creating Strong Passwords”. Microsoft Windows 2000 Documentation. http://www.microsoft.com/windows2000/en/professional/help/default.asp?url=/windows2000/en/professional/help/windows_password_tips.htm (29 Junl 2003).

SANS Institute. “SANS/FBI The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts’ Consensus”. Version 3.23. 29 May 2003. http://www.sans.org/top20/ (29 Jun 03).

SANS Institute. “SANS/FBI The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts’ Consensus”. Version 3.23. 29 May 2003. http://www.sans.org/top20/#index (29 Jun 03).

Microsoft Corporation. “Enabling Strong Password Functionality in Windows 2000”. KB225230. 28 May 2003. http://support.microsoft.com/default.aspx?scid=kb;en-us;225230 (29 Jun 03).

@stake Corporation. “LC4”. http://www.atstake.com/research/lc/ (29 Jun 2003).

Transdigital Solutions 2003. “Password Creator Professional”. http://www.transdig.com/products/pcp/pcp.cfm (29 Jun 2003).

Microsoft Corporation. “AddUsers Automates Creation of a Large Number of Users”. KB199878. 14 May 2003. http://support.microsoft.com/default.aspx?scid=kb;en-us;199878 (29 Jun 03).

Microsoft Corporation. “AddUsers.EXE”.
ftp://ftp.microsoft.com/bussys/winnt/winnt-public/reskit/nt40/i386/addusers_x86.exe (29 Jun 03).

Zetetic Enterprises. “Strip v1.0 (Secure Tool for Recalling Important Passwords)”
http://www.zetetic.net/products.html (29 Jun 2003).

Counterpane Labs. “Password Safe”.
http://www.counterpane.com/passsafe.html (29 Jun 2003).

SourceForge.net. “Password Safe”.
http://sourceforge.net/projects/passwordsafe/ (29 Jun 2003).

Intel Corporation. “Moore’s Law”. Intel Research - Silicon - Moore’s Law.
http://www.intel.com/research/silicon/mooreslaw.htm (29 Jun 2003).

Virtual Journals. “Virtual Journal of Quantum Information”.
http://www.vjquantuminfo.org/quantuminfo/?jsessionid=2720431057338922023 (29 Jun 2003).

Schmidt, Vasco. “New generation of computers can break secret codes - Nevanlinna Prize winner Peter Shor has proved that factorising large numbers is possible at ultrafast speeds using quantum computers”. Technische Universität Berlin. 18 Aug 1998.
http://elib.zib.de/ICM98/TU-Presse/pi184e.htm (29 Jun 2003).

Hanson, Todd. “Los Alamos scientists make seven bit quantum leap”. 24-Jan-2003.
http://www.lanl.gov/worldview/news/releases/archive/00-041.shtml (29 Jun 2003).

Snyder, Joel. “Turning the Network Inside Out” Information Security. June 2003.
http://www.infosecuritymag.com/2003/jun/cover.shtml (29 Jun 2003).

Song, Dug. “dsniff”. http://www.monkey.org/~dugsong/dsniff/ (29 Jun 2003).

SourceForge.net. “ettercap”.
http://ettercap.sourceforge.net/ (29 Jun 2003).

Stoneburner, Gary , Goguen, Alice, Feringa, Alexis. “Risk Management Guide for Information Technology Systems - Recommendations of the National Institute of Standards and Technology” Washington: U.S. GOVERNMENT PRINTING OFFICE. October 2001. pp. 14 - 23
http://csrc.nist.gov/publications/nistpubs/800-30/sp800-30.pdf (29 Jun 2003).

RSA Security. “RSA SecurID Tokens”.
http://www.rsasecurity.com/products/securid/tokens.html (29 Jun 2003).