Evaluating Windows Vista user account security

Alaa M. Abdul-Hadi* ***Ammar A. Abbas****

Received 29, December, 2009
Accepted 13, June, 2010

Abstract:

In the current Windows version (Vista), as in all previous versions, creating a user account without setting a password is possible. For a personal PC this might be without too much risk, although it is not recommended, even by Microsoft itself. However, for business computers it is necessary to restrict access to the computers, starting with defining a different password for every user account. For the earlier versions of Windows, a lot of resources can be found giving advice how to construct passwords of user accounts. In some extent they contain remarks concerning the suitability of their solution for Windows Vista. But all these resources are not very precise about what kind of passwords the user must use. To assess the protection of passwords, it is very useful to know how effective the widely available applications for cracking passwords. This research analyzes, in which way an attacker is able to obtain the password of a Windows Vista PC. During this research the physical access to the PC is needed. This research shows that password consists of 8 characters with small letter characters and numbers can easily be cracked if it has know usual combinations. Whereas a Dictionary Attack will probably not find unusual combinations. Adding captel letter characters will make the process harder as there are several more combinations, so it will take longer time but is still feasible. Taking into account special characters it will probably take too long time and even most Dictionary Attacks will fail. For rainbow tables the size of the table has to be considered. If it is not too big, even these small passwords cannot be cracked. For longer passwords probably the simplest ones, small letter characters and numbers, can be cracked only. In this case brute force takes too long time in most cases and a dictionary will contain only a few words this long and even the rainbow tables become too large for normal use. They can only be successful if enough limitations are known and the overall size of the table can be limited.

Key words: Operating System, System Security, Authentication System, Trusted Computing, Windows Vista, Memory Protection.

Introduction:

Several protections needs to be established to fully restrict access to a PC, either physical or by software, defining a password are the most essential one. It can easily be compared to a key for a door. If a person has the key, he can enter. If he does not have it, he needs to find another way in, which will necessitate more effort. Sometimes, an attacker does not achieve his aim, taking the other way, For example when an attacker wants to impersonate another person. For a malicious person there can be numerous reasons why he wants to gain direct access to a PC and this is by getting the password.

The analysis was distributed to several steps. The first step was to answer the question"Is the attacker able to get the files where the passwords are stored". This is a determining factor, if the

*Computer Center, University of Baghdad, Baghdad, Iraq

attacker is able to obtain and crack the passwords or just to guess using adequate tools. As guessing is rather slow and useful only if the attacker has no possibility to obtain the encrypted password [1], this research focused on examining the methods for cracking passwords and the methods' effectiveness. On one hand the speed and on the other hand the kind of passwords they are able to reveal in adequate time.

The other task for this research is to examine the possibility to get into Windows to replace or delete the password file. Of course, If the file is replaced the system should recognize this and if it is deleted no access should be possible, otherwise the setting of passwords is useless. Furthermore both operations should leave marks, like files created by an unknown user.

The methods used and examined are Brute Force Attacks, Dictionary Attacks and Rainbow table Attacks.

A Brute Force Attack is the most simple and exhaustive one and will ever lead to a success. The reason is that the attacker encrypts every possible combination of allowed characters. Eventually, the attacker will find the combination matching the encrypted password [2]. The time for the worst case is dependent on the processor speed and an efficient program style. As the processors become perpetually faster, by Moore's Law doubling every 1.5 years. Moore himself recently stated this "an attacker is able to search more and more and thus longer passwords can be cracked in the same amount of time" [2].

Dictionary Attacks take a different approach. Most people do not use random passwords as they cannot remember them. Therefore they use names of relatives, puppets, places and so on. Only some expert users change letters with characters like"S" and"$" or"a" and"@". The Attacker either creates a list with all commonly used words or downloads it somewhere from the Internet. A lot of "normal" dictionaries for text processing applications can be downloaded from the Internet. Then the attacker lets the PC create the alternative notations and possible combinations with numbers. When attacking a password the attacker uses this list. There is a high likelihood that the password will be within this list and the more the attacker knows about the person, the lesser words he needs in the list. The list contains far less combinations than there are. Hence, this approach is much faster, especially for longer words, but holds the risk of missing unusual passwords out [2].

To speed up the first two attacks it is possible to calculate the encrypted passwords and store them together with the plain text in a list. When the attacker gets the encrypted password, he just needs to look up the corresponding password in the list. For a dictionary this may be possible, but for all possibilities of a Brute Force Attack this list becomes very huge. Assuming the ASCII character set is used, where one character is one Byte (8 bits). Furthermore assuming that a password is taken as recommended by Microsoft with a size of 6 characters [1], if we assume that the password contains one uppercase letter and one non-alphabetic character and the rest can be anything, then the possibilities for the above password(6=1(uppercase)+1(special character+4(any character). Each password may be consist of:

52 letters (26 uppercase and 26 lowercase) + 10 numbers (0-9) + approximately 38 special characters = 100 usable characters, The possibilities for the above password is:

1(uppercase)=26*($^{6}_{1}$)

1(special character=48*($^{5}_{1}$)

Where :

$(^{n}_{j})$=n!/j!(n-j) ! {the number of combinations of n objects taken j at a time}

4(any character)=100^4

BThe sum of the possibilities is :

100^4 *(26*($^{6}_{1}$))*(48*($^{5}_{1}$))=3744000000000

By multiplying the total number of bits (8) used for storing each character of the password by the number of characters for the password (6) this yields 48 bits used for each password, and then there are:

For each plain text hash value combination:

6 Bytes plain text + 16 Bytes hash value = 22 Bytes

The storage space is:

22*3744000000000=82368000000000 Bytes

To understand the above number we divide it by one Giga Byte (1073741824)

82368000000000/1073741824=76.711 GB of data.

For each additional character the needed space increases by a factor of roughly 97. This shows that the method is faster but not very practical. Even though the current maximum size of a single hard disk is nearly one terra-byte (1024^4), around 80 hard disks are needed to store just the list. Thinking of a list for 7 character passwords, more than 7.000 of such hard disks would already be needed. For a single person that is unthinkable to posses, for a big company or a government agency, it may be possible but is rather unlikely.

A slightly different approach, known today as Rainbow table's method, was introduced by Philippe Oechslin at the Crypto 2003 [3]. His idea was to reduce the needed space by arranging the password/hash-combinations in an array of chains with a certain length and to store only the first and the last password of this chain. To do so, an appropriate reduction function is needed, that reduces the hash value of a password to derive another password, thus creating a chain of passwords and hashes. When cracking a password, the hash value is reduced and compared with the list of last passwords. If it matches a password,

Then the password the attacker is looking for is in this chain and all that is needed to be done is to recalculate the chain starting with the first password until the matching hash value is found. If no end password matches, then the password is hashed and reduced to a new password until either a matching password is found or there is no match possible, because of the attributes of the table. The speed of cracking the passwords depends on the implementation but it is really fast compared to other methods. The reduction function is very important in this method. It has to create reliable chains and it may not happen that a password is created already used in this or any of the other chains. This would produce a loop with an objectionable outcome (e.g. no equal end passwords for two or more chains) (4).

Two PCs were used with an Intel Core 2 Duo 3 GHz processor, 3 GB RAM and a 230GB hard disk. The Systems were running a new installation of Windows Vista. The processor speed and the amount of RAM are important, regarding the amount of computations that can be done within limited time. The hard disk space is important for storage intensive methods.

To retrieve the password file from the system, a Linux boot CD was used and a USB-Stick to store file on an independent media.

Different applications are available to implement the methods described above. Some of them are

specialized for a certain method; others implement all of the named methods. The applications chosen are Cain, Proactive Password Auditor and John the Ripper. Cain is a freeware password recovery tool for different kinds of passwords and supports all three methods. It even includes a sniffing module, with the possibility to directly crack passwords found in the sniffed packets and the program"Winrtgen" to create rainbow tables. Given the encrypted password file Cain is able to retrieve the hash values itself. Then the different methods can be applied to each hash value separately or to all together.

The second program implementing all methods is Proactive Password Auditor. This product is specialized in Windows passwords.

A special program for Dictionary Attacks is "John the Ripper". It is a command line application and offers a range of options to choose from when cracking passwords, amongst others the encryption format and the wordlist to use. The password hashes need to be available in a text file. Therefore, another application is needed preparing the password file and creating the list of hash values.

In order to carry out the Dictionary Attack this research uses the"wordlist.txt" dictionary.

Materials and Methods:

In order to carry out the examinations some preparations had to be made. A set of user accounts with passwords of different complexity had to be created. Table 1 gives an overview about these passwords. There are two major groups of passwords, within which the complexity is from simple to complex. The first seven passwords were only eight characters long, which is the normal size used in password policies at the moment. The second group consisted of seven passwords having the same complexity as the first but were 14 characters long.

Table (1): Suggested Passwords and user accounts

	Name	Password
1	John Doe	12345678
2	Anton	Abcd1234
3	Bridget	ABcd1234
4	Clara	Abca1230
5	Daniel	AbcA1230
6	Emei	Abc123#?
7	Flora	AbC123+)
8	John Doe 2	12345678901234
9	Anton2	abcdefg1234567
10	Bridget2	ABCDefg1234567
11	Clara2	Abcdea12345ous
12	Daniel2	ABcdeA12345Ous
13	Emei2	Abcde12345#?!/
14	Flora2	ABcde1235+)!/

The hash values of these passwords were the input for the different tools and methods. Through this it is easily illustrated what kind of passwords they are able to crack and to compare the results.

1. Getting the Passwords:

The first task is to find the location of the password file and access it. The file can be found in the "c:/windows/system32/config" folder and is called "SAM"[2]. Another file is needed in order to access the passwords, which is the "SYSTEM" file. It contains the system key with which the passwords are encrypted.

The second task is to copy the files. This was not possible because Windows did not allow accessing it in any way. Even a reboot in "Safe mode" did not help.

Therefore the Boot-CD is used and Linux is booted from the CD. After logging the access to the folder is made. The"C" partition was mounted to "sda2", so the path was "mnt/sda2/windows/system32/config". Both files were copied to a USB stick. Because Windows was not booted, this

file was no longer secured by the operating system.

2.Deleting or Replacing the SAM file:

The second task was to test if it is possible to get access to the System by deleting or replacing the SAM file.

As observed in first task the PC has to be booted with a different system to access the SAM file. Therefore the Boot-CD is used to boot Linux. Because the NTFS partition was mounted as read-only it had to be unmounted (”umount /mnt/sda2”) and then mounted again with write enabled (”mount /dev/sdax /mnt/sdax -t ntfs-3g”). The option”-t ntfs-3g” tells the mount-command to use the ntfs-3g driver that can access NTFS partitions with write enable. Now changing to the folder”/mnt/sda2/windows/system32/config” and deleting the SAM file with the command ”del SAM”. Then the Boot-CD is removed and reboot the PC to examine the changes. Windows was not able to boot. The screen to select the last working configuration appeared, but none could be booted.

Next examination is to see if the system would boot with a different SAM file. Therefore, other PC in the laboratory is booted using the Boot-CD. Then a copy of the SAM file of this PC on the USB stick is made. Then the first PC is booted with the Boot-CD, mounted the partition with write enable and copying the SAM file of the other PC into the”config” folder. When trying to reboot the PC, the screen to select the last working configuration appeared.

Both tests showed that it is not possible to simply delete the file or to replace it by a pre-configured SAM file. To carry out the following examinations the SAM file is replaced by the original one and everything worked fine again.

3. Brute Force Attack:

To examine the Brute Force Attack the programs Proactive Password Auditor (referred to as PPA) and Cain was used. In both cases the prepared passwords of the SAM file that was extracted in section 2.1 was used. As a Brute Force Attack can be carried out with different configurations, two tests were made. The first for passwords with all characters possible but with a minimum length of 8 and a maximum length of 14. This would represent an attacker with little knowledge only. For the second test lowercase characters and numbers with a length of 8 were used. This would represent an attacker that already has some knowledge or anticipation about the password.

First the capabilities of PPA were tested. The passwords were loaded. This is done byselecting”RegistryData(SAM,System)” on the ”Hashes” tabulator. When pressing the “DUMP” button a pop up window appeared where the path to the SYSTEM and the SAM file need to be selected.

After doing so,”DUMP” is pressed in the pop up Window and PPA conducted a simple Brute Force, it takes about 5 to 10 seconds. Only the simplest password Of John Doe (12345678) was used. In order to carry out the first test the ”Brute-Force-Attack” tabulator was used and ”NT-Attack(a selection that configures the program to target the Microsoft Authentication Protocol(NTLM))” was used, alphabet characters(upper-/lowercase), numbers and special characters and set length to 8 to 14. No passwords were cracked using PPA. The Cain is the second application that is tested. After starting Cain the first step was to change to the tabulator ”Cracker” as the application offers a lot of functionalities. In order to load the passwords”File/Add to list” has to

be chosen from the menu. A pop up opens where"Import Hashes from a SAM database" and then the path to the SAM file needs to be selected.

The next step was to carry out the first test. The rows containing the test users are used. From the context menu the"Brute Force Attack-/NTLM Hashes" is selected.

In the opening window the key size to the defined range (8-14) is defined. The first test takes about 4 hours and it was not able to crack any password. For the second test the configuration to 8 characters lower case characters and numbers is set, and after pressing "Start", for the first test the result was that: the test would have lasted more than40.000.000.000.000 years(calculated by the program). In case of the second test it takes only 12 days to run through all possibilities. The second test was able to find the passwords for Anton and Bridget[5].

4. Dictionary Attack:

In order to carry out a Dictionary Attack, as the name says, a dictionary file is needed. To have comparable results the"wordlist.txt" file is used that comes with the tool OPHCrack.

In Cain the procedure is nearly the same as for the Brute Force Attack. First the passwords were loaded. Then the passwords that should be cracked were selected and "Dictionary Attack/NT Hashes" was selected from context menu. In the following window the dictionary file was selected. Cain offers some options, from which all but "Case perms" are selected. For the first test the options was not changed and the "Start" is pressed to start the attack. For the second test the "Case perms" is chosen, automatically "Upper Case" was deselected. This option makes Cain try upper and lower case variations of each listed word.

The first test took nearly 30 seconds and revealed the first two passwords of John Doe and Anton. The second test took about 5 minutes and additionally revealed the password of Bridget. Overall it took 5 minutes to reveal 3 passwords.

For the second Dictionary Attack test John the Ripper was used. But instead of Cain, John the Ripper is not able to obtain the hash values itself. To prepare the hash values the"pwdump5" was used. The command"pwdump5.exe -f SAM-k SYSTEMpwhash.txt" extracted the user names and hash values from the SAM files and wrote it to the file"pwhash.txt". Afterward the output file needs to be cleaned from the system ids plus the leading double points. To start the attack the command"john-mmx386.exe -wordlist=Wordlist.txt pwhash.txt" was executed. This made"John the Ripper" execute the attack using the given dictionary.

The output of the program was"Loaded 34 password hashes with no different salts (NT LM DES [32/32 BS]) guesses: 0 time: 0:00:00:00 100% c/s: 8859K
Trying: ZYRIAN - ZYZZOGE" showed that John the Ripper was not able to crack any of the passwords. Even to enable the option "-ru" did not help.
Altogether Cain was able to crack 3 passwords when the case permutation option was enabled, however John the Ripper failed to crack any password.

5. Creating Rainbow Tables:

Before attacking the passwords with a Rainbow table Attack such a table either needs to be created manually or downloaded from the Internet. The Cain application can be accessed via the Windows start menu and resides in the folder of Cain after installation. In order to create a new

table “Add Table” was pressed in the window that opened. Another window popped up where the location of the data that has to be entered to create the table(s).

First the minimum of requirements is used, these were”ntlm” passwords with a minimum and maximum length of 8, index 0, Chain length 10000, Chain count 40.000.000, number of tables 1 and character set ”loweralpha-numeric”.

In an information region beneath the input fields the application calculated the key space (2,8 Trillion), the needed disc space (610 MB) and the success probability (12%) of this table. In this small table only a small part of all passwords would be included. Therefore a larger table that comes with Cain program is used by pressing the button ”Benchmark” on the bottom of the window. After short time further information was presented telling that it would last 3 and half days to create this one table. As twelve percent is much to low, if the number of tables is increased to 22, then the probability got over the 95%. But on the other hand it would have taken a quarter of a year to calculate all these tables. The maximum time it would have taken to search these tables is 15 minutes, if there are 22 tables and 30 seconds, there is only one table.

As the time to create the tables was much too long the tables used in this research is a group of tables for lower alpha characters and numeric, with a length of 1 to 8 characters and a success probability of 98%. With a size of about 9,8 GB.

6. Rainbow table Attack:

The attack with rainbow tables could be examined with Cain only, because of the problem with PPA described in section 2.3. Table in section 2.5 is used, for lower case characters and numbers.

In order to carry out the test the passwords have to be loaded again to have no revealed passwords, as had been done in section 2.3 and 2.4. After selecting all 14 passwords”Cryptanalysis Attack/NTLM Hashes/via RainbowTables (Rainbow crack)” is selected from the context menu. In the following window the rainbow tables that is used in the attack is selected. At last the attack is started by pressing ”Start”. The overall attack runs about 10 minutes. In the monitor region the test with each rainbow table file is shown. The first password from John Doe is found in file 1 after 300 seconds, the second password from Anton is found in file 4 after 12.27 seconds.

Results and Discussion:

From section 2.1 and 2.2 the conclusion that can be drawn is that: in the currently used normal configuration the user accounts do not prohibit access to the system. Using a Boot-CD to boot the PC Windows’ security mechanisms is not activated and every file can be accessed. Therefore an attacker does not need a password of a user account to access the local data. And because it is this easy, replacing the password file with a manipulated one is easy. The only way to prevent this is to use encryption. Windows Vista includes two mechanisms for a partition wide encryption. One is to use the Encrypted File System (EFS), the other is BitLocker which is an application using a Trusted Platform Module (TPM) to encrypt the system partition. Both, EFS and BitLocker, could prevent an attacker from accessing the system partition and by this to access the password file. In this case all attacks described in this paper would not be possible because it is assumed that the password file can be accessed

and is not encrypted. Otherwise the attacker's only possibility would be to guess.

But encryption with EFS and BitLocker is currently rarely used and the attacker is able to obtain the password file. And although he would be able to access all local files there are still reasons why he would need to log in as a user known to the system. In most companies the files are not stored on the local PCs, but on a file server. If the attacker can obtain the password file then he can use the described methods to crack the password. Later the attacker can go back to the user's PC and carry out, attack using the cracked password.

Now the attacker acts as this user and can access all files the user could access. All is done in the name of this user and because of using the same PC nearly no evidence is given that someone else used the user's account. This show how important it is for everyone to choose a secure password to prevent fraudulent use.

The results of section 2.3 to 2.6 give an answer to the question" what is a secure password". As explained technically there are three ways to crack the password. Brute Force is the most reliable but even the most time-consuming attack, underlying the average time of a full run. The test showed that for 8 character passwords with lower case characters and numbers only, the time is still reasonable with a maximum of 12 days. But to crack the most complex password with 14 characters and all characters allowed even all computers on the world together would have needed more than a lifetime.

Adding upper case characters and staying with an 8 character password the possibilities for each position nearly doubles.

36^8 = 2.821.109.907.456(possibilities 8 characters password having lowercase (26) and numbers (10))

62^8 = 218.340.105.584.896 (possibilities for a password with 8 characters having lowercase (26), uppercase (26) and numbers (10))

218.340.105.584.896/2.821.109.907.456 = 77, 4

Now the maximum time would be 12* 77, 4= 928 days. To crack the password in a reasonable time more than 10 PCs would be needed. Hence, passwords should have a length of at least 8 with upper and lower case characters and numbers.

But as mentioned before people often use similar passwords that they can easily remember. Therefore, in most cases it is not needed to carry out a Brute Force Attack. With a high probability the password will be in a good organized dictionary file. The file contains a fraction of the overall possibilities and longer passwords than eight characters, too. As was stated in section 2.4 it took 5 minutes to reveal passwords with lower and upper case characters and numbers. It is very likely that every hacker possesses such a dictionary file or can get anything from the internet and he can easily upgrade it with new words of every length.

The more the attacker knows about the user the more he can narrow the number of possible passwords. The only way to prevent this attack is to not use whole words or simple combinations like"abcd","1234". The best would be to take a complete random password or at least a combination of letters looking random including numbers in between the letters.

As the result of the Dictionary Attack shows, it is very fast, but it is more or less restricted to commonly used passwords, too. The Rainbow tables in section 2.5and 2.6 include

random passwords but being approximately as fast as or even faster than a Dictionary Attack. In the examination it took less than 10 minutes to search all table files for different passwords. But instead of the Dictionary Attack only two passwords were found. The reason for this is the used table. It was made for 1 to 8 character passwords with lower case characters and numbers. On this account the attack was not able to find any other passwords, but if more random passwords were used for the first two user accounts, the result would have been different. The Dictionary Attack would have found the third password but not the first two, but the Rainbow table Attack would still have found the two passwords with a probability of 98%. Therefore, the kind of password a Rainbow table Attack can crack is related to the table. In order to make a statement about the security of a password a great effort is needed. But because the table is a pre computation of nearly all possibilities the effort must be nearly as high as a total Brute Force Attack.

What has to be taken into account is that the computation is done before the attack. Therefore, the attacker has more time then he would have for a Brute Force Attack and can compute a greater amount of passwords. Furthermore he can get the table from the internet and there are projects creating tables with the help of distributed computing. This enables an attacker to obtain tables for larger passwords than he would be able to compute himself. But this means that the assumption about the lenght of a password needs to be revised. For a Brute Force Attack a length of 8 characters (upper/lower case, numbers) was sufficient. When Rainbow tables are used the attacker is able to obtain tables for passwords with this complexity. Passwords that can be defined as secure against Rainbow table Attacks either need to be of the same complexity with a length of 9 or including special characters.

When the results of the three attacks are put together the minimum requirements for a secure password are to use lower and upper case characters, numbers and special characters with a minimum size of 8 characters. And in addition it should be randomly chosen because otherwise any information could help the attacker to narrow the password space even further.

If a really secure password is wanted than the length should be 10 or longer characters. This produces such a big password space that no complete table could be generated. The longer the password is the more secure it becomes.

Conclusions:

Significant progress has been made toward the goal of Evaluating Windows Vista user account security. A significant amount of work remains to be completed to achieve this goal, and encrypting the user account is expected to emerge as a common technology in the next few years.

The following conclusion can be drawn from testing the suggested model:

1-In the currently used normal configuration in windows vista the user accounts do not prohibit access to the system.

2- Replacing the password file with a manipulated one is possible in windows vista.

3- To prevent access to the system and replacing the password file the use of encryption is the only solution. Windows Vista includes two mechanisms for a partition wide encryption. One is to use the Encrypted File System (EFS), the other is BitLocker.

4- The minimum requirements for a secure password are to use lower and upper case characters, numbers and special characters with a minimum size of 10 characters. And in addition it should be randomly chosen because otherwise any information could help the attacker to narrow the password space even further.
5- The more the attacker knows about the user the more he can narrow the number of possible passwords.
6- Dictionary Attack shows, it is very fast, but it is more or less restricted to commonly used passwords, too. The Rainbow tables include random passwords but being approximately as fast as or even faster than a Dictionary Attack. The kind of password a Rainbow table Attack can crack is related to the table.

Reference:

1.Grimes, R. A. and Johansson, J. M. 2007, "Windows Vista Security: Securing Vista Against Malicious Attacks", John Wiley & Sons, Inc., USA.
2.Pfleeger,C.P. and Pfleeger, S.H. 2007, "Security in Computing", third edition, Pearson Education, Inc., New Jersey, USA. P (40).
3.Oechslin, P. 2003, "Making a Faster Cryptanalytic Time-Memory Trade Off ", In proceeding of Crypto 2003 int. Conference, Springer – Verlag, 2729 : 617 – 630.
4.Oechslin, P. 2005, "Password Cracking: Rainbow Tables Explained", (ISC)2, Newsletter, Security Transcends Technology, Palm Harbour, Florida, USA,
5.Kong, A. Zhang, D. and Kamel, M. 2006. "Analysis of Brute –Force Break-Ins of Palm Print Authentication System", IEEE Transaction System, MAN, and Cybernetics-part B: Cybernetics, 36 (5): 1201 – 1205.

تقييم امنية حساب المستخدم في ويندوز فيستا

علاء محمد عبد الهادي ***عمار عوني عباس****

*مركز الحاسبة-جامعةبغداد.

الخلاصة:

في نظام التشغيل ويندوزفستا كما في كل النسخ السابقة, يمكن عمل حساب لكل مستخدم من دون عمل كلمة عبور. ويكون ذلك دون مخاطر كبيرة في الحاسبات الشخصية على الرغم انه لاينصح بذلك حتى من شركة مايكروسوفت. بالنسبة الى الحاسبات التجارية يجب ان يحصر الدخول للحاسبات على اشخاص محددين باستخدام كلمات العبور لكل مستخدم.في النسخ القديمة من الويندوز توجد العديد من المصادر التي تنصح المستخدم بالطريقة المثالية لعمل كلمة العبور لحساب المستخدم. ويتضمن ذلك ملاحظات حول ملائمة تلك الحلول للاستخدام في ويندوز فيستا. لكن كل هذه المصادر لا تحدد نوع كلمات العبور الواجب استخدامها.لتقييم الحماية التي تقدمها كلمات العبور من المفيد تقييم فعالية التطبيقات المستخدمة لفتح كلمات العبور. يقوم هذا البحث بتحليل الطرائق التي يمكن للمهاجم من خلالها ان يحصل على كلمة المرور في الحاسبة التي يوجد بها ويندوز فيستا. هذا البحث يظهر ان كلمة مرور تتالف من ثمانية اجزاء وتتكون من حروف صغيرة وارقام من السهل فتحها اذا كانت تحتوي على كلمات معروفة. بينما الهجوم بوساطة القاموس لن يستطيع فتحها اذا لم تكن تحتوي على كلمات معروفة. اذا اضفنا حروف كبيرة على كلمة السر يصعب العملية اذ يزيد عدد الاحتمالات ومعظم الهجمات بوساطة القاموس سوف تفشل. يجب ان ياخذ بالحسبان حجم جدول قوس قزح عند القيام بهجوم على كلمة عبور, فاذا لم يكن الجدول كبيراً جداً حتى كلمات المرور الصغيرة لا يمكن فتحها. بالنسبة لكلمات المرور الطويلة فان كلمات المرور التي تحتوي على حروف صغيرة وارقام فقط هي التي يمكن فتحها. في هذه الحالة فان الهجوم العنيف ياخذ وقتاً كبيراً والقاموس سوف يحتوي على كلمات قليلة وحتى جداول قوس قزح سوف يكون طويلاً للاستخدام العادي. الهجوم سوف يكون ناجحاً اذا كانت هناك بعض المحددات للجدول وكان حجم الجدول محدداً.