



---

## Whitepaper

# Everything Administrators need to know about Windows password security.

---

© 2000 – 2008 nFront Security  
All Rights Reserved.

Altus Network Solutions, the Altus Network Solutions logo, nFront Password Filter, and the nFront Password Filter logo are trademarks of Altus Network Solutions, Inc. All other trademarks or registered trademarks are the property of their respective owners.

---

*Provided by Microset Systems Inc.*

[www.msi.net](http://www.msi.net) | 1-888-674-7674 | [sales@msi.net](mailto:sales@msi.net)



---

## Table of Contents

---

<b>1.0 What is in a password?</b> .....	<b>3</b>
1.1 Three methods of user authentication .....	3
1.2 Why is password security important .....	3
1.3 Who is at the most risk?.....	3
1.4 Common Passwords.....	4
<b>2.0 How Windows Passwords are Stored .....</b>	<b>5</b>
Syskey .....	6
Cached Credentials.....	6
<b>3.0 How passwords are compromised .....</b>	<b>7</b>
3.1 Getting Usernames .....	7
3.2 Getting the Password Hashes .....	7
3.3 Anatomy of a Password Cracker .....	8
3.4 The Methods of Password Cracking.....	8
The brute force attack .....	8
The dictionary attack .....	10
The hybrid attack.....	10
Rainbow Crackers .....	12
3.5 – Cracking the password hashes of cached accounts .....	13
3.6 Password Cracker Applications and Websites .....	14
<b>4.0 Avoiding Password Compromise .....</b>	<b>16</b>
4.1 Passwords versus Passphrases .....	16
4.2 User Education .....	16
4.3 Selecting a good password.....	16
4.4 Developing a good password policy .....	17
<b>5.0 Summary .....</b>	<b>17</b>
<b>About Us.....</b>	<b>18</b>

---

## 1.0 What is in a password?

---

A password exists for the sole purpose of allowing the computer to identify the end-user. If someone else has your password they can essentially take your identity on the corporate network.

There are other mechanisms which can prove your identity to the computer. Before we pickup with our discussion of passwords, let's explore those for just a moment.

### 1.1 Three methods of user authentication

There are 3 authentication systems available for most company networks:

1. **Passwords.** Passwords are chosen by end-users and vary in complexity and "hackability." Strong passwords contain a mix of character types and do not contain any words that could be found in a dictionary. Weak passwords are very simple and easily guessed (like the password of "password").
2. **Smart Cards.** Smart cards fall into a category called two-factor authentication. The smart card is like a banking ATM card. You must have the physical card and a PIN number to use the computer. The factors are the card (something you have) and the PIN (a number you know). Only with both factors are you allowed to logon. The problem for large companies is the cost of the card readers at over \$30 per user.
3. **Biometric Security.** Biometric security uses a voice recognition system, a fingerprint scanner, retina scanner, etc. to authenticate the user. Again the cost is very expensive.

Most companies cannot afford option 2 or 3 for all users so passwords are their only defense against computer "identity theft." Poorly chosen passwords expose many companies to unnecessary risk, especially if passwords are needed to access the wireless network or VPN.

### 1.2 Why is password security important

Passwords are important because a password is the only thing that identifies you to the computer system. If you choose a password that is too simple or easily-guessed you are exposing yourself as an easy target for hackers.

Compromised passwords are much more costly than many other security breaches. Security exploits like denial of service attacks (DoS) simply disable systems and their responsiveness for a period of time. Compromised passwords on the other hand, allow a hacker to takeover someone's identity on the network. Consequently, they may gain access to confidential data, trade secrets, customer information, passwords to other key systems, etc. Hackers can also use the passwords to do things like read the email of company executives, etc. If a company has a data breach that involves customer information the costs can be significant because the company's reputation is damaged.

### 1.3 Who is at the most risk?

Large companies are often at the most risk because they usually have weaker password policies. In most large companies they must use mainframe or UNIX systems for back-end data processing. Well, many UNIX and mainframe systems do not accept more than 8 character passwords. An 8 character password of only lower case letters can be cracked very quickly.

---

In many cases, large companies will run a password cracker periodically (once a week or once a month) and crack weak passwords. For accounts with weak passwords, the user is contacted and must change his or her password.

## 1.4 Common Passwords

There are countless surveys on passwords and password strength all over the Internet. In general many surveys show a high usage of words like:

password	abc123	qwerty123	123456
<username>	<username>123	letmein	january
password123	admin	changeme	<birthday>

One survey conducted by a British bank revealed the following password usage:

### Common passwords

- 23% child's name
- 19% partner's name
- 12% birthdays
- 9% football team
- 9% celebrities and bands
- 9% favorite places
- 8% own name
- 8% pet's name

Password hackers understand human nature. They know if your boss says to include a number of you are likely to

- add a number to the end of your password
- add a birthday and specific year to the end of your password
- separate 2 words with a number in the middle (like cat1dog)

With this in mind, hackers use wordlists and cracking programs with algorithms to account for our human nature to select “lazy” passwords.

There are actually many online websites where you can download free “wordlists” of common passwords or you can buy some as well. These lists can be used with cracking tools to help guess the user’s password more efficiently.

Here are a couple sites with wordlists:

<http://www.openwall.com/wordlists/> (for purchase)

<ftp://ftp.ox.ac.uk/pub/wordlists/> (for free)

---

## 2.0 How Windows Passwords are Stored

---

Windows does not store passwords in clear text. Instead all passwords go through a process to generate an LM hash and an NT hash. Hashes are one-way mathematical functions and for all practical purposes you can assume that hashed passwords cannot be reverse engineered. Here are the steps performed by Windows to generate each type of password hash.

### Lan Manager or LM Hash

1. Convert password to upper case
2. Pad the plaintext with null characters to make it 14 bytes long.
3. Split into two 7 character (byte) chunks
4. Use each 7 byte chunks separately as keys to DES encrypt an ASCII string constant
5. Concatenate the two cipher texts from step 4 to product the hash.
6. Store the hash in the SAM file.

### NT Hash

1. Take the Unicode mixed-case password and use the Message Digest 4 (MD4) algorithm to obtain the hash.
2. Store the hash in the SAM file.

The LanMan hash is only stored for passwords that are 14 characters or fewer (so if you are an admin always use a 15+ char password!). The NT Hash handles passwords up to 128 characters in length.

Each of the password hashes are stored as part of the System Accounts Database (SAM) on workstations and member servers. For domain controllers the information is stored in the Active Directory.

By design the LM Hash is weak for the following reasons:

1. It converts all passwords to upper case. This eliminates 26 characters that a hacker would otherwise have the guess or crack.
2. The splitting of passwords into two separate 7 character chunks cuts the cracking time in half since each chunk can be attacked separately. Furthermore, since the hash of seven null characters (`\0\0\0\0\0\0\0`) is known, a password cracker can immediately reveal if a user has a password that is less than 8 characters long.
3. The technique uses no SALT value. Of course neither does the NT Hash. In both cases the lack of a SALT value makes it possible to use techniques like Rainbow Cracking. A SALT is a random value that is used to hash the password. Both the password hash and SALT must be stored. Since the SALTS can vary it would eliminate the possibility of using pre-generated lookup tables (rainbow tables) with password crackers. Of course if a hacker is ambitious enough, he or she can generate new tables based on the SALT.

The password hashes are stored in the `c:\windows\system32\config\SAM` file which is world readable by default but exclusively locked by the operating system so you cannot readily view it, even as an administrator. Of course there are tricks to get around this. A great trick if you just want to look around is to launch the registry editor under a SYSTEM account. Go to a command prompt and type the following:

```
at \\comptuername 15:00 /interactive "regedt32.exe"
```

---

Do this where 15:00 is the next minute of your clock. You will notice that the registry editor opens and you can drill down much further than HKLM\SAM\SAM (which is where you are restricted as a user/admin). You will notice names, aliases, different hashes, etc.

Of course utilities like PWDump make life so much easier and use techniques like DLL injection to bypass syskey.

### ***Syskey***

Syskey is a technology developed by Microsoft to move the SAM database encryption key off the Windows server. The SysKey utility can also be used to configure a start-up password that must be entered to decrypt the system key so that Windows can access the SAM database. This approach was designed to keep old cracking utilities from cracking Windows passwords.

Of course, as hackers often do, they have discovered ways around this. Utilities like PWDump 2 use DLL injection to get the password hashes from lsass.exe bypassing SysKey. Newer versions of L0pht and others can also get around the SysKey system.

See Microsoft KB 310105 for more information on SysKey (<http://support.microsoft.com/kb/310105>).

### ***Cached Credentials***

Windows also caches the credentials of the last 10 interactive users who have logged on to a specific PC. The cached passwords are located in the registry in the following locations:

HKLM\SECURITY\CACHE\NL\$1  
Through  
HKLM\SECURITY\CACHE\NL\$10

There are several password cracking techniques for discovering the usernames and passwords contained in the cached account credentials. See the next section for more details.

---

## 3.0 How passwords are compromised

---

Hackers don't just sit around guessing passwords. When a hacker wishes to crack a password, he or she writes (or downloads) a program to perform a brute force attack, a dictionary attack, a hybrid attack or a program to perform rainbow cracking. A hacker may also use a bootable CD to hijack the username / password database from a local workstation or server if the machines are accessible.

### 3.1 Getting Usernames

It is usually not too challenging to get a list of usernames to try for the purpose of gaining access to a system. Often a company's website will give you some obvious choices. If the choices are not so obvious you can employ a tool like MetaGoofil (<http://www.edge-security.com/soft.php>) to pull all the metadata from Word docs, PPT files, PDF files, etc. posted on a company website. That gives you a great list of document authors (i.e. usernames) for a password cracking exploit.

### 3.2 Getting the Password Hashes

Getting the password hashes is easier than you think. Do not forget this: without physical security you have no security. Let's say your admin workstation is out in cubicle land with everyone else's machine. Now suppose I am a new hire in the Engineering department and I am working on my MCSE at night...on your network. Perhaps I decide to build a bootable CD and use it to boot my workstation and dump the local password hashes into a text file that I can take home to crack. If I can get the local admin password to my workstation I may be able to use that password to get to your workstation.

If I cannot get to your admin workstation using my local admin password, perhaps I may use something like Bart's PEBuilder (<http://www.nu2.nu/pebuilder/>) and a password change tool like Sala's Password Renew (<http://www.kood.org/windows-password-renew/>) to reset your local admin password on your workstation. Do you have a tool to check or monitor that? So then I can scan your local workstation for some useful items like passwords to other systems or data to help me with privilege escalation.

If that does not work, then perhaps I will plant a tool to give me a copy of your Protected Storage (you know, all your website passwords and stuff!). I could use a tool like Protected Storage PassView (<http://www.nirsoft.net/utils/pspv.html>) and I could plant it on your system using the Run registry key (HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run). I can set it to run in a silent mode and build a nice text file that I can use for cracking your protected storage passwords.

If someone cannot get the hashes, they can still use tools like THC-Hydra (<http://freeworld.thc.org/>) to crack passwords using brute-force, dictionary and hybrid techniques.

Of course, if I am a more ambitious hacker and not your average script kiddie I could write my own program / virus and get it into your system from the outside via email. The program could then sit on a workstation and listen for hashed passwords crossing the network (kind of like KerbCrack - [http://vil.nai.com/vil/content/v\\_100071.htm](http://vil.nai.com/vil/content/v_100071.htm)). I could then email those hashes offsite for cracking or run a local cracker.

So, in summary, there are many ways to get password hashes. It is much easier when on the inside of a network but certainly not impossible from the outside.

### 3.3 Anatomy of a Password Cracker

Password crackers can operate in many different ways and can try over 3 million passwords per second when “guessing” a password (Figure 2, note the Rate). Common password crackers include L0phtCrack, Crack 5, KerbCrack, John the Ripper (JtR) and Rainbow Crack.

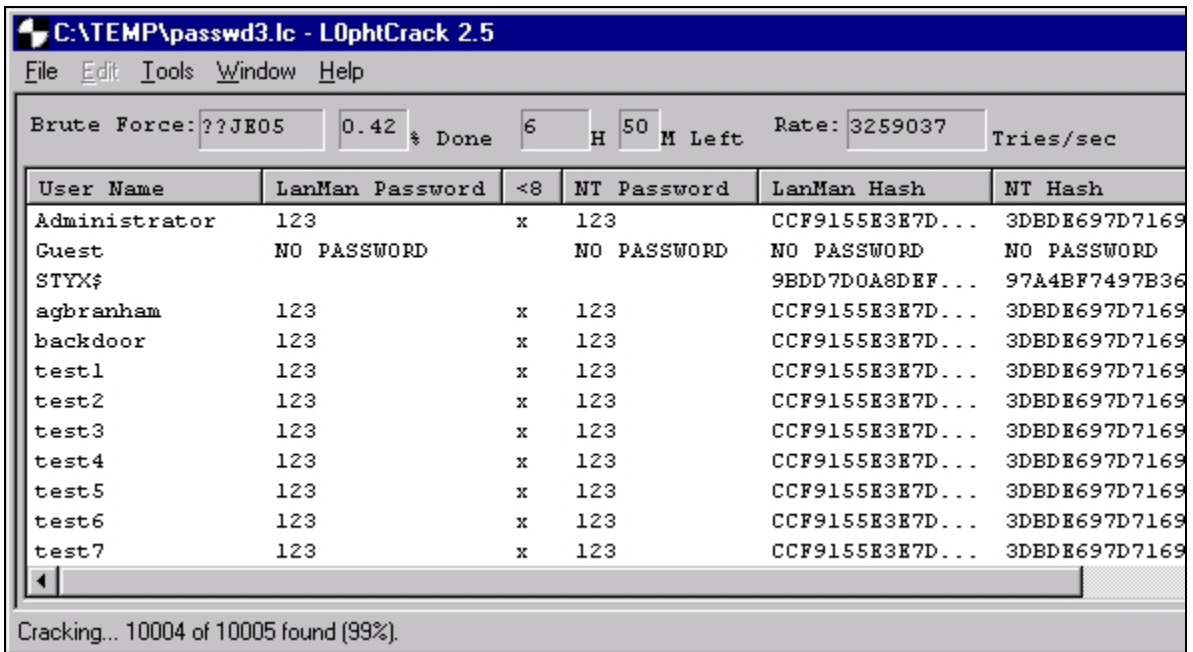


Figure 1: An example password cracker.

### 3.4 The Methods of Password Cracking

#### ***The brute force attack***

A *brute force attack* program starts by iterating through all of the possible character for a single-character password. Then the program moves on to two-character password combinations. When you consider there are 26 lower case letters, 26 upper case letters, 10 digits and about 32 non-alphanumeric characters, you have about 94 possible guesses for each character in a password.

For every additional character added to a password's length the possible character combinations increase exponentially. So for a two character password there are 94 possibilities for the first character and 94 possibilities for the second character. The total number of combinations is  $94 \times 94$  or  $94^2$ . That's only 8836 possible character combinations and would be easily cracked in less than one second with today's computing power.

*Given enough time and computing power any password can be cracked using a brute force technique.*



Figure 2: The brute force attack options

**How to combat the brute force attack:**

- 1. Force users to select longer passwords.** Longer passwords require more time to crack. Since the time increases exponentially the difference in cracking times for 6 letter versus 8 letter passwords is huge. nFront Password Filter can be used to require different minimum password lengths for different groups of users. Groups like Domain Admins should certainly use passwords that are 15 characters or more. Other groups can have lower minimums.
- 2. Force users to select passwords with characters from different character sets.** Notice the options in Figure 3. Often hackers will only use A-Z so the cracking attempt goes much faster. If you force users to use passwords containing special characters, the hackers efforts will go unrewarded and they will likely move on to easier prey. nFront Password Filter can be configured to require X of 4 character types and allows you to control the min and max of each character type accepted.

---

## ***The dictionary attack***

A *dictionary attack* program uses a "dictionary" of possible passwords. The dictionary may be a dictionary like Webster's Dictionary or it may be a dictionary containing common male names, common female names, common pet names, popular movie titles, slang words, etc. Dictionary password cracking is much, much faster than the *brute force* methods.

Hackers often use "dictionaries" of common passwords to attempt to logon to a system.

### **How to combat the dictionary attack:**

- 1. Force users to select passwords that are not in a hacking dictionary.** Ideally, you would like to check each user's password against a dictionary of common passwords. In fact many companies run a password cracker each night and disable accounts whose passwords are cracked. nFront Password Filter can apply dictionary checking using a customizable multiple language dictionary file. nFront Password Filter can scan a user's proposed new password against over 2 million common passwords in less than one second. If the password is found in the dictionary, the password change can be rejected. Thus, weak passwords are not even allowed on the network.
- 2. Force users to select passwords that contain a mixture of character types.** While this is a good idea it is not perfect. For example, many hacking dictionaries will contain '!@#\$\$%^' which is created by simply typing 123456 with the SHIFT key pressed. Such common sequences may pass a check for special characters but their simplistic approach is still vulnerable to password hackers. nFront Password Filter can force a password to contain X of 4 character types. The dictionary checking feature can be used to block trivial key sequences.

## ***The hybrid attack***

The hybrid attack leverages the dictionary by taking each dictionary entry and creating a few variations of the dictionary word (like adding a prefix or suffix of digits). Hackers know that if people have to put a number in a password they generally use a number at the beginning or end of the password.



**Figure 3: Note the Dictionary/Brute Hybrid Option**

Figure 4 shows a screenshot from L0phtCrack. However, programs like John the Ripper and Crack are much more elaborate. The following list shows some of the variations that can be used in the John the Ripper program.

- Append or prepend defined characters to a word.
- Reverse a word.
- Duplicate a word.
- Mirror a word, i.e. append the reversed word.
- Rotate a word either left or right, i.e. move the first letter to the end or the last letter to the front.
- Upper case a word.
- Lower case a word.
- Make only the first letter a capital.
- Make all but the first letter a capital.
- Toggle the case of all characters.
- Toggle the case of a character at a set position.
- Minimum and maximum word lengths can be set or long words can be truncated at a set length.
- Suffixes (s, ed, ing) may be added to words.
- First, last or any specific character may be deleted.
- Characters can be replaced at a set location.
- Characters can be inserted at a set location.
- "Shift" the case, i.e. substitute the other character on the same key, e.g. 'a' and 'A' or '5' and '%'.
- Shift the characters left or right by keyboard position (so an 's' becomes an 'a' or 'd').
- Replace all of one character with another.
- Replace all characters of a class (for example vowels, letters, non letters, digits) with a specific character.
- Remove all occurrences of any character from a word.
- Remove all characters of a class from a word.
- Reject a word if it contains or doesn't contain a character, or characters from a class.
- Reject a word if the first, last or set character is or is not a specific character or from a class.

- Reject a word unless it contains at least so many of a character or characters from a class.

This hybrid attack is a very dangerous threat because it leverages our human nature to make passwords resemble a pattern (like only capitalizing the first letter, or putting the number at the beginning or end instead of the middle).

#### How to combat the hybrid attack:

1. **Force users to select passwords that are not variations of a password from a hacking wordlist or dictionary.** Getting a user to select a complex password is a very, very difficult task. User education goes a really long way to help harden your network. User's need advice on selecting good passwords. They need to take approaches such as using the first letter from each word in a sentence that is easy to remember and including numeric and special characters. Use nFront Password Filter to perform dictionary substring checking will reject a password that contains any dictionary word in any variation of case (lower, upper or mixed case).

### **Rainbow Crackers**

Recall that a brute force attack goes through all combinations of characters, hashes each combination and tests the hash to see if it matches the target hash. Much of the time and computing power is spent generating the hashes. With that in mind, some people decided to pre-generate the password hashes for all possible 14 character or less passwords using various character sets. The result is a set of tables that can be used as lookup tables to discover a password. So the time to crack the password is now narrowed down to the time of lookup the hash in a table. Since the only time needed is that to lookup the hash is a very large table, it is an extremely fast cracking technique if you know the password hash.

**“It can crack the password Fgpyyh804423  
in 160 seconds.”**

Jeff Atwood  
www.codinghorror.com

So any password hash less than 14 characters can be cracked almost instantly if you have a copy of the password hash. Of course the tables are not just a few hundred KB. They range in size depending on the character set. The table below list the sizes and average crack times (according to the specifications found at Project Rainbowcrack, <http://www.antsight.com/zsl/rainbowcrack/>).

Char set	Table size	Load/crack time	System
A-Z	610 MB	6s/24s	P4 3Ghz,512MB
A-Z+0-9	3 GB	412/39s	Same
A-Z+0-9+Top keys	24 GB	148s/178s	P4 2.8 GHz,1GB
All keyboard char	64GB	290s/1658s	P4 3GHz,512MB

---

You do not even have to waste your time downloading tables. You can actually visit <http://www.plain-text.info/> and plug in your password hash value. Check back after a few minutes (depending on their queues and workload) and voila.

#### How to combat rainbow crackers:

1. **Turn off the storage of the LanMan Hash.** You can prevent Windows from storing the lanman hash. See the following KB article for the GPO and registry settings to do so: <http://support.microsoft.com/kb/299656>.
2. **Have Domain Admins and others use a 15 or more character password.** By using a 15 or more character password the lanman hash will not be stored. You can use nFront Password Filter to require a minimum of 15 characters or more for specific groups of users like Domain Admins.

Here are some great websites for further research on Rainbow Cracking:

Website	Description
<a href="http://www.antsight.com/zsl/rainbowcrack/">http://www.antsight.com/zsl/rainbowcrack/</a>	Good sites for further research on Rainbow Cracking.
<a href="http://en.wikipedia.org/wiki/Rainbow_table">http://en.wikipedia.org/wiki/Rainbow_table</a>	Wikipedia entry.
<a href="http://rainbowtables.shmoo.com/">http://rainbowtables.shmoo.com/</a>	Site to download rainbow tables.
<a href="http://www.ethicalhacker.net/content/view/94/24/">http://www.ethicalhacker.net/content/view/94/24/</a>	This is a well-written tutorial on rainbow cracking and using tools like Cain and Abel.

### 3.5 – Cracking the password hashes of cached accounts

Windows NT based systems like Windows 2000 and XP cache the passwords and userids for the last 10 interactive users. This allows workers to use laptops and other machines when the network is down or when traveling. The cached passwords are stored in the local registry in the following locations:

```
HKLM\SECURITY\CACHE\NL$1  
Through  
HKLM\SECURITY\CACHE\NL$10
```

The registry keys store a variety of information like the username, the domain, the NT hash and sometimes the LM hash. Until recently it was possible to use tools like CacheDump and patches for JTR and Cain to launch a password cracking attempt on the hashes. CacheDump seems to have disappeared off the face of the Earth. However, the current version of Cain can dump and cracked hashed passwords. Since it only cracks local cached passwords you may want to fumble with the CACHE.LST file used by Cain.

#### How to combat cached password hacking:

1. **You could set the CachedLogonsCount to 0 for desktops.**  
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\

---

CachedLogonsCount

Of course if the network is down and a domain controller is not available you are in trouble so consider this measure carefully before implementing.

2. **Never logon to a client computer with a Domain Admin account.** Always perform admin work on client workstations using the local admin logon.
3. **Do not let users have admin rights on their machine.** This will thwart the novice hackers since most cached pw dumping programs require admin rights. For the more experienced hackers, getting the info merely requires booting to an alternative OS and copying the SAM and SECURITY hive files for offline cracking.

### 3.6 Password Cracker Applications and Websites

There are a variety of password crackers out there. Before you go assembling your own hodgepodge of tools on your own CD or cool USB key, checkout what has already been done. You can find many tools online by simply searching for “pen testing” or “penetration testing” tools.

Currently one of the most comprehensive live CDs out there is the BackTrack and you can find it here ([www.remote-exploit.org/backtrack.html](http://www.remote-exploit.org/backtrack.html)).

<http://www.securityfocus.com/pen-test> - has some great articles on doing your own audits.

Here are some other tools / combinations of tools of interest:

- L0phtcrack (commercial) – A very popular tool for SAM cracking. It can read PWDump files, or dump the hashes itself. It can do dictionary, brute force and hybrid attacks. Admin access is needed to dump the hashes.

More info

<http://www.atstake.com/prodcuts/lc>

- SAMInside (commercial) - Password cracker that lets you get around SysKey by extracting the system key from the SYSTEM hive. You do not need to be an admin on the system to get the hashes, just copy off the SAM and SYSTEM files using a boot CD.

More info

<http://www.insidepro.com/eng/saminside.shtml>

- ElcomSoft (commercial) – Elcomsoft makes commercial password crackers for Windows, Adobe Acrobat, MS Office, Lotus, etc. They were one of the first applications to crack Vista passwords when Vista was released.

More info

<http://www.elcomsoft.com/>

- 
- PWDump2/Pwdump3 (free) - Uses DLL injection to get the password hashes from lsass.exe bypassing SysKey. Pwdump3 adds network support so you can dump hashes from across the network. You must be an administrator to dump the hashes.

More info

[http://www.bindview.com/Services/RAZOR/Utilities/Windows/pwdump2\\_readme.cfm](http://www.bindview.com/Services/RAZOR/Utilities/Windows/pwdump2_readme.cfm)  
<http://vh224401.truman.edu/pub/win32/apps/pwdump3/>

- Cain (free) – Cain is a very, very powerful tool that can perform almost all cracking techniques mentioned in this document. It has a few limitations like requiring admin privileges to dump hashes, only dumping local hashes, etc. However, many limitations can be circumvented by editing its local files.

More info

<http://www.oxid.it/cain.html>

- SAMDump2/Bkhive/John the Ripper (JTR) (free) - Nicola Cuomo discontinued development of SAMDump2 and Bkhive. However, the folks at Ophcrack have continued the development. The tools can be downloaded from the Ophcrack site. One tool dumps the SYSTEM hive and the other dumps the SAM hive. The SYSTEM hive is needed to extract the hashes from the SAM hive so both tools are needed to get the hashes for JTR to crack.

More info

<http://ophcrack.sourceforge.net/bkhive.php>  
<http://www.openwall.com/john>

- Ophcrack (free) – Ophcrack is a Windows password cracker that uses Rainbow Tables.

More info

<http://sourceforge.net/projects/ophcrack/>

---

## 4.0 Avoiding Password Compromise

---

Having everyone on your network select good, strong passwords is essential to your network security plan. Today, companies spend millions on perimeter security buying firewalls, intrusion detection products, intrusion prevention products, products to log web access, etc. In all cases, those products do no good if an external or internal hacker is using a legitimate username and password.

### 4.1 Passwords versus Passphrases

Passphrases are simply long passwords. Your Windows password can actually be as long as 127 characters (the XP GUI accepts on 29 char so you must use a command line to create longer passwords). Unless the passphrase is super simple (like “the quick brown fox”) it is superior to a password. Asking if passphrases are superior to passwords is like asking if long passwords are better than shorter ones. The answer is “Of Course, as long as the long password is not simple.”

Generally passphrases are thought of as sentences or phrases, thus, most passphrases will contain spaces and dictionary words. Passphrases should be easier for us to remember as humans. Remembering “@#tya\*~An&GH” as a password is almost impossible (without a post it to write it on!). However, remembering a phrase like “JAKE, my dog, ate the Wall Street Journal!” is much easier to remember even though it takes longer to type.

If you are seeking more clarification on the topic of passphrases, you may want to visit the Wikipedia entry (<http://en.wikipedia.org/wiki/Passphrase>). Also, you may be interested in reviewing the Technet discussion from October 2004 (<http://www.microsoft.com/technet/community/columns/secmgmt/sm1004.mspx>).

### 4.2 User Education

We cannot expect users to be security experts. However, we can expect them to take reasonable security pre-cautions.

Users need to know the dangers of weak passwords:

- Trivial and weak passwords put the entire company at risk.
- A weak 6 or 8 character password can be cracked very easily.

Users need to understand how to create smart passwords:

- Teach and encourage the use of passphrases.
- Teach your users techniques like thinking of a sentence and removing the vowels. (nFront Password Filter even has a setting to reject passwords with vowels. What a great way to eliminate dictionary words!).

### 4.3 Selecting a good password

Good passwords should exhibit the following characteristics:

- Good passwords never contain your username or any part of your full name.
- Unless the password is long (15 or more characters), the password contains no words at all. No words that could be found in a dictionary. No names of friends, children, spouses, etc. No names of pets. No names of colleges, universities, team mascots, etc.

- 
- Good passwords should be LONG. In fact a good password should be 15+ characters to avoid Rainbow Crackers. However, a password of 8 to 10 characters is usually sufficient for most non-admin accounts. Notice that the difficulty of password guessing increases exponentially with password length.
  - Passwords that substitute “@” for “a” and “1” for “i” are not as good as you think. Avoid this type of substitution since most hacking dictionaries contain these variations.
  - Good passwords contain at least one character from each of the following character sets: (1) Uppercase (2) Lowercase (3) Numeric and (4) Non-alphanumeric characters. By using characters from all for sets you are forcing the potential hacker to use a larger set of characters to guess the password. This exponentially increases the time to crack the password and convinces most hackers to find easier prey.
  - Good passwords are changed often.

#### 4.4 Developing a good password policy

We have worked with many companies and encountered many strange requests for nFront Password Filter modifications to meet overly-complex password policies. A good password policy should be easy for users to understand and result in passwords that are difficult for hackers to guess.

- The “hack-ability” of network passwords is limited to the weakest passwords allowed on your network (i.e. your password strength is determined by the weakest passwords on your network).
- Discourage users from using common words or personal information. A good technique for this is to disallow the use of vowels. Using a dictionary check is great but can only go so far. We have one hospital group that uses a 2.5 million word dictionary. Even with a dictionary of that size it is not possible to include every single college mascot, city name, hotel name, etc. Disallowing vowels is a better approach and easy to convey to end users. If you cannot disallow vowels, we would encourage you to at minimum enable the dictionary check and put obvious words like your company name into the dictionary.txt file.
- Force users to a mix of character types. Preferably you want users selecting passwords that contain uppercase, lowercase, numeric and non-alphanumeric characters.
- Encourage users to include special characters and numbers within the middle of the password and not just at the beginning and end. Hackers know that it is human nature to put these characters at the beginning or end. They also know that when we ask users to include a number most users will choose the number “1”

### 5.0 Summary

---

Good passwords are just as important as firewalls and antivirus software. The password crackers of today can operate at amazing speeds (like taking less than 3 minutes to crack a 14 character password with a rainbow cracker). You should implement a good password policy that require a minimum of 15 character for administrators. If you allow non-admin groups to use passwords of less than 15 characters you should ensure the passwords are sufficiently complex and do not contain the userid, trivial key sequences, etc. You should educate your users on the dangers of weak passwords and give them techniques and methods of create good passwords. You should make sure all of your service accounts have strong passwords.

---

## About Us

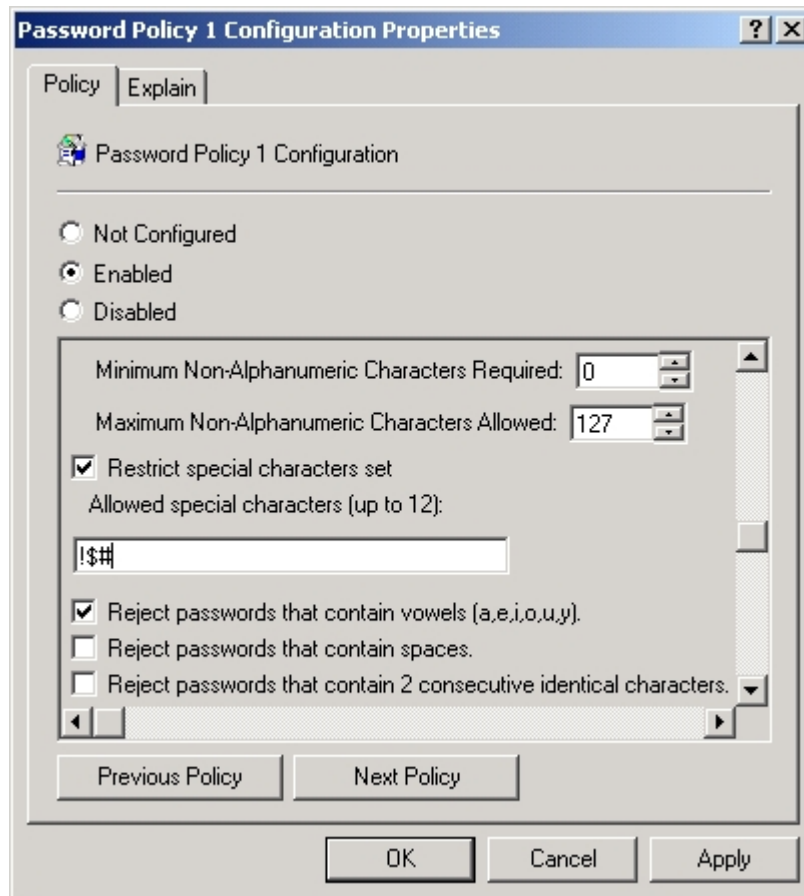
---

nFront Security is a division of Altus Network Solutions, Inc. Altus Network Solutions has provided consulting, training and programming services to companies worldwide since 1997. In 2003 Altus released Passfilt Pro, a password policy configuration and enforcement tool for Windows NT and Active Directory environments. The tool achieved widespread adoption by many major worldwide corporations. In August of 2007 nFront Security was created to move forward with the enhancement of the product (now called nFront Password Filter) and to take other security software products to market.

nFront Password Filter is a robust password policy enhancement and enforcement tool for Windows. The software is installed on Windows domain controllers and runs as a thread under the LSA. It is treated as an operating system component and cannot be bypassed like other simple password rule systems based on Java or other web technologies. The password rules are configured via a Group Policy Object. With the MPE edition you can create up to 6 different password policies with each policy linked to one or more global or universal security groups. The system also includes an optional client which can be deployed to workstations.

You can learn more at <http://www.nFrontSecurity.com>.

*Below are screenshots of the nFront Password Filter GPO configuration and the optional client.*



Example of nFront Password Filter policy settings.



Example of nFront Password Filter Client with optional password strength meter.