Windows Password Vulnerability and Preventive Measures

* Dinesh N. Patil
** B. B. Meshram

Abstract

With the rise in the use of the Internet, cyber crimes have also increased. One of the most prominent attack that can cause a breach in the security of sensitive system is hacking the password hashes. This paper discusses the window password hashes used in the Windows NT-based operating systems and the loopholes in them. The paper also covers various attacking techniques used by attackers in order to gain access to the password . The experiment carried out to identify and extract the password hashes from the volatile memory is also discussed. Finally the paper suggests mechanism for password protection.

***Keywords* : Hash function, Registry, Hives, Salting, Volatile memory, SAM**

I. Introduction

The use of passwords is known to be ancient. Guards at the entry point would challenge those wishing to enter an area or approaching it to provide a password, and would only allow a person or group to pass if they knew the password. In recent times, usernames and passwords are commonly used by people during a log in process that controls access to protected computer systems, mobile phones, and automated teller machines. Passwords have been used by computer users for many purposes such as logging into accounts, retrieving e-mail, accessing applications, databases, networks, web sites, and even reading the morning newspaper online. Passwords have been used with computers since the earliest days of computing. MIT's CTSS, one of the first time sharing systems, was introduced in 1961. It had a LOGIN command that requested a user password. After typing PASSWORD, the system turned off the printing mechanism.

Microsoft had provided the password protection for Windows 95/98 operating systems, which was having a password of 7 characters. Windows has two main flavors. The older flavors are referred to as "Win9x" and consist of Windows 95, 98, 98SE and Me. The newer flavors are referred to as "NT class" and consist of Windows NT3, NT4, 2000, XP and Vista and further versions [1]. In Windows 95, 98, and ME the encrypted passwords were stored on .pwl file causing easy hacking or disabling of the passwords. However, in later versions of Windows (NT class), Security Accounts Manager (SAM) database, or SAM database is used to store the password. SAM is a hive file that exists in the Windows registry and access to it is tightly controlled while window is running. Instead of storing your user account password in clear-text, Windows generates and stores user account passwords in the form of hashes. A hash is a small set of data that is mathematically tied to some larger set of data from which the hash is calculated [2]. Hashing results in the transformation of a string of characters into a shorter fixed length value representing the original string. Since it is difficult to find two messages with the same hash value, hash values are used to verify the integrity of messages. Operating systems often apply a hash function to a password and store the encrypted result instead of the plaintext password. The hash function has special properties. It is hard to determine the original password from the hash value and it is difficult to determine another password that has the same hash value. A user who needs to authenticate with

* *Ph. D. Scholar,* Veermata Jijabai Technological Institute, Matunga, Mumbai. E-mail : dinesh9371@gmail.com
** *Professor,* Veermata Jijabai Technological Institute, Matunga, Mumbai . E-mail : bbmeshram@vjti.org.in

the operating system tells the system his or her password, and the system hashes it and then compares the resulting value with the value stored in the password hash database. SAM File holds the password hashes for every account on the local machine, or domain. The location of this registry hives is found in %systemroot%/*system32\config\directory.* When the password for a user account contains fewer than 15 characters, Windows generates both a LAN Manager hash (LM hash) and a Windows New Technology LAN Manager hash (NTLM hash) of the password [4].

Various attempts have been made to break NTLM; however, LM is known to be broken. Attackers are using various techniques to break the password. SAM does not allow direct access to the password but there are certain tools using which hashes can be extracted from the SAM database of the registry. Microsoft also provided facility for encryption of SAM file by using SYSKey encryption to protect the SAM file that hosts hashed passwords [11]. Even though SYSKey provides an additional layer of protection, tools such as SAMDUMP can recover the SYSKey boot key from the system hive by using Bkhive to dump the SAM file. Such recovery provides an attacker a copy of the user's hashed password. Another problem is that SYSKey does not protect passwords while they are loaded in memory. Once the hashes are obtained, then there are more chances of password being broken by the attacker.

The organization of this paper is as follows: section 2 introduces the basic concepts of the LM and NTLM hashes and their vulnerability. It also covers the various approaches that have been used by attackers in order to gain access to password. Section 3 is about the experimentation based on the volatility-2.0 tool to extract hashes from the SAM and the consequent possible decoding of the hashes. Section 4 suggests protection mechanism to avoid the breaching of the security of the password. Section 5 concludes the result.

II. The Windows Password Hashes

Windows NT-based operating systems today are the most widely used operating system in the world today. This popularity had made it vulnerable to targets for various kinds of hackers, intruders and dishonest users. The rise of the Internet has worsened the situation. To protect the personal data of the user and the data about the system from the attackers, password protection technology is being extensively used. Password protection is the primary protection in the Windows operating system. In order to prevent attackers from gaining access to the password, hashes are computed from the original plaintext password and stored in the SAM hive file of the Windows registry. This section discusses the LM hashes and NTLM hashes created in the Windows NT-based operating systems.

A. LM Hash

The LM hash is also known as the LAN Manager hash. The LAN Manager hash is a primary hash function that is being used to store password for Microsoft Windows versions prior to Windows NT. However LM hash is still being continued in later versions of Windows for backward compatibility. It is computed as follows [7]:

1. Convert all lower case characters in the plain text password to upper case
2. Pad the password with NULL characters until it is exactly 14 characters long
3. Split the password into two 7 character chunks
4. Use each chunk separately as a DES key to encrypt a specific string "KGS!@#$%"
5. Concatenate the two cipher texts resulting LM hash having size of 128-bit and store the result in SAM hive file

However ,the LM hash has its vulnerability and can be broken. The vulnerability of the LM hash is as follows:

The LM hash takes a 14-character password and splits it into two parts. If the password is less than 14-characters then the last part is padded with null characters. The halves can be cracked separately, which makes it easy to guess the password if one of the hives is cracked. The defense against this is to disable LM hashes. Another is to have the number of 14 characters in the password.

2. The LM hash also does not use cryptographic salt - a standard technique to prevent pre-computed dictionary attacks.

A time-memory trade-off cryptanalysis attack such as a rainbow table is therefore possible.

3. Only upper-case character set is considered in making the password.

This reduces the task of the attacker in cracking the password. Attackers use password cracking tools to crack LM Hashes.

DES encryption was cracked by the Electronic Frontier Foundation in 1998 in about 23 hours [10].The LM hash is weak, and it is therefore, prone to fast brute force attack. Microsoft has long recommended that it be disabled.

B. NTLM Hash

In NTLM, each character in the input password is converted into Unicode (16-bits characters representation) before applying the MD4 encryption algorithm. The hash length obtained is 128 bits and works for local account and Domain account (Active Directory account). A Microsoft TechNet article [8] indicated NTLM hashes were applicable to Windows 7, Server 2003, Server 2008, Server 2008 R2, and Vista. Another TechNet article [9] stated that NTLM is used for Windows 8, 8.1, Server 2012, and Server 2012 R2.The NTLM hash is computed as follows [14]:

1. Convert the plain text password to Unicode (Little Endean format)
2. Add the byte zero after each character of the password
3. Apply MD4 Algorithm to the password obtained from step2, resulting in NTLM hash

However, the NTLM hash has its vulnerability which can be exploited by the attacker to break it. The vulnerability of the NTLM hash is as follows:

1. The weakness of MD4 Algorithm

Since NTLM hash is created using MD4, the shortcoming in MD4 may be exploited.

2. Byte Zero insertion

The byte zero inserted after each character in the password are ASCII strings, the attacker can predict half of the message and its position.

3. No use of Salt

Salt is a random numeric value that is combined with password before computing the one-way function. This can be exploited by the attacker in calculating hashes for all possible passwords.

NTLM remains vulnerable to Pass-the-hash attack, which is a variant of the Reflection attack. Several flaws were discovered in Windows implementation of the NTLM authentication mechanism in February 2010 by Amplia Security which broke its security allowing attackers to gain read/write access to files and remote code execution.

III. Password Attack Techniques

Access to modern computer systems is controlled by passwords. In order to gain access to the target system, attacking the password is a straight forward approach being used by attackers. Attackers are coming with new techniques and tools to get access to the password of the target system. Some of the techniques are discussed in this section.

1. *Password Guessing:* The most used password attack is password guessing. The password is guessed by the attacker using either a manual or automated approach. Attackers basically try multiple combinations of usernames and *passwords* until the one that works is found out. Some of the tools used for password guessing are TSGrinder, SQLRecon, and Hydra.

Password guessing attacks can be classified into two viz., Brute Force Attack and Dictionary Attack.

Brute Force Attack: A Brute Force attack is a type of password guessing attack and it consists of trying every possible code, combination, or password until you find the correct one. This type of attack may take long time to succeed if the password is complex.

Dictionary Attack: In this type of password guessing attack, attacker uses a dictionary of common words to identify the user's password.

2.*Key Logger Attack:* A keylogger is any piece of software or hardware that has the capability to intercept and record input from the keyboard of a compromised machine [12]. The keylogger often has the ability to sit between the keyboard and the operating system and intercept all of the communications without the user's knowledge. A hacker uses a program to track all of a user's keystrokes. Everything that users type including their login IDs and passwords can be recorded. The two main categories are software-based and hardware-based keyloggers. The most commonly used kind of keylogger is a software-based tool, often installed as part of a larger piece of malware, such as a Trojan or rootkit.

3.*Phishing:* Attacker sends phishing mail which lead to a malicious website where the user gives authentication details considering the malicious website as the original. The authentication details may be username and password of online banking system.

4.*Shoulder surfing:* A person working in the organization may collaborate with the attacker and may try to get the details of the authentication by watching a user of interest closely when the authorized user is logging into the system.

5.*Offline cracking:* If the system is on and unattended for a long time, then malicious insiders in an organization may extract the hashes from the Windows registry. These passwords hashes can then be cracked using password cracker tool. Tools used are John the Ripper, Ophcrack, and Pwdump.

6.Rainbow tables: A rainbow table consists of hashes computed for all possible passwords. A rainbow table can have a size of hundred of gigabytes. The attacker after extracting hash from a target system performs the look up in the rainbow table for a possible match with the plaintext password.

7.Password Resetting: The attacker boots from a CD-program to avoid the typical Windows protection. Password resetters programs contain a bootable version of Linux that can mount NTFS volumes and can help in locating and resetting the password of Administrators.

A widely used password reset tool is the free Petter Nordahl-Hagen program.

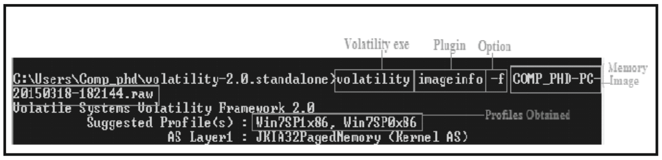
8.Pass-the-hash:In Pass-the-hash attack, an attacker steals hashed user credentials and, without cracking it, reuses it to trick an authentication system into creating a new authenticated session. The tools used in Pass-the-hash attack are Pshtoolkit, Msvctl, and Metasploit.

Even though passwords are the most convenient means of authentication, they are prone to attack. The attacker may attack a password either online or offline. While offline attacks are possible only if an attacker has physical access to the system, online attacks can be performed by eavesdropping on the information being passed on the network.

IV. Experiment

In order to find out the vulnerability of Windows password, the experiment was carried out using volatility framework tool to extract hashes from the volatile memory of Windows7 as in [13]. Volatility is implemented in Python under the GNU General Public License, for the extraction of digital artifacts from volatile memory (RAM) samples. Extraction techniques are performed completely independent of the system being investigated but offer unprecedented visibility into the runtime state of the system.

Step1. Obtaining the profile of the image

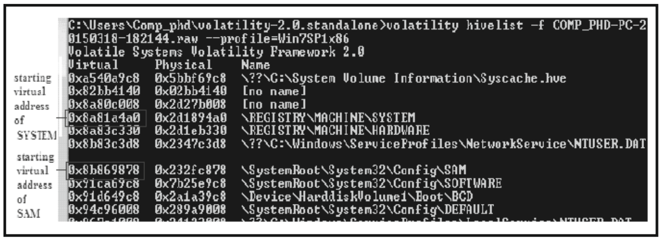


```
C:\Users\Comp_phd\volatility-2.0.standalone>volatility imageinfo -f COMP_PHD-PC-
20150318-182144.raw
Volatile Systems Volatility Framework 2.0
          Suggested Profile(s) : Win7SP1x86, Win7SP0x86
                     AS Layer1 : JKIA32PagedMemory (Kernel AS)
```

Fig 1. Volatility command to obtain profile of image

In order to use a particular data structure, algorithms, and symbols, volatility needs the profile information of the memory image obtained from a system. This profile information is obtained using image info plugin with the volatility tool as shown in Fig. 1. The –f option specifies the filename.

Step2. Obtaining the information about the hive



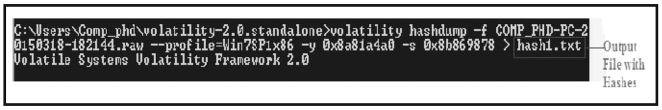
```
C:\Users\Comp_phd\volatility-2.0.standalone>volatility hivelist -f COMP_PHD-PC-2
0150318-182144.raw --profile=Win7SP1x86
Volatile Systems Volatility Framework 2.0
Virtual     Physical    Name
0xa540a9c8  0x5bbf69c8  \??\C:\System Volume Information\Syscache.hve
0x82bb4140  0x02bb4140  [no name]
0x8a80c008  0x2d27b008  [no name]
0x8a81a4a0  0x2d1894a0  \REGISTRY\MACHINE\SYSTEM
0x8a83c330  0x2d1eb330  \REGISTRY\MACHINE\HARDWARE
0x8b83c3d8  0x2347c3d8  \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT

0x8b869878  0x232fc878  \SystemRoot\System32\Config\SAM
0x91ca69c8  0x7b25e9c8  \SystemRoot\System32\Config\SOFTWARE
0x91d649c8  0x2a1a39c8  \Device\HarddiskVolume1\Boot\BCD
0x94c96008  0x289a9008  \SystemRoot\System32\Config\DEFAULT
```

starting virtual address of SYSTEM
starting virtual address of SAM

Fig. 2. Volatility command to extract hive virtual address

The hives are loaded in the physical memory when the system is on. The starting virtual addresses of the hives such as SYSTEM and SAM are required to extract information from them. The SYSTEM hive needs to know whether the user account has password or not. The SAM hive consists of the authentication details for the user.The hivelist plugin is used to extract hive details for a specific profile of a memory image.

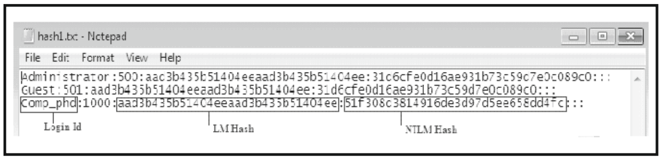
Step3 . Obtaining the hashes



```
C:\Users\Comp_phd\volatility-2.0.standalone>volatility hashdump -f COMP_PHD-PC-2
0150318-182144.raw --profile=Win7SP1x86 -y 0x8a81a4a0 -s 0x8b869878 > hash1.txt
Volatile Systems Volatility Framework 2.0
```

Fig. 3. Volatility command to extract hashes

By providing the profile information and virtual address of the SYSTEM and SAM hives the hashes can be extracted and stored in a text file using hashdump plugin. Fig. 3 shows hashes extracted from SAM having the starting virtual address of 0x8b869878 are stored in the text file hash1.txt. The hashes are obtained for all the users of the system. Since the SAM maintains the authentication details of all the users of the system,



```
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Comp_phd:1000:aad3b435b51404eeaad3b435b51404ee:51f308c3814916de3d97d5ee658dd4fc:::
```

Fig. 4. Sample hashes in a text file

Step 4. Decoding the Hashes

Once the hashes are extracted they can be decoded either by using rainbow table or any other hash cracking tool. In addition online tools are available for hash cracking which can be used by the attacker.

The above experimentation shows that hash can be extracted from the SAM hive of the registry and once it is extracted, it can be decoded. The password remains safe if it is not loaded in the volatile memory; however the Windows password remains vulnerable to attack when the hashes are loaded in the volatile memory.

V. Preventive Measures

This section discusses the preventive measures that are needed to implement in the existing password protection mechanism to protect it from getting hacked by the hacker. The measures are as follows:

A. Salting of the password

The hashing of each and every password is done exactly the same way. This is the main reason that the rainbow table works for the cracking the hashes.

To avoid this the password is needed to be salted, which means appending or prepending a random string called salt. The salted password is then hashed. As the attacker won't be able to know what the salt is, so he won't be able to precompute the rainbow table. However, the salt should not be reused to avoid being guessed. The length of the salt should be long enough to avoid building rainbow table by the attacker.

The following example [15] illustrates how it is hard to crack the passwords if these are salted. As the salt is a random string, a hash function such as SHA256 generates different hash for the same password for each salt.

The original text is 'hello' which is hashed using hash function *hash*

hash("hello")

The resulting hashes

2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

The text 'hello' is now appended with the salt 'QxLUF1bgIAdeQX' and the hash function is applied to it.

hash("hello" + "QxLUF1bgIAdeQX")

The resulting hashes are obtained as below.

9e209040c863f84a31e719795b2577523954739fe5ed3b58a75cff2127075ed1

The following examples demonstrate the use of different salts and the resulting hashes by using the same hash function.

hash("hello" + "bv5PehSMfV11Cd") = d1d3ec2e6f20fd420d50e2642992841d8338a314b8ea157c9e18477aaef226ab

hash("hello" + "YYLmfY6IehjZMQ") = a49670c3c18b9e079b9cfaf51634f563dc8ae3070db2c4a8544305df1b60f007

As the above example depicts that if the password is appended with salt and then hashed, it results in different hashes for each hashing.

B. Increasing the complexity of the password

A password is not immune from being cracked even though being long. If the password is comprised of some combination of letters, numbers, punctuation marks, mathematical and other conventional symbols, then password guessing by the hacker will get quite tougher. Also forming the rainbow table won't be easy.

A strong password should:

❖ Have at least eight characters

❖ Not contain your username, real name, or company name

❖ Not contain a complete word.

❖ Be significantly different from previous passwords.

❖ Contains characters from each of the following four categories:

TABLE I. CHARACTERS FOR PASSWORD

Character category	Examples
Uppercase letters	A, B, C
Lowercase letters	a, b, c
Numbers	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Symbols found on the keyboard (all keyboard characters not defined as letters or numerals) and spaces	` ~ ! @ # $ % ^ & * ( ) _ - + = { } [ ] \ | : ; " ' < > , . ? /

C. Evolving Algorithms with increased iteration rate for obtaining hashes

The hashing algorithms that require many computing cycles or the iteration rate for generating the hashes may increase the cost for attackers, as the decryption of the hashes consumes time and other resources. Nowadays, memory-hard functions are becoming more popular as the hashing algorithm because of the large computing power required for generating hashes.

If the iterations of an algorithm are increased for obtaining the hashes it increases the computing time for obtaining hashes. This can be improved by designing the hardware implementation of the algorithm with higher iteration rate. Further, the hardware design should comprise of pipelining and parallel execution of instructions to reduce the time for computing independent instructions.

VI. Conclusion and Future Work

The Windows operating system stores password in the form of hashes. Even though hashes are obtained using one way hash function, still these can be broken once extracted from the volatile memory. The preventive measures if implemented in generating password and its hashes will considerably decrease the chances of cracking the password.

Study needs to be carried in the future out for password protection mechanism in Windows10 and finding out the loopholes in it and suggesting protection mechanism.

References

[1] "A comparison of Linux and Windows." Available : http://www.michaelhorowitz.com/Linux.vs.Windows.html, 2007

[2] "Passwords Technical Overview," Available: https://technet.microsoft.com/en-us /library/hh994558%28v=ws.10%29.aspx,2012

[3] "Defending the pass-the-hash attacks" Available: http://www.microsoft/com/security/sir/strategy/default.aspx# !Password_hashes, 2015

[4] "How to prevent Windows from storing a LAN manager hash of your password in Active Directory and local SAM databases?" Available: https://support.microsoft.com/en-us/kb/299656,2015

[5] R. Allen, *Active Directory cookbook*, 3rd edition,O'Reilly Media publications, Dec 2008

[6] D. Todorov, *Mechanics of User Identification and Authentication*,Auerbach Publications, June, 2007

[7] "Microsoft Windows 2000 Security Hardening Guide" Available: https://technet.microsoft.com/en-us/library/dd277300.aspx#ECAA,2003

[8] "Password Technical Overview" Available: http://technet.microsoft.com/en-us/library/hh994558%28v=ws.10%29.aspx , 2012

[9] "NTLM Overview". Available: http://technet.microsoft.com/en-us/library/hh831571.aspx, 2012

[10] Sanders, "How I cracked your windows password [part-1]" . Available : http://www.windowsecurity.com/articlesQTutorials/authentication and encryption/HowQCrackedQWindowsQPasswordQ Part1.html, 2010

[11] George Khalil, SANS Institute, "Password Security--Thirty-Five Years Later", 2014

[12] D. Fisher, "What is a keylogger?". Available: https://blog.kaspersky.co.in/keylogger/, 2013

[13] D. Dieterle , "Memory forensics: How to pull password from a memory dump," in *Cyber Arms-computer Security*, 2011

[14] "Selecting Secure Passwords". Available : https://technet.microsoft.com/en-us/library/cc875839.aspx,2015

[15] "Salted Password Hashing –Doing it right". Available: https://crackstation.net/hashing-security.htm, 2016