How I Cracked your Windows Password (Part 2)

Going through the process of cracking passwords with different free tolls whilst providing tips for defending your password from being cracked.

- Published: **Feb 10, 2010**
- Updated: **Feb 10, 2010**
- Section: **Articles :: Authentication, Access Control & Encryption**
- Author: **Chris Sanders**
- Company: **EWA Government Systems, Inc.**
- Rating: **4.1/5 - 14 Votes**

If you would like to read the first part in this article series please go to How I Cracked your Windows Password (Part 1).

Introduction

In the first part of this series we examined password hashes and the mechanisms Windows utilizes to create and store those values. We also touched upon the weaknesses of each method and possible avenues that can be used to crack those passwords. In the second and final article in this series I will actually walk you through the process of cracking passwords with different free tools and provide some tips for defending against having your password cracked.

It is always crucial to note that the techniques shown here are strictly for educational purposes and should not be used against systems for which you do not have authorization for.

Obtaining Password Hashes

In order to crack passwords you must first obtain the hashes stored within the operating system. These hashes are stored in the Windows SAM file. This file is located on your system at C:\Windows\System32\config but is not accessible while the operating system is booted up. These values are also stored in the registry at HKEY_LOCAL_MACHINE\SAM, but again this area of the registry is also not accessible while the operating system is booted.

There are a few different options here depending on the level of access you have to the machine you are auditing.

Physical Access

If you have physical access, one of the most effective methods is to boot the computer into a different operating system. If you are comfortable using Linux then this means you can simply boot to a Linux live CD that is capable of reading NTFS drives, mount the Windows partition, and copy the SAM file to external media.

If you are not quite comfortable doing this, you can use P. Nordahl's famed Offline NT Password Editor, available here. This is a bootable Linux distribution designed to aid system users who have forgotten their passwords by allowing them to reset them. The software takes the users input, creates a valid hash, and replaces the old hash in the SAM file with the new one. This is useful to us because we can also use the distribution to simply read the SAM file and get the hash data.

In order to do this, boot from the CD image and select your system partition, the location of the SAM file and registry hives, choose the password reset option [1], launch the built in registry editor [9], browse to SAM\Domain\Account\Users, browse to the directory

of the user you wish to access, and use the cat command to view the hash contained in the files. The output will be in hex format, but it works with a simple conversion.

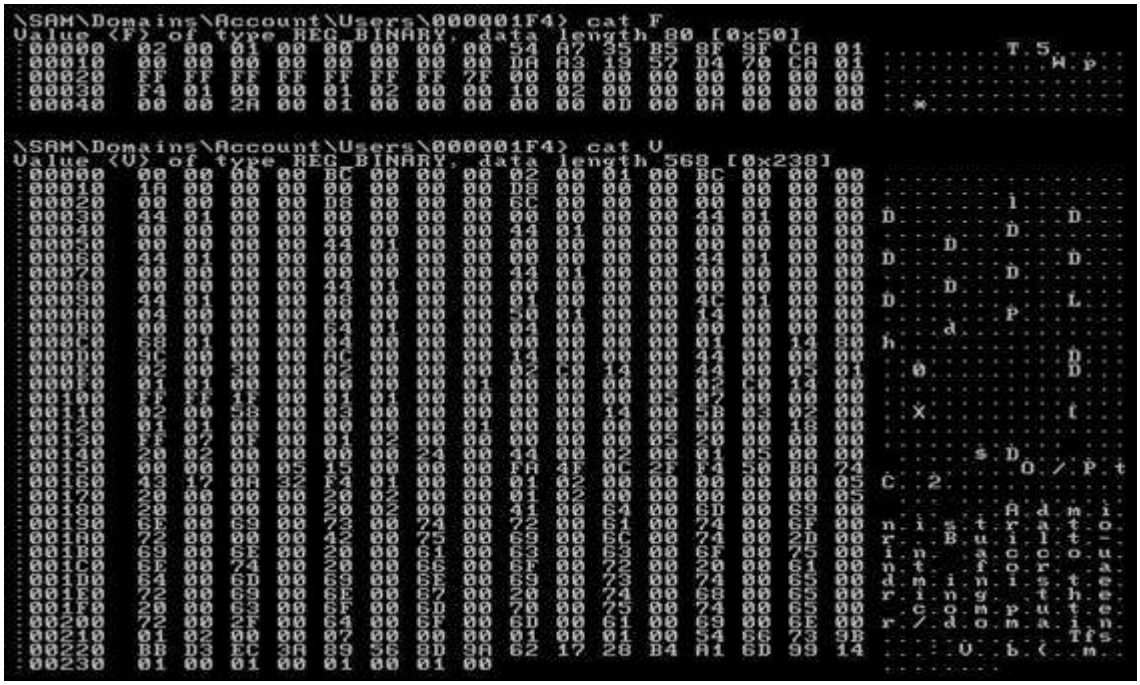

Figure 1: Hex output of the SAM hash

Before using the Offline NT Password Editor to actually reset a password, be sure that you are not using Encrypted File System (EFS) on anything released after Windows XP/2003. If you do this, it will cause the operating system to lose its EFS keys, resulting in more problems than just a forgotten password.

Console Access

If you are performing password auditing activities without physical access to the device in question, but you still have console access through remote desktop or VNC, then you can obtain password hashes through the use Fizzgig's fgdump utility, obtainable here.

Once you have downloaded fgdump to host you can simply run it with no options to create a dump of the local machine SAM file.

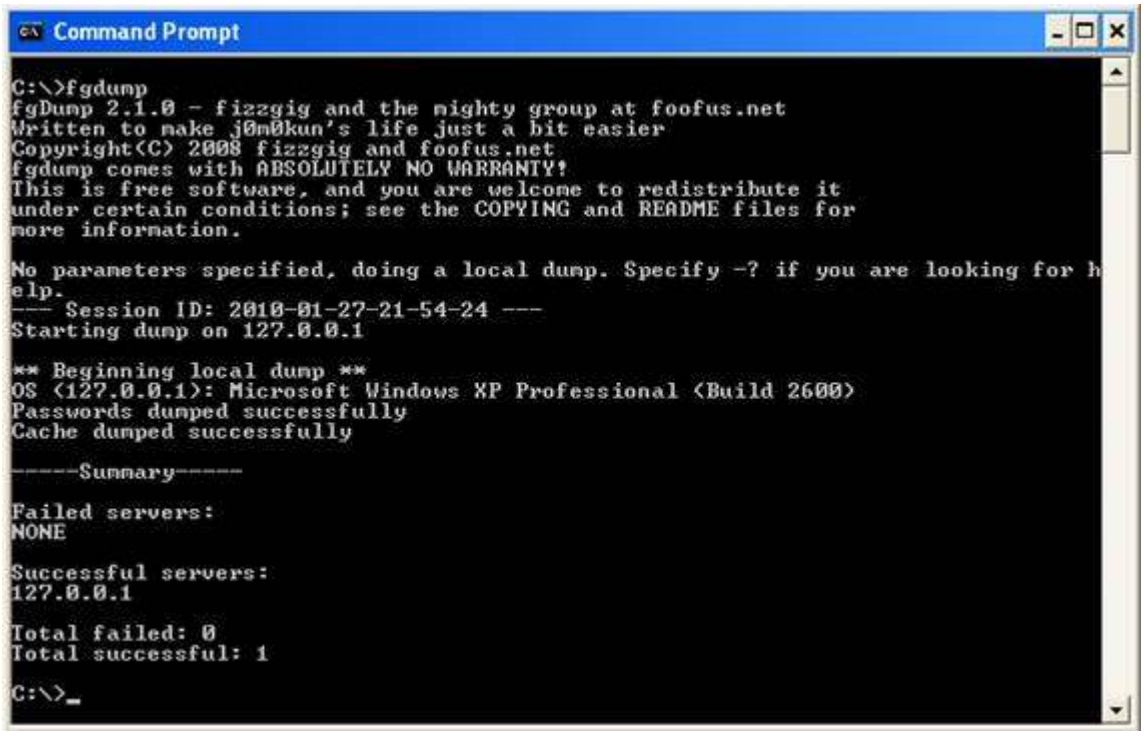

Figure 2: Confirmation the Fgdump Utility Ran Correctly

Once this is completed, a file will be generated in the same directory the utility was launched from that contains a list of all user accounts, their LM hashes, and their NTLMv2 hashes.

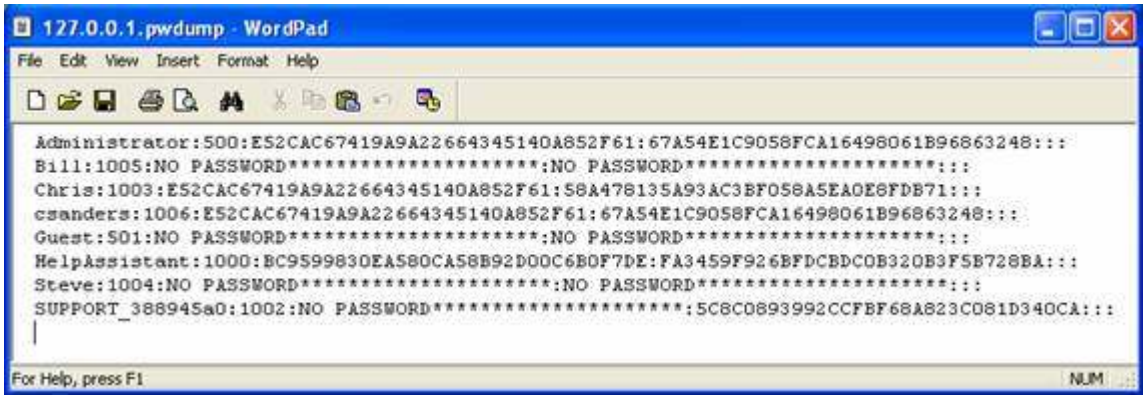

Figure 3: Password Hashes Output by Fgdump

Network Access

Finally, if you do not have any interactive access to the machine that has the hashes you want, your best bet is to attempt to sniff the hashes as they travel across the network during the authentication process. Of course, this will only work if the client is authenticating to a domain controller or accessing resources on another client, otherwise, you are more out of luck than a one armed man in a paper hanging contest.

If you are on the same network segment as the target client you can use the Cain and Abel program to intercept the password hashes as they are transmitted between devices. Cain and Abel is a free utility downloadable from here. Using Cain and Abel you can initiate a process called ARP cache poisoning, which is a man in the middle attack that takes advantage of the ARP protocol to route the traffic between two hosts through your computer. While ARP cache poisoning is active you can use Cain and Abel's built in network sniffer, making it possible for you to intercept NTLM password hashes that are being communicated between the poisoned hosts. The theory behind ARP cache poisoning and how to do it are another lesson in itself and a bit beyond the scope of this article, but if you wish to learn more about ARP cache poisoning you can do so here.

Cracking Passwords Using Cain and Abel

Now that we actually have password hashes we can try to crack them. If you have already downloaded and installed Cain & Abel then you are already a step ahead because we will be using it to crack our sample LM passwords.

If you have not yet installed Cain and Abel you can download it from here. The installation is just a matter of hitting next a few times. If you do not already have it installed, you will also be prompted to install the WinPCap packet capture driver used for Cain and Abel's sniffing features. Once installed you can launch the program and click on the Cracker tab near the top of the screen. After doing this, click on the LM & NTLM Hashes header in the pane on the left, right click in the blank area in the center of the screen, and select Add to List.

Cain will not accept a simple copy and paste of the password hash, so you will have to place the hash in a text file formatted a special way. If you extracted your hashes using fgdump then you should already have the text file you need, which contains hashes on a line by line format.

Figure 4: Accepted Formatting of Passwords Hashes

If you extracted your password hashes manually you will need to create a file with a line entry for every user account. Each line should contain the username, the relative identifier (RID) portion of the users SID, and the hashes. The format of these elements should be:

Username:RID:LMHash:NTLMHash:::

Browse to this file, select it, and click next to import the hashes into Cain and Abel. Once this is done, you can right click the account whose password you want to crack, select the Brute Force Attack option, and choose LM hashes. The brute force attack method attempts every possible password combination against the hash value until it finds a match. On the screen that follows you can select the characters you want to use for the brute force attack and the minimum and maximum password lengths. Notice that the character set is automatically configured to use only uppercase characters and number with a maximum length of 7, due to the characteristics of LM hashes.

In our example scenario where we have a password of PassWord123 we will see immediate partial results as the program returns that "Plaintext of 664345140A852F61 is D123". We have already cracked the second half of the password hash. On a modern computer, going through every single possible password combination should take no longer than 2 ½ to 3 hours, guaranteeing an eventual success.

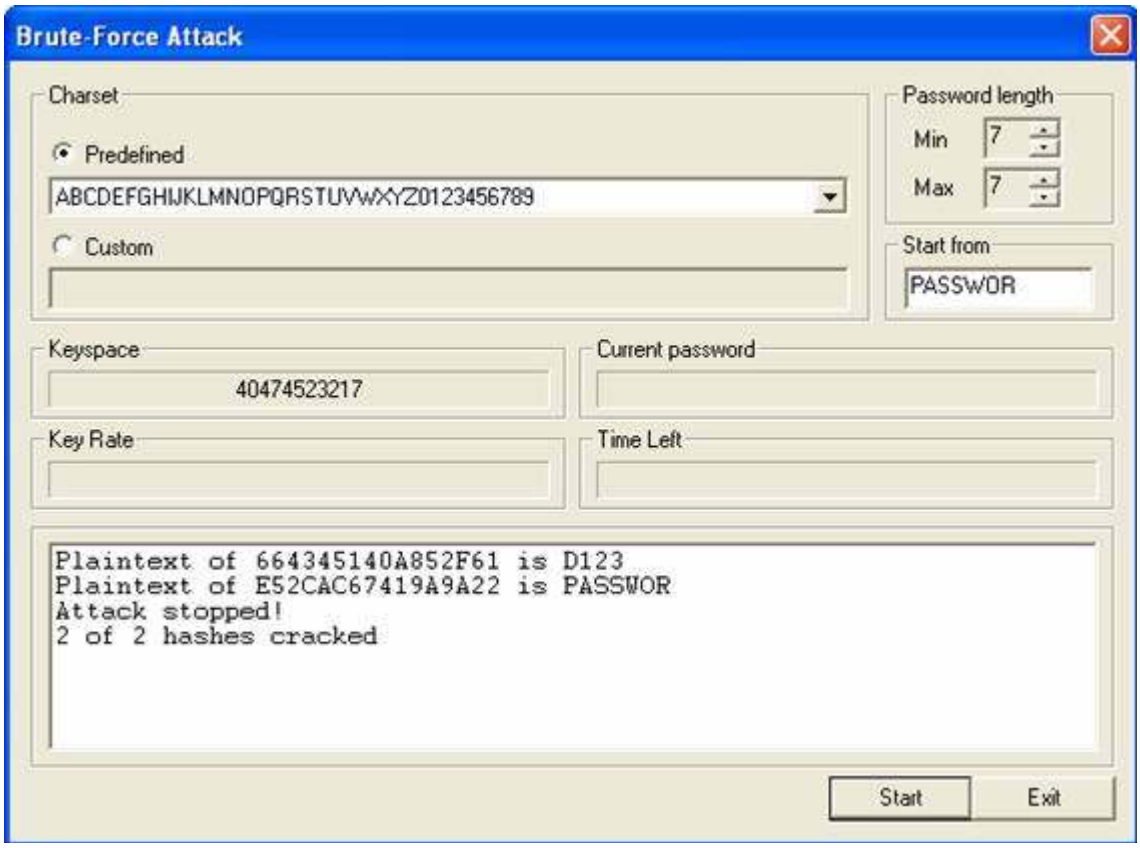

Figure 5: Cain Successfully Cracks the LM Password Hash

Cracking Passwords Using John the Ripper

Cain and Abel does a good job of cracking LM passwords but it is a bit slow and its functionality for cracking NTLMv2 hashes is even slower. If you are comfortable using the command line for your password cracking activities, then John the Ripper is one of the fastest and most highly preferred cracking engines.

You can download John the Ripper from here. Once you have extracted the contents of the file you will find the john-386.exe executable in the /run subdirectory. John has a few different modes it can be run in, but to run it in its default mode all you have to do is supply the file containing the password hash as an argument when you run the executable from a command prompt.

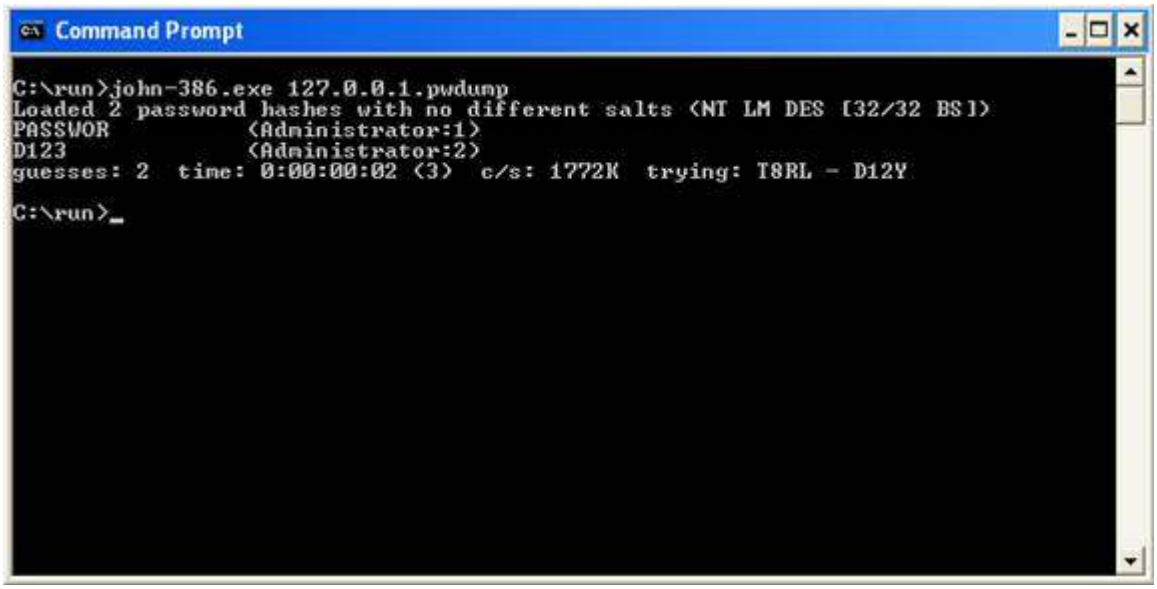

Figure 6: John the Ripper Attempting to Crack a Password

Once it has completed, John the Ripper displays the cracked passwords and stores the results in its john.pot file. In most situations the default cracking mode is fine, but John the Ripper also has these cracking modes available:

- Single Crack Mode – Uses variations of the account name
- Wordlist Mode – Relies on a dictionary for password guesses
- Incremental Mode – Relies on a brute-force style attack
- External Mode – Relies on another (user supplied) application for password guessing

John is very efficient in all of its cracking modes and is my typical program of choice for password cracking.

Cracking Passwords Using Rainbow Tables

When you suspect an NTLMv2 password of being highly complex and in turn being too time consuming to crack, the only logical resolution is the use of rainbow tables. A rainbow table is a lookup table consisting of password hashes for every possible password combination given the encryption algorithm used. As you can imagine, rainbows tables can take up quite a bit of storage space. In the past these tables were far too processor and storage space intensive to create and store, but with the advances of modern computing its becoming more and more common for both ethical penetration testers and malicious hackers to keep external hard drives containing sets of rainbow tables.

Finding a place to generate or download a set of rainbow tables is just a Google search away if you prefer to do that, but there are better methods for the “casual” password cracker. One such method is by using a web service containing its own set of rainbow tables. One such web service is this. This site maintains multiple sets of rainbow tables for which you can submit password hashes for cracking, along with a list of recently cracked passwords for efficiency.

In order to submit hashes to plain-text.info you can simply click the Add Hashes link to specify the hash and encryption mode. If this hash has already been cracked then you will be displayed results, and if not this will submit the hash into the queue. You can monitor the queue status by going to the Search link and searching for the hash, which will tell you its queue position. Complex passwords can often taken some time via this method, but it is typically quicker than allowing your own hardware to do the work.

Defending Against Password Cracking

People tend to think that the goal of encryption is to make encrypted text to where nobody can ever decipher it, but this is a bit of an ill conceived notion. That thought relies on the belief that computers are able to generate random numbers for the purposes of encryption, but in all honestly computers don’t do “random” so well, as “random” is completely reliant upon programmed logic. As a result of this, the real goal of encryption is to make the encrypted text so hard to crack that the amount of time it would take to crack outweighs the benefit of doing so.

With this in mind, there are a few things that can be done on a windows system to prevent your password from being cracked.

Use Complex and Changing Passwords

The most logical way to prevent people from cracking your password is to make it incredibly complex. If your password contains lowercase letters, uppercase letters, numbers, special symbols, and is fairly long, it won't be able to be cracked in any reasonable amount of time. In order to given things an added degree of complexity, changing your password frequently means that when an attacker cracks your password it will have already been changed. There is no single greater defense than using a strong password that is changed frequently.

Disable LM Hashing

By now you should be thoroughly versed on the weaknesses of LM hashes. The good thing for us is that we do not have to use them anymore. Modern Windows operating systems can be configured to use NTLMv2 exclusively with a few registry modifications.

You can disable the storage of LM hashes by browsing to HKLM\System\CurrentControlSet\Control\LSA in the registry. Once there, create a DWORD key named NoLMHash, with a value of 1.

Another step is to disable LM authentication across the network. Once again, browse to HKLM\CurrentControlSet\Control\LSA. Once there, locate the key named LMCompatibiltyLevel. This can be set to 3 to send NTLMv2 authentication only which is a great setting for domain clients. The alternative is to set this value to 5 which configured the device to only accept NTLMv2 authentication requests, which is great for servers.

The only instance in which these settings might cause an issue are cases in which you have Windows NT 4 and older client on your network. However, in all honesty, if you still have those types of systems on your network then getting rid of them is the best security device I can give you.

Use SYSKEY

SYSKEY is a Windows feature which can be implemented to add an extra 128 bits of encryption to the SAM file. SYSKEY works by the use of a user created key which is used to encrypt the SAM file. Once enabled, SYSKEY cannot be disabled.

It's important to keep in mind that SYSKEY only protects the SAM file itself, securing it against being copied. SYSKEY does NOT protect against tools which extract hashes from running memory, such as Cain and fgdump.

You can read more about SYSKEY at http://support.microsoft.com/kb/143475.

Conclusion

Password cracking is an instrumental skill for someone attempting to break into a system, and because of this it is a necessity that system administrators understand how passwords are stored, stolen, and cracked. As potential intruders poke and prod at systems their mouths will water at the sight of an LM hash and their goal will be more than half way completed if users are using simple passwords. Remember, knowing is half the battle, so if you take this information and do nothing about it you are only half way there. Using the defensive techniques provided you can help deter attackers from compromising passwords of your systems.

If you would like to read the first part in this article series please go to How I Cracked your Windows Password (Part 1).