Cracking Passwords Version 1.1

by: J. Dravet

February 15, 2010

Abstract

This document is for people who want to learn to the how and why of password cracking. There is a lot of information being presented and you should READ IT ALL BEFORE you attempted doing anything documented here. I do my best to provide step by step instructions along with the reasons for doing it this way. Other times I will point to a particular website where you find the information. In those cases someone else has done what I attempting and did a good or great job and I did not want to steal their hard work. These instructions have several excerpts from a combination of posts from pureh@te, granger53, irongeek, PrairieFire, RaginRob, stasik, and Solar Designer. I would also like to thank each of them and others for the help they have provided me on the BackTrack forum.

I will cover both getting the SAM from inside windows and from the BackTrack CD, DVD, or USB flash drive. The SAM is the Security Accounts Manager database where local usernames and passwords are stored. For legal purposes I am using my own system for this article. The first step is to get a copy of pwdump. You can choose one from http://en.wikipedia.org/wiki/Pwdump. Update: I used to use pwdump7 to dump my passwords, however I have come across a new utility called fgdump from http://www.foofus.net/fizzgig/fgdump/ This new utility will dump passwords from clients and Active Directory (Windows 2000 and 2003 for sure, not sure about Windows 2008) where pwdump7 only dumps client passwords. I have included a sample hash.txt that has simple passwords and should be cracked very easily. NOTE: Some anti-virus software packages flag pwdump* and fgdump as trojan horse programs or some other unwanted program. If necessary, you can add an exclusion for fgdump and/or pwdump to your anti-virus package so it won't flag them. However it is better for the community if you contact your anti-virus vendor and ask them to not flag the tool as a virus/malware/trojan horse.

You can find the latest version of this document at http://www.backtrack-linux.org/

Contents

1 LM vs. NTLM

The LM hash is the old style hash used in MS operating systems before NT 3.1. It converts the password to

uppercase, null-pads or truncates the password to 14 characters. The password is split into two 7 character halves and uses the DES algorithm. NT 3.1 to XP SP2 supports LM hashes for backward compatibility and is enabled by default. Vista supports LM hashes but is disabled by default. Given the weaknesses in the LM hash it is recommended to disable using LM hashes for all MS operating systems using the steps in http://support.microsoft.com/kb/299656

NTLM was introduced in NT 3.1 and does not covert the password to uppercase, does not break the password apart, and supports password lengths greater than 14. There are two versions of NTLM v1 and v2. Do to a weakness in NTLM v1 is should not be used. Microsoft has included support for NTLM v2 for all of its operating systems either via service pack or the Directory Services client (for windows 9X). You enable NTLM v2 by following the instructions at http://support.microsoft.com/kb/239869. For maximum security you should set the LMCompatibility to 3 for Windows 9X and LMCompatibilityLevel to 5 for NT, 2000, XP, and 2003. Of course you should test these changes BEFORE you put them into a production environment.

If LM hashes are disabled on your system the output of pwdump and/or the 127.0.0.1.pwdump text file will look like:

Administrator:500:NO PASSWORD**********:00AB1D1285F410C30A83B435F2CA798D:::
Guest:501:NO PASSWORD**********:31A6CAE0D36AD931B76C59D7E1C039C0:::
HelpAssistant:1000:NO PASSWORD**********:BF23C2595478A6279F7CB53EF76E601F:::
SUPPORT_3845a0:1002:NO
PASSWORD**********:0C8D62E10A6240BACD910C8AB295BB79:::
ASPNET:1005:9F07AE96CA4310752BDC083AAC960496:A99C1C3DB39E3C732EF5C2F63579AF96:::

The first field is the username. The second field is the last four numbers of the SID for that username. The SID is a security identifier that is unique to each username. The third field is the LM hash. The forth field is the NTLM hash.

If you do not have a ASPNET user account do not worry about it. If you do have a ASPNET user account do NOT change the password as I am told that will break something. What I did was delete the account and then recreate it using: systemroot%\Microsoft.NET\Framework\v1.1.4322\aspnet_regiis.exe /i

2 Syskey

To make it more difficult to crack your passwords, use syskey. For more information on syskey see http://support.microsoft.com/kb/310105. The short version is syskey encrypts the SAM. The weakest option but most convenient is to store a system generated password locally; locally means the registry. The up side is the SAM gets encrypted and you can reboot the server remotely without extra equipment. The next option is password startup. This is slightly more difficult to get around, but if you remotely reboot the server, it will stop and wait for someone to enter the password. You will need a KVM over IP or a serial port concentrator so you can enter the password remotely. The most secure option is the system generated password stored on a floppy disk. The downside to this option is floppy disks fail, you misplace the floppy disk, newer equipment does not have a floppy disk drive, no remote reboots, and you will probably leave the floppy in the drive so you can remote reboot and that defeats security. I use a system generated password stored locally, weak but better than not doing it. To disable syskey use chntpw and follow its instructions.

3 Cracking Windows Passwords

3.1 Extracting the hashes from the Windows SAM

3.1.1 Using BackTrack Tools

3.1.1.1 Using bkhive and samdump2 v1.1.1 (BT2 and BT3)

1. # mount /dev/hda1 /mnt/XXX
   mount your windows partition substituting hda1 for whatever your windows partition is
2. if the syskey password is stored locally you need to extract it from the registry so you can decrypt the SAM. If syskey is setup to prompt for a password or the password is on a floppy, stop now and read the syskey documentation in this document for more information about syskey. If you installed windows to something other C:\WINDOWS please substitute the correct path. WARNING the path is case sensitive. The filenames of sam, security, and system are case sensitive. On my system these files are lowercase. I have come across other XP systems where they are uppercase. On the Vista system I have used the filenames are uppercase.

   BackTrack 2 users use the following:
   # bkhive-linux /mnt/XXX/WINDOWS/system32/config/system syskey.txt

   BackTrack 3 users use the following:
   # bkhive /mnt/XXX/WINDOWS/system32/config/system syskey.txt

3. # samdump2 /mnt/XXX/WINDOWS/system32/config/sam syskey.txt >hash.txt
   samdump2 will dump the SAM to the screen and the > character redirects the output to a file called hash.txt
   you can also run samdump2 with the -o parameter to write the output to a file
   # samdump2 -o hash.txt /mnt/XXX/WINDOWS/system32/config/sam syskey.txt

3.1.1.2 Using new samdump2 v2.0 (BT4)

The current version is 2.0.1 and has the benefit of being able to extract the syskey on its own. This means dumping the hashes in now a 1 step process instead of two. To upgrade and run sampdump2 v2.0.1:

1. download the current sampdump2 from http://sourceforge.net/project/showfiles.php?group_id=133599
2. # tar -xjvf samdump2-2.0.1.tar.bz2
3. # cd samdump2-2.0.1
4. # make
5. # cp samdump2 /usr/local/bin/samdump20
   this will keep the existing version. If you want to overwrite the existing version do:
   # cp samdump2 /usr/local/bin/
6. mount your windows partition substituting hda1 for whatever your windows partition is
   # mount /dev/hda1 /mnt/XXX
7. if the syskey password is stored locally samdump2 v2.0 will extract it from the registry so it can decrypt the SAM. If syskey is setup to prompt for a password or the password is on a floppy, stop now and read the syskey documentation in this document for more information about syskey. If you installed windows to something other C:\WINDOWS please substitute the correct path. WARNING the path is case sensitive. The filenames of sam, security, and system are case sensitive. On my system these files are lowercase. I have come across other XP systems where they are uppercase. On the Vista system I have used the filenames are uppercase.
8. # samdump2 /mnt/XXX/WINDOWS/system32/config/system /mnt/XXX/WINDOWS/system32 /config/sam >hash.txt
   samdump2 will dump the SAM to the screen and the > character redirects the output to a file called hash.txt
   you can also run samdump2 with the -o parameter to write the output to a file
   # samdump2 -o hash.txt /mnt/XXX/WINDOWS/system32/config/sam syskey.txt

3.1.1.3 Cached Credentials

The only Linux based application to dump cached credentials I found is creddump which can be found at http://code.google.com/p/creddump/. samdump v2.0.1 couldn't do this so I wrote the code to dump cached credentials. I have submitted it upstream so I hope to see this feature in the next version.

3.1.2 Using Windows Tools

3.1.2.1 Using fgdump

To dump local passwords:

1. Login to the system as an administrator and get to a command prompt (Start, Run, cmd). Since this my system I know administrator password. You could also try to use metasploit to attack your system to get to a command prompt.
2. Download one of the fgdump files from http://swamp.foofus.net/fizzgig/fgdump/downloads.htm and unzip it.
3. run the fgdump utility you downloaded
   C:\> fgdump -v
4. copy the 127.0.0.1.pwdump file to a floppy or USB thumb drive if you are going to use BackTrack to crack the hashes

You can dump passwords from remote systems but only if you know the remote local administrator password or have domain administrator privledges.

1. Login to the system as an administrator and get to a command prompt (Start, Run, cmd). Since this my system I know administrator password. You could also try to use metasploit to attack your system to get to a command prompt.
2. Download one of the fgdump files from http://swamp.foofus.net/fizzgig/fgdump/downloads.htm and unzip it.
3. run the fgdump utility you downloaded
   C:\> fgdump -v -h hostname -u Username -p Password
   where hostname is the name or ip of the remote system you want to retreive the passwords from
   Username is the username of the account to connect to the remote system with; usually Administrator or Domain\Administrator or an account with Domain Administrator privledges.
   Password is the password of the above account
   NOTE: If you have a firewall installed on the remote system this will not work.
4. copy the 127.0.0.1.pwdump file to a floppy or USB thumb drive if you are going to use BackTrack to crack the hashes

3.1.2.2 Using gsecdump

Thanks to williamc for pointing out another password dumping tool. These instructions are based on the Exploitation part of his Intranet Exploitation tutorial.

1. Login to the system as an administrator and get to a command prompt (Start, Run, cmd). Since this my system I know administrator password. You could also try to use metasploit to attack your system to get to a command prompt.
2. Download the gsecdump file from http://www.truesec.com/PublicStore/catalog/categoryinfo.aspx?cid=223. You have to click on the Hamta filen link to download it. Once downloaded, unzip it.

3. Download the psexec tool from http://technet.microsoft.com/en-us/sysinternals/bb897553.aspx and unzip it
4. run as follows:
   C:\> psexec \\hostname -u username -p password -s -f -c gsecdump.exe -s > hash.txt
   hostname is the name of the PC where you want psexec to run. AKA the target.
   username is the username to login to the remote PC.
   password is the password of the above username. If you don't put in it here you be prompted to enter it. If you are prompted the password won't be displayed.
   -s tells the process to runas the system account
   -f copies the program to the target pc even if it exists
   -c copies the program to the target pc
   gsecdump.exe is the utility you want to run
   -s tells gsecdump to dump the SAM/AD hashes
   the > character redirects the output to a file called hash.txt
   NOTE: If you have a firewall installed on the remote system this will not work.
5. copy the hash.txt file to a floppy or USB thumb drive if you are going to use BackTrack to crack the hashes

3.1.2.3 Using pwdump

1. Download one of the pwdump files from http://en.wikipedia.org/wiki/Pwdump and unzip it.
2. Login to the system as an administrator and get to a command prompt (Start, Run, cmd). Since this my system I know administrator password. You could also try to use metasploit to attack your system to get to a command prompt.
3. run the pwdump utility you downloaded
   C:\> pwdump7 >c:\hash.txt
   pwdump7 will dump the SAM to the screen and the > character redirects the output to a file called hash.txt
4. copy the hash.txt file to a floppy or USB thumb drive if you are going to use BackTrack to crack the hashes

3.1.2.4 Cached Credentials

When a user logs into a domain their password is cached in the registry so that in the event that the Domain Controller or network goes down the user can still login to their PC. To export these registry keys you need a tool call cachedump. It can be downloaded from ftp://ftp.openwall.com/john/contrib/cachedump/

The readme.txt in the zip contains everything you want to know about where the cached credentials are stored, how cached credentials work, how they are hashed, and how the tool works. Cachedump does not work on Windows Vista. Vista changed the way that cached creditionals work.

You can also download the fgdump with source file from http://www.foofus.net/fizzgig/fgdump/ and get cachedump and its source code.

To use:

1. Extract the cachedump.exe from the zip
2. Login to the PC as an administrator
3. Goto a cmd prompt (Start, Run, cmd)
4. C:\> cd \path to cachedump.exe
5. C:\> cachedump.exe -v

This runs cachedump.exe in verbose mode. I suggest running cachedump in verbose the first time you use it so you know what is going on and can identify any problems. Once you have good information displayed on the screen you can use:
C:\> cachedump.exe >cache.txt
and this will redirect the output from the screen to a file called cache.txt

Now you can use John The Ripper or Cain and Abel to crack the hashes. Please note that Cached Credentials use a different hash than LM or NTLM. The lowercase username is salted with the password.

The best way to protect yourself from this is to disable cached credentials. Change the value of the following registry key: HKEY_LOCAL_MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\WINLOGON\CACHEDLOGONSCOUNT to 0. You can do this manually or with Group Policy.

3.2 Extracting Windows Password hashes remotely

3.2.1 Using BackTrack Tools

3.2.1.1 Using ettercap

You can use ettercap and the man in the middle attacks to sniff the username and password of a user over the network. DO NOT ATTEMPT THIS WITHOUT PERMISSION OF THE USER WHOSE ACCOUNT YOU WANT TO SNIFF.

You can read an ettercap tutorial at http://openmaniak.com/ettercap.php which covers the basics on how to use ettercap. There so much that ettercap can do and there are many tutorials covering how to use it I am not going to duplicate the effort. Just do a quick search using your favorite internet search engine for ettercap tutorials and read.

3.2.1.2 Using hashdump (metasploit)

I am not going to cover this in great detail. To use hashdump you first have to use metasploit to compromise the PC from which you want the password hashes. There are already a number of tutorials that explain how to use metaspolit. The best documentation is at http://www.metasploit.com/framework/support/. Once you have compromised the PC using metasploit you can extract the hashes doing:

use priv

hashdump

3.2.2 Using Windows Tools

3.2.2.1 Using fgdump

1. Download one of the fgdump files from http://swamp.foofus.net/fizzgig/fgdump/downloads.htm and unzip it.
2. Login to the system as an administrator and get to a command prompt (Start, Run, cmd). Since this my system I know administrator password. You will need the administrator password or the username and password of a an account that is in the local Administrators group on the PC from which you want the hashes.
3. run the fgdump utility you downloaded

C:\> fgdump -v -h hostname or IP_Address_of_Target -u username -p password
where username and password are an account with administrator privileges.

copy the 127.0.0.1.pwdump file to a floppy or USB thumb drive if you are going to use BackTrack to crack the hashes

3.2.2.2 Using pwdump6

1. Download pwdump6 from http://en.wikipedia.org/wiki/Pwdump and unzip it.
2. Login to the system as an administrator and get to a command prompt (Start, Run, cmd). Since this my system I know administrator password. You will need the administrator password or the username and password of a an account that is in the local Administrators group on the PC from which you want the hashes.
3. run the utility you downloaded
   C:\> pwdump6 -u username -p password hostname or IP_Address_of_Target>c:\hash.txt
   where username and password are an account with administrator privileges.
   pwdump6 will dump the SAM to the screen and the > character redirects the output to a file called hash.txt
4. copy the hash.txt file to a floppy or USB thumb drive if you are going to use BackTrack to crack the hashes

3.3 Cracking Windows Passwords

3.3.1 Using BackTrack Tools

My strategy for cracking windows passwords is like this:

1. Get/Develop a really good wordlist/dictionary
2. Find the password policy that is enforced for the account you are trying crack
3. Crack the LM hash using John the Ripper
4. Crack the NTLM hash with the results of the cracked LM hash and the password policy information using mdcrack

If there is no LM hash to crack I proceed to cracking with John the Ripper using the password policy information and my wordlist. Then I use rainbowtables if the tables match the password policy. http://plain-text.info is back up and running so I check if they have the cracked password already. To successfully use a rainbow table you need to know the password policy. No sense downloading a rainbow table that contains letters and numbers when the password policy requires a symbol (!@#$%^&* etc).

3.3.1.1 John the Ripper BT3 and BT4

Version 1.7.2 shipped with BackTrack 3. Version 1.7.3.1 with jumbo patch 5 shipped with BT4. The john the ripper that ships with BT4 requires at least a P4 with SSE2 instructions. If you don't have a processor that supports SSE2 then you have download and compile john yourself. See the next section for instructions on how to do this.

3.3.1.1.1 Cracking the LM hash

john only needs to know the path to the hash.txt to begin bruteforcing and return the uppercase password
/usr/local/john/john hash.txt

3.3.1.1.2 Cracking the NTLM hash

john only needs to know the path to the hash.txt to begin bruteforcing and return the password
/usr/local/john/john --format:NT hash.txt
will begin to bruteforce the NTLM hashes

3.3.1.1.3 Cracking the NTLM hash using the cracked LM hash

Stasik told me it is much easier to crack the NTLM hash if you know the character set. This way you do not need to bruteforce all possible characters combinations. Once you have TESTTEST, feed a custom character set of tesTES to john and it will return the proper case password much faster than if you did not limit the character set. The issue is john has no easy way to limit the character set. You will have to modify the john.conf file and include the following code that Solar Designer has kindly published to the john-users mail list:

```
[List.External:customcharset]
int running; // Are we already running?
int last; // Last character position, zero-based
int c0, c[0x100]; // Cyclic charset
void init()
{
  int length, cm, i;
  length = 10; // password length
  c[c0 = 't'] = 'e'; // change the t and the e to the first and second letters of the custom character set
  c['e'] = 's'; // change the e and the s to the second and third letters of the character set
  c['s'] = 'T'; // change the s and T to the third and fourth letters
  c['T'] = 'E'; // etc
  c['E'] = 'S'; // etc
  c[cm = 'S'] = c0; // change the S to the last letter of the character set

  // If you cannot see the pattern then do not bother with this trick.
  // If you can make the necessary changes to suit you environment.
  running = 0;
  last = length - 1;
  i = 0;
  while (i < length) word[i++] = cm; word[i] = 0;
}

void generate()
{
  int i;
  i = last;
  while ((word[i] = c[word[i]]) == c0)
    if (!i--) {
      if (running++) word = 0;
      return;
    }
}
```

Once you make the necessary changes begin cracking using:
/usr/local/john/john -external=customcharset --format:NT hash.txt

Some notes from Solar Designer:

1. Being an external mode, this is not the fastest way to generate candidate passwords, although its performance is acceptable. Some further optimizations are possible (e.g., cache the last character outside of the word[] array). Also, be careful when you edit it (such as for a different charset) - errors in the way the cyclic charset is defined may result in the "while" loop in generate() becoming endless.
2. In order to actually crack an NTLM hash with this, you need a build of JtR with support for NTLM hashes. You may do a custom build with the latest jumbo patch (john-1.7.2-all-9.diff.gz), which means that you will need to install Cygwin on your Windows system, or you can download such a build made by someone else (one is linked from the JtR homepage - it is for an older version of the patch, though, so it is many times slower at NTLM hashes).
3. On a modern system, with a recent jumbo patch, and with the proper "make" target for your system, this should complete its work against an NTLM hash (or against many such hashes) in just a few minutes.

3.3.1.1.4 Cracking cached credentials

john only needs to know the path to the hash.txt to begin bruteforcing and return the password
/usr/local/john/john --format:mscash hash.txt

3.3.1.2 John the Ripper - current

The current version of John the Ripper doesn't ship with BT4. It adds some new features (dumbforce and knownforce) and speeds up several algorithms. However given the way BT4 handles updates I don't recommend updating the package yourself unless your processor doesn't support SSE2 instructions (i.e. something less than a P4). I recommend going to http://www.backtrack-linux.org/forums/tool-requests/ and requesting they update the package to the latest version. Do NOT ask them to drop the SSE2 requirement. The SSE2 instructions provide real benefits to the cracking process. If you need to compile your own version here is how.

3.3.1.2.1 Get and Compile

We first have to remove the existing package and then we can download and compile the program.

1. open a terminal window
2. # dpkg -r john
3. # wget http://www.openwall.com/john/g/john-1.7.4.2.tar.bz2
4. goto ftp://ftp.openwall.com/john/contrib/
   and look for something like john-1.7.4.2-jumbo-1.diff.gz
   This is version 1 of the jumbo patch for john 1.7.4.2. Download the latest version that is there.
5. # tar -xvf john-1.7.4.2.tar.bz2
6. # cd john-1.7.4.2
7. # zcat ../john-1.7.4.2-jumbo-1.diff.gz | patch -p1 -Z
   You should see a long list of patching file XXX. If you see X out of Y hunk ignored it means that the patch did not apply correctly. You either downloaded the wrong version of john or the jumbo patch. Start over and make sure the jumbo patch matches the version of john you download.
8. # cd src
9. # make
   This will display the various systems you can compile john for. I have a P3 that supports MMX so I will use the command: make linux-x86-mmx. To see which options your CPU supports do a: cat

/proc/cpuinfo and look at the flags. If you have a P4 you probably have SEE2 (check the cpuinfo flags) then you would use: make linux-x86-see2.

10. # make linux-x86-mmx
    You should see the following when john is done compiling:
    gcc DES_fmt.o DES_std.o DES_bs.o BSDI_fmt.o MD5_fmt.o MD5_std.o MD5_apache_fmt.o BFEgg_fmt.o BF_fmt.o BF_std.o AFS_fmt.o LM_fmt.o NT_fmt.o XSHA_fmt.o DOMINOSEC_fmt.o lotus5_fmt.o oracle_fmt.o MYSQL_fmt.o mysqlSHA1_fmt.o KRB5_fmt.o KRB5_std.o md5_go.o rawMD5go_fmt.o md5_eq.o PO_fmt.o md5.o hmacmd5.o hmacMD5_fmt.o IPB2_fmt.o rawSHA1_fmt.o NSLDAP_fmt.o NSLDAPS_fmt.o OPENLDAPS_fmt.o base64.o md4.o smbencrypt.o mscash_fmt.o NETLM_fmt.o NETNTLM_fmt.o NETLMv2_fmt.o NETHALFLM_fmt.o mssql_fmt.o mssql05_fmt.o EPI_fmt.o PHPS_fmt.o MYSQL_fast_fmt.o pixMD5_fmt.o sapG_fmt.o sapB_fmt.o NS_fmt.o HDAA_fmt.o batch.o bench.o charset.o common.o compiler.o config.o cracker.o crc32.o external.o formats.o getopt.o idle.o inc.o john.o list.o loader.o logger.o math.o memory.o misc.o options.o params.o path.o recovery.o rpp.o rules.o signals.o single.o status.o tty.o wordlist.o mkv.o mkvlib.o unshadow.o unafs.o undrop.o unique.o x86.o x86-mmx.o sha1-mmx.o md5-mmx.o -s -L/usr/local/lib -L/usr/local/ssl/lib -lcrypto -lm -o ../run/john
    rm -f ../run/unshadow
    ln -s john ../run/unshadow
    rm -f ../run/unafs
    ln -s john ../run/unafs
    rm -f ../run/unique
    ln -s john ../run/unique
    rm -f ../run/undrop
    ln -s john ../run/undrop
    gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include -L/usr/local/lib -funroll-loops genmkvpwd.c
    gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include -L/usr/local/lib -funroll-loops -D_JOHN_MISC_NO_LOG misc.c -o miscnl.o
    gcc genmkvpwd.o mkvlib.o memory.o miscnl.o -s -lm -o ../run/genmkvpwd
    gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include -L/usr/local/lib -funroll-loops mkvcalcproba.c
    gcc mkvcalcproba.o -s -lm -o ../run/mkvcalcproba
    gcc -c -Wall -O2 -fomit-frame-pointer -I/usr/local/include -L/usr/local/lib -funroll-loops calc_stat.c
    gcc calc_stat.o -s -lm -o ../run/calc_stat
    make[1]: Leaving directory `/root/john-1.7.4.2/src'
11. # cd ..
12. # mv run /pentest/passwords/john

You now have the latest version of John the Ripper and it supports more algorithms than the vanilla John the Ripper thanks to the jumbo patch.

3.3.1.2.2 Cracking the LM hash

john only needs to know the path to the hash.txt to begin bruteforcing and return the uppercase password
/usr/local/john/john hash.txt

3.3.1.2.3 Cracking the LM hash using known letter(s) in known location(s) (knownforce)

I haven't figured out how to use this feature. John the Ripper is a very powerful tool however it is not very intuitive to use. I can point you to the John the Ripper wiki which has maillist excerpts cover how to use dumbforce and knownforce. The url is http://openwall.info/wiki/john/mailing-list-excerpts

3.3.1.2.4 Cracking the NTLM hash

john only needs to know the path to the hash.txt to begin cracking and return the password
/usr/local/john/john --format:NT hash.txt
will begin to bruteforce the NTLM hashes

3.3.1.2.5 Cracking the NTLM hash using the cracked LM hash (dumbforce)

I haven't figured out how to use this feature. John the Ripper is a very powerful tool however it is not very intuitive to use. I can point you to the John the Ripper wiki which has maillist excerpts cover how to use dumbforce and knownforce. The url is http://openwall.info/wiki/john/mailing-list-excerpts

3.3.1.2.6 Cracking cached credentials

john only needs to know the path to the hash.txt to begin bruteforcing and return the password
/usr/local/john/john --format:mscash hash.txt

3.3.1.3 Using MDCrack

For whatever reason I have been unsuccessful in getting mdcrack-183 to work with any version of wine. This is strange as I know I had it working previously. To use mdcrack with BackTrack you should upgrade wine to the latest development version of wine and then use mdcrack-182.zip

For BackTrack 3 users:

1. Goto http://www.winehq.org/site/download and click on slackware.
2. download the latest tgz file, which as of this writting is 1.1.29
3. open a xterm window
4. # upgradepkg wine-1.1.29-i486-1kjz.tgz
5. Now you can download mdcrack-182.zip download mdcrack
   # wget http://membres.lycos.fr/mdcrack/download/MDCrack-182.zip
   or
   # wget http://c3rb3r.openwall.net/mdcrack/download/MDCrack-182.zip
6. # mkdir mdcrack
7. # mv MDCrack-182.zip mdcrack
8. # cd mdcrack
9. # unzip MDCrack-182.zip

For BackTrack 4 users:

1. open a xterm window
2. # wget -q http://wine.budgetdedicated.com/apt/387EE263.gpg -O- | sudo apt-key add -
3. # sudo wget http://wine.budgetdedicated.com/apt/sources.list.d/jaunty.list -O /etc/apt/sources.list.d /winehq.list
4. # sudo apt-get update
5. # sudo apt-get install wine
6. Now you can download mdcrack-182.zip download mdcrack
   # wget http://membres.lycos.fr/mdcrack/download/MDCrack-182.zip
   or
   # wget http://c3rb3r.openwall.net/mdcrack/download/MDCrack-182.zip
7. # mkdir mdcrack

8. # mv MDCrack-182.zip mdcrack
9. # cd mdcrack
10. # unzip MDCrack-182.zip

3.3.1.3.1 Cracking the LM hash

MDCrack doesn't crack LM hashes.

3.3.1.3.2 Cracking the NTLM hash

wine MDCrack-sse.exe --algorithm=NTLM1 NTLMHASH
NTLMHASH would be D280553F0103F2E643406517296E7582 for example

The result should be TestTest
The only way to speed up cracking is to know the minimum length of the password and use --minsize= to specify it.

3.3.1.3.3 Cracking the NTLM hash using the cracked LM hash

Stasik told me it is much easier to crack the NTLM hash if you know the character set. This way you do not need to bruteforce all possible characters combinations. Once you have TESTTEST, feed a custom character set of tesTES to mdcrack and it will return the proper case password much faster than if you did not limit the character set.

wine MDCrack-sse.exe --charset=tesTES --algorithm=NTLM1 D280553F0103F2E643406517296E7582

If you know the password length you can use:

wine MDCrack-sse.exe --charset=tesTES --algorithm=NTLM1 --minsize=8 --maxsize=8 D280553F0103F2E643406517296E7582

The password is TestTest however mdcrack 1.8.3 returns sestTest. I filed a bug report with Gregory Duchemin, the author of mdcrack, and he has fixed the problem with version 1.8.4.

3.3.1.4 Using Ophcrack

3.3.1.4.1 Cracking the LM hash

download ophcrack and the rainbow tables from http://sourceforge.net/project/showfiles.php?group_id=133599. If you have the hard drive space I would recommend downloading XP free fast formally known as SSTIC04-5K. If this is a demo or do not have a lot of disk space download XP free small formally known as SSTIC04-10K. This is not a typo; SSTIC04-5K is a larger download than SSTIC04-10K. You can also purchase the XP Special table which contain longer passwords and the special characters. There are special tables for Vista. The small table is Vista free and is free. There is a table you can purchase called Vista Special which contains hashes for passwords up to 8 characters. See http://ophcrack.sourceforge.net/tables.php for the details. The rainbow tables that ophcrack uses are NOT compatible with the rainbow tables generated by rtgen.

1. # tar -xvzf ophcrack-2.4.1.tar.gz
2. # cd ophcrack-2.4.1
3. # ./configure

4. # make
5. # make install
6. # ophcrack
7. click on the load button and select the appropriate option, I will select local SAM.
8. click on the tables button and select the rainbow table you installed.
9. click on the launch button. You will see pre-loading table boxes on the screen. You may also see a message that says "All LM hashes are empty. Please use NThash tables to crack the remaining hashes." This means that the administrators have disabled windows ability to save LM hashes.
10. wait until ophcrack is done

3.3.1.4.2 Cracking the NTLM hash

You will have to purchase the NTLM rainbow tables from http://www.objectif-securite.ch/en/products.php. The rainbow table contains 99% of passwords of made of following characters:

length 1 to 6:
0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"#$%&'()*+,-./:;&<=>?@[\]^_`{|}~ (space included)

length 7:
0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

length 8:
0123456789abcdefghijklmnopqrstuvwxyz

You CANNOT generate your own rainbow tables for ophcrack to use. If you know that the password meets the above specs you can purchase the table and give it a try.

3.3.1.4.3 Cracking the NTLM hash using the cracked LM hash

There is no way to do this. If ophcrack cracks the LM hash you should switch to john or mdcrack to get the NTLM password.

3.3.2 Using Windows Tools

3.3.2.1 John the Ripper

3.3.2.1.1 Cracking the LM hash

1. download john the ripper from http://www.openwall.com/john/
2. open a command prompt (Start, Run, cmd, enter)
3. cd to where you extracted john (I extracted john to the root of my C drive) so it would be cd \john171w\run
   C:\> cd \john171w\run
4. john only needs to know the path to the hash.txt to begin bruteforcing and return the uppercase password
   C:\> john-386 C:\hash.txt

3.3.2.1.2 Cracking the NTLM hash

You cannot as the windows binary of john the ripper that you can download from the website does not support NTLM. You will have to download the source code, one of the patches that adds support for NTLM and compile it yourself. There are also one or two places where you can download a john binary that already has the patches applied.

3.3.2.1.3 Cracking the NTLM hash using the cracked LM hash

You cannot as the windows binary of john the ripper that you can download from the website does not support NTLM. You will have to download the source code, one of the patches that adds support for NTLM and compile it yourself. There are also one or two places where you can download a john binary that already has the patches applied.

3.3.2.2 MDCrack

3.3.2.2.1 Cracking the LM hash

MDCrack doesn't support LM hashes.

3.3.2.2.2 Cracking the NTLM hash

1. download mdcrack from http://membres.lycos.fr/mdcrack/ or http://c3rb3r.openwall.net/mdcrack/ and extract the files
2. open a command prompt (Start, Run, cmd, enter)
3. cd to where you extracted the files C:\> cd \mdcrack-183
4. C:\MDCrack-183> MDCrack-sse.exe --algorithm=NTLM1 NTLMHASH
   NTLMHASH would be D280553F0103F2E643406517296E7582 for example

The result should be TestTest

3.3.2.2.3 Cracking the NTLM hash using the cracked LM hash

Stasik told me it is much easier to crack the NTLM hash if you know the character set. This way you do not need to bruteforce all possible characters combinations. Once you have TESTTEST, feed a custom character set of tesTES to mdcrack and it will return the proper case password much faster than if you did not limit the character set.

C:\MDCrack-183> MDCrack-sse.exe --charset=tesTES --algorithm=NTLM1
D280553F0103F2E643406517296E7582

If you know the password length you can use:

C:\MDCrack-183> MDCrack-sse.exe --charset=tesTES --algorithm=NTLM1 --minsize=8 --maxsize=8
D280553F0103F2E643406517296E7582

The password is TestTest

3.3.2.3 Ophcrack

3.3.2.3.1 Cracking the LM hash

1. download ophcrack from http://sourceforge.net/project/showfiles.php?group_id=133599
2. start the installation and select next until you get to the Select Components screen. Select one of the two download options. If you have the hard drive space I would recommend downloading SSTIC04-5K. If this is for a demo or do not have a lot of disk space download SSTIC04-10K. This is not a typo; SSTIC04-5K is a larger download than SSTIC04-10K. Answer the rest of the install questions and click on install. The installer will start downloading the rainbow table you selected. The other options on the Select Components screen will cost you money as you have to purchase the DVD or CD. The rainbow tables that ophcrack uses are NOT compatible with the rainbow tables generated by rtgen.
3. start ophcrack using the icon on the start menu bar.
4. click on the load button and select the appropriate option, I will select local SAM.
5. click on the tables button and select the rainbow table you installed.
6. click on the launch button. You will see pre-loading table boxes on the screen. You may also see a message that says "All LM hashes are empty. Please use NThash tables to crack the remaining hashes." This means that the administrators have disabled windows ability to save LM hashes.
7. wait until ophcrack is done

3.3.2.3.2 Cracking the NTLM hash

You will have to purchase the NTLM rainbow tables from http://www.objectif-securite.ch/en/products.php. The rainbow table contains 99% of passwords of made of following characters sets:

length 1 to 6:
0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"#$%&'()*+,-./:;&<=>?@[\]^_`{|}~ (space included)

length 7:
0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

length 8:
0123456789abcdefghijklmnopqrstuvwxyz

You CANNOT generate your own rainbow tables for ophcrack to use. If you know that the password meets the above specs you can purchase the table and give it a try.

3.3.2.3.3 Cracking the NTLM hash using the cracked LM hash

There is no way to do this. If opcrack cracks the LM hash you should switch to john or mdcrack to get the NTLM password.

3.3.2.4 Using Cain and Abel

1. Download Cain and Abel from http://www.oxid.it/cain.html
2. Install Cain and Abel using the default settings
3. Start Cain and Abel
4. Click on the Cracker tab
5. Click somewhere inside the table
6. Click on File, Add to list.
7. Select Import hashes from local system and click next
8. Right click on the account you want the password for and select Brute-force attack. Chose the option you want to use (for windows passwords either LM hashes or NTLM hashes).
9. Select the character set you want to use, set the minimum and/or maximum password length if you

know it to decrease the amount of cracking time needed.
10. Click on Start and wait

3.3.3 Using a Live CD

3.3.3.1 Ophcrack

The ophcrack LiveCD is good when you have physical access to the PC. Just download ophcrack-livecd from http://sourceforge.net/project/showfiles.php?group_id=133599, burn the iso to a CD, boot from the CD, and start cracking using the included SSTIC04-10K rainbow table. This will crack the LM hashes. To crack NTLM hashes you have to purchase 1 of the NTLM hash tables. See section 3.2.2.3.2 for details.

4 Changing Windows Passwords

4.1 Changing Local Windows Passwords

Warning: If the user account you are going to change the password on has encrypted files you will lose access to the encrypted files until you remember the old password. This only applies to local user accounts and not domain accounts. In a properly setup forest/domain, the administrator can recover encrypted files for domain users.

4.1.1 Using BackTrack tools

4.1.1.1 chntpw

chntpw is a part of BackTrack and can reset a forgotten local user account password. If you don't have BackTrack you can download a CD or floppy disk image from http://pogostick.net/~pnh/ntpasswd/, unzip the file, and either burn the iso the to a CD or use rawrite2 to transfer the image to a floppy. Follow the directions at http://pogostick.net/~pnh/ntpasswd/walkthrough.html on how to use the software. The current version supports Vista.

1. # mount -t ntfs-3g /dev/XXX /mnt/XXX
2. # cd /pentest/password/chntpw
3. # chntpw

```
chntpw version 0.99.5 070923 (decade), (c) Petter N Hagen
chntpw: change password of a user in a NT/2k/XP/2k3/Vista SAM file, or invoke registry editor.
chntpw [OPTIONS] <samfile> [systemfile] [securityfile] [otherreghive] [...]
-h This message
-u <user> Username to change, Administrator is default
-l list all users in SAM file
-i Interactive. List users (as -l) then ask for username to change
-e Registry editor. Now with full write support!
-d Enter buffer debugger instead (hex editor),
-t Trace. Show hexdump of structs/segments. (deprecated debug function)
-v Be a little more verbose (for debuging)
-L Write names of changed files to /tmp/changed
-N No allocation mode. Only (old style) same length overwrites possible
See readme file on how to get to the registry files, and what they are.
Source/binary freely distributable under GPL v2 license. See README for details.
```

NOTE: This program is somewhat hackish! You are on your own!

4. # chntpw /mnt/sda1/WINDOWS/system32/config/sam /mnt/sda1/WINDOWS/system32/config/system /mnt/sda1/WINDOWS/system32/config/security -u test

   chntpw version 0.99.5 070923 (decade), (c) Petter N Hagen
   Hive </mnt/sda1/WINDOWS/system32/config/sam> name (from header): <\SystemRoot\System32 \Config\SAM>
   ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c <lf>
   Page at 0x7000 is not 'hbin', assuming file contains garbage at end
   File size 262144 [40000] bytes, containing 6 pages (+ 1 headerpage)
   Used for data: 240/19064 blocks/bytes, unused: 15/5320 blocks/bytes.

   Hive </mnt/sda1/WINDOWS/system32/config/system> name (from header): <SYSTEM>
   ROOT KEY at offset: 0x001020 * Subkey indexing type is: 686c <lh>
   Page at 0x5c4000 is not 'hbin', assuming file contains garbage at end
   File size 6291456 [600000] bytes, containing 1366 pages (+ 1 headerpage)
   Used for data: 91697/4664232 blocks/bytes, unused: 3600/1333656 blocks/bytes.

   Hive </mnt/sda1/WINDOWS/system32/config/security> name (from header): <emRoot\System32 \Config\SECURITY>
   ROOT KEY at offset: 0x001020 * Subkey indexing type is: 666c <lf>
   Page at 0xc000 is not 'hbin', assuming file contains garbage at end
   File size 262144 [40000] bytes, containing 11 pages (+ 1 headerpage)
   Used for data: 879/40312 blocks/bytes, unused: 11/4392 blocks/bytes.

   * SAM policy limits:
   Failed logins before lockout is: 0
   Minimum password length : 0
   Password history count : 0

RID	---------- Username ----------	Admin?	Lock?
01f4	Administrator	ADMIN	
03fc	test		
01f5	Guest		dis/lock
03e8	HelpAssistant		dis/lock
03ea	SUPPORT_388945a0		dis/lock

----------> SYSKEY CHECK <----------
SYSTEM SecureBoot : 1 -> key-in-registry
SAM Account\F : 1 -> key-in-registry
SECURITY PolSecretEncryptionKey: 1 -> key-in-registry

********** SYSKEY IS ENABLED! **********
This installation very likely has the syskey passwordhash-obfuscator installed
It's currently in mode = 1, key-in-registry-mode
SYSKEY is on! However, DO NOT DISABLE IT UNLESS YOU HAVE TO!
This program can change passwords even if syskey is on, however
if you have lost the key-floppy or passphrase you can turn it off,
but please read the docs first!!!

** IF YOU DON'T KNOW WHAT SYSKEY IS YOU DO NOT NEED TO SWITCH IT OFF!**
NOTE: On WINDOWS 2000 it will not be possible
to turn it on again! (and other problems may also show..)

NOTE: Disabling syskey will invalidate ALL
passwords, requiring them to be reset. You should at least reset the
administrator password using this program, then the rest ought to be
done from NT.

Do you really wish to disable SYSKEY? (y/n) [n] n

RID : 1020 [03fc]
Username: test
fullname: test
comment :
homedir :

User is member of 1 groups:
00000221 = Users (which has 3 members)

Account bits: 0x0010 =
[ ] Disabled | [ ] Homedir req. | [ ] Passwd not req. |
[ ] Temp. duplicate | [X] Normal account | [ ] NMS account |
[ ] Domain trust ac | [ ] Wks trust act. | [ ] Srv trust act |
[ ] Pwd don't expir | [ ] Auto lockout | [ ] (unknown 0x08) |
[ ] (unknown 0x10) | [ ] (unknown 0x20) | [ ] (unknown 0x40) |

Failed login count: 0, while max tries is: 0
Total login count: 0

- - - - User Edit Menu:
1 - Clear (blank) user password
2 - Edit (set new) user password (careful with this on XP or Vista)
3 - Promote user (make user an administrator)
(4 - Unlock and enable user account) [seems unlocked already]
q - Quit editing user, back to user select
Select: [q] > 2
New Password: TestTest
Password changed!

Hives that have changed:
Name
0 </mnt/sda1/WINDOWS/system32/config/sam>
Write hive files? (y/n) [n] : y
0 </mnt/sda1/WINDOWS/system32/config/sam> - OK

The password for the user account test has been set to TestTest. Rarely chntpw screws up and the password is not reset properly. Your best bet is to blank the password (option 1).

4.1.2 Using a Live CD

4.1.2.1 chntpw

chntpw comes in two versions: floppy and cd iso. The floppy version only contains drivers for the more popular hard drive controllers. Since the number of drivers is limited, the floppy version will not boot your PC if you have a controller that is not supported. Also floppy drives are becoming hard to find on PCs. I recommend you download the iso as it contains all possible drivers and CD/DVD drives are more common than floppy drives these days. You can download either version from http://pogostick.net/~pnh/ntpasswd/ and click on bootdisk. Just below the download links are instructions on what to do with the file you downloaded. The iso is approximately 3MB in size. This Live CD only contains the drivers necessary to find your hard drives and the chntpw command. This is a great tool when all you need to do a change a password. The current version supports Vista.

4.1.2.2 System Rescue CD

The System Rescue CD is a Linux bootable CD you can use to repair your PC and recover data after a crash. You can download the iso from http://www.sysresccd.org/ The iso is almost 250MB for version 1.1 for x86. The website has some really good documentation on how to use the cd. To change a local user password with the System Rescue CD, you use the chntpw command.

4.2 Changing Active Directory Passwords

The pwdump7 utility only work on locally stored SAMs. It will not work on windows server 2000, 2003, or windows 2008 Active Directory. pwdump6 from http://www.foofus.net/fizzgig/pwdump/ can dump windows server 2000 and 2003 Active Directory to a file that john the ripper can crack. fgdump is a newer utility that can dump Active Directory and cached credentials. See section 3.1.2 on how to use fgdump

See http://www.jms1.net/nt-unlock.shtml for resetting a windows 2000 domain account.
See http://www.nobodix.org/seb/win2003_adminpass.html for resetting a windows 2003 domain account.

5 plain-text.info

The http://plain-text.info website is where you can take a LM or MD5 hash and see if someone has already cracked it. The website is easy to use and has a very large list of hashes for you to search. The website also has support for NTLM hashes, but none of the hashes I had were cracked there.

6 Cracking Novell NetWare Passwords

If you have a windows client with the Novell NDS client installed you will notice that the client creates a local account on the PC. Usually the local password is the same as the NDS account password. By dumping the local SAM you can usually get a persons NDS password. The above trick I know works as I have used this once. The other techniques listed here I cannot say if they work or not as I retired the last Novell NetWare server a long time ago.

I would recommend starting to read the documentation at http://www.nmrc.org/pub/faq/hackfaq/hackfaq-19.html to hackfaq-26.html for learning quite abit about attacking Novell servers v2.x to 4.x.

If you have access to the console and the floppy drive of the server you can try one of the console attacks from http://www.nmrc.org/pub/faq/hackfaq/hackfaq-21.html.

If all you have access to is a client you can try some of the client attacks from http://www.nmrc.org/pub/faq/hackfaq/hackfaq-22.html

If you don't have local administrator rights and your server is Novell you can watch the video at http://youtube.com/watch?v=GEl-CsUOY6A and get local administrator rights to your XP box. The attack is very simple you just have to pull your network cable at the right time. UPDATE: The user has removed the video from youtube.

You can use Pandora from http://www.nmrc.org/project/pandora/ to do online and offline attack against Novell 4.x and 5.x servers.

From williamc's posting of at the old remote-exploit forums:
"I confirmed an interesting vulnerability with Novell. According to this advisory, the Novell password can be dumped out of memory: http://www.securityfocus.com/archive/1/402767
We confirmed this on a Novell 7 environment while performing a pentest.
Use both psexec and pmdump as follows:
psexec \\hostname -u username -p password -s -f -c pmdump -list

Find the PID of the Gwise.exe service. Then:
psexec \\hostname -u -p password -s -f -c pmdump PID PID_dump.txt

This will dump the memory to \\hostname\c$\windows\system32
Open the dump file in a hex editor and search for an organizational string, like an OU in the targets memory dump file.

From there you can find the Novell password for the user within the file.

To simplify this, if you have already found one Novell password, dump the PID for gwise of that user and search for the password. From there, you can work backwards to find the OU format, and apply it to other targets. For instance, at this particular location, the client's OU is similar to clientname.AA_FINANCE.AA_NW

If you do a search for AA_FINANCE.AA_NW in the memory dump you will see the password in plain text. Suppose the CFO is whomever.AA_FINANCE.AA_NW. By dumping his memory, you can search for the OU and reference your original dump, match up the location and password."

7 Cracking Linux/Unix passwords

Unix passwords are located in the /etc/shadow file or the /etc/passwd file. Next unix passwords are "shadowed" and we must "unshadow" them before we can do anything. Without getting into too much detail shadowing the passwords removes the passwords, which are usually stored in world readable /etc/passwd, and moves them to /etc/shadow which can only be read and written to by root or programs run as suid root. Goto http://www.tldp.org/HOWTO/Shadow-Password-HOWTO-2.html if you want details on how shadowing works. Finally unix passwords are salted. The short version is a salt is randomly generated value that is used to encode the user's password, which is usually already encrypted; thus adding another layer of security. The long version can be found at http://www.tldp.org/HOWTO/Shadow-Password-HOWTO-2.html

1. boot BackTrack and login as root
2. # cd /usr/local/john
3. # unshadow /etc/passwd /etc/shadow >saltedpasswords
   You have now written the unshadowed "salted" passwords to a file called saltedpasswords
4. # john saltedpasswords

The above steps will crack the passwd file on the CD. If you want to crack a passwd file located on a hard drive:

1. boot BackTrack and login as root
2. # mount /dev/hda1 /mnt/XXX
   mount your Linux partition substituting hda1 for whatever your Linux partition is
3. # cd /usr/local/john
4. # unshadow /mnt/XXX/etc/passwd /mnt/XXX/etc/shadow >saltedpasswords
5. # john saltedpasswords

8 Cracking networking equipment passwords

Your first step should be to try the default username and password. You can find a list over 3500 devices at http://artofhacking.com/etc/passwd.htm.

8.1 Using BackTrack Tools

8.1.1 Using Hydra

Say you have wireless router to which you have forgotten the password. The easiest thing to do in this case is to reset the router to factory defaults. However if you have a lot of custom settings and your backup is nonexistent, out-dated, corrupted, or the backup restored a password which you do not remember, you can try a brute force attack on the router. From http://freeworld.thc.org/thc-hydra/. Hydra currently supports: TELNET, FTP, HTTP-GET, HTTP-HEAD, HTTPS-GET, HTTPS-HEAD, HTTP-PROXY, LDAP2, LADP3, SMB, SMBNT, MS-SQL, MYSQL, POSTGRES, REXEC, RSH, RLOGIN, CVS, SNMP, SMTP-AUTH, SOCKS5, VNC, POP3, IMAP, NNTP, PCNFS, ICQ, SAP/R3, SSH2, Teamspeak, Cisco auth, Cisco enable, Cisco AAA (incorporated in telnet module)

For the sake of this document I will use a Linksys WRT54GL, hardware v1.1, Linksys firmware 4.30.11 and dd-wrt v2.4sp1. The first thing you have to do is find out if the device uses a username AND password to login. Several devices only require a password to login into the device as admin, root, system, etc. The easiest way to find this out is to goto the vendor's website and download installation manual which will give you this information. For the Linksys firmware

1. boot BackTrack and login as root
2. hydra -l "" -P wordlist.txt -f -v -e ns 192.168.1.1 http-get /
   -l is for a username which is null in this case
   -P is a wordlist of passwords to try
   -f stop hydra when it finds the password
   -v is for verbose
   -e try no password and password the ip address of the device one of the currently supported options
   192.168.1.1 is the IP of the AP
   http-get is the correct option to this AP
   / is where you have to put in the username and password. In this case you have to put in the username and password before you can do anything. You will have to figure this out for yourself as each device is different.
3. some other option you may need are:
   -t TASKS run TASKS number of connects in parallel (default: 16)
   You may have to adjust this number down as larger numbers may cause the router to crash or misbehave. I usually use 10.
   -v / -V verbose mode / show login+pass combination for each attempt

As I said earlier, http-get is the correct option for my Linksys AP (with the factory firmware). When you goto the AP website a dialog box opens and prompts for a username and password. Each AP is different and as

such you will have to change options as RaginRob found out. The following is a slightly modified version from his tutorial.

I recently started playing around with Hydra and tried to hack my router. After searching the forum and googling around a while I noticed that there are only some howto's for routers that have http-auth authentication. That is, when you go to 192.168.2.1 e.g. and before showing anything you have to enter login and password in a popup. My router (T-Com Sinus 154 DSL Basic 3) and many others I've dealt with so far work differently. When I want to login to my router, I have to go to 192.168.2.1, a web interface with a password field shows up, and I have to enter the password which is then checked by /cgi-bin/login.exe via http-post.

It was quite tricky to find out how to use this authentication with hydra, so I guess there are some of you that can benefit from this. I'll describe how I did it, so you can adapt the method and use it with your own router.

First of all I examined the login page of the web interface. Be sure to look at the frame source and not the frameset. You should see the form and the action, here's what I saw:

The form is defined as:

```
<form name="tF" method="post" action="/cgi-bin/login.exe" onSubmit="evaltF();">
```

Somewhere in the form there will be the field that takes the password:

```
<input type="password" name="pws" class="stylepwd" size="12" maxlength="12">
```

This is probably the most important data you need. You need to write down the field name ("pws" in my case). The size attribute comes in very handy too because it tells us that the password's max length is 12 characters.

After that I tried to get familiar with Hydra's options. I figured out that you need the following options:

hydra -l "" -P passwords.txt -t 1 -f -v -V 192.168.2.1 http-post-form /cgi-bin
/login.exe:pws=^PASS^:loginpserr.htm
-l Sets the login name. In the end I don't need a login name but hydra gets kind of pissed when you don't pass something, so I gave an empty string.
-P The wordlist to use for the password
-t 1 task only, not really necessary, I just wanted to make sure Hydra doesn't choke on too many requests -f Hydra shall stop when a working password is found
-v be verbose. and even more. I skipped that in the final version but it's ok for debugging
192.168.1.1 is victim's ip
http-post-form the method to use
/cgi-bin/login.exe:pws=^PASS^:loginpserr.htm
This is the most important part. Here we tell Hydra what to pass the passwords to. The argument consists of three parts separated by ":".

The first part is the script that takes the POST data, we found that in the frame source above.

The second part is the field name of the password field with an added =^PASS^. ^PASS^ is the variable that hydra substitutes with the passwords in the wordlist.

The third part is the "incorrect" condition. Hydra has to find out somehow if the current password that was send to the router is correct or not. You have to find a string that is actually IN A NEGATIVE RESPONSE from the router. As we don't have the password yet we can't know what the router will send if the password is correct, therefore we have to check if it is NOT, which we can find out easily. To find out what the router sends back to hydra I used Wireshark.

Open up wireshark, go to the router login page, start capturing and then login with a wrong password. After that, stop capturing and apply a "http" filter. You will see the POST data sent from hydra to the router (you should also see the "pws=blabla" in the details, that's where hydra sends the passwords from the wordlist). Below that you'll find the router answer. In my case it says something like "This page has moved to loginpserr.htm" packed in some basic HTML. So I used the string loginpserr.htm to validate the .. uhm... faultyness. OMFG %-]

Hydra will consider a password as CORRECT when the router answer DOES NOT contain the given string. So be sure to take an expression that somehow sounds like "incorrect" or "wrong". If you took "the" for example, and the POSITVE response would be something like "the password you entered was correct", hydra will not recognize it as correct but incorrect.

If your router does not only need a password but also a username, you can easily add the according login name to the last part. So if you need to send the field "login" or whatever it is called in your case with the value "admin" as the only username you could use

/cgi-bin/login.exe:login=admin&pws=^PASS^:loginpserr.htm

When you need to try a whole username list then you can specify the list via
-L usernames.txt
and
/cgi-bin/login.exe:login=^USER^&pws=^PASS^:loginpserr.htm

For dd-wrt do

1. boot BackTrack and login as root
2. hydra -l admin -P wordlist.txt -f -e ns 192.168.1.1 http-get /login.asp
   -l is for a username which is admin in this case. dd-wrt allows the user to chose the username that is require to login to the device so it could be anything.
   -P is a wordlist of passwords to try
   -f stop hydra when it finds the password
   -v is for verbose
   -e try no password and password the ip address of the device one of the currently supported options
   192.168.1.1 is the IP of the AP
   http-get is the correct option to this AP
   / is where you have to put in the username and password. In this case you have to put in the username and password before you can do anything. You will have to figure this out for yourself as each device is different.
3. some other option you may need are:
   -t TASKS run TASKS number of connects in parallel (default: 16)
   You may have to adjust this number down as larger numbers may cause the router to crash or misbehave. I usually use 10.
   -v / -V verbose mode / show login+pass combination for each attempt

8.1.2 Using Xhydra

Xhydra is Hydra with a GUI. This graphical interface is a good place for newbies to start practicing and then move to the command line. Pureh@te has made a video that demonstrates how to use Xhydra to bruteforce the password on a Wireless AP. You can watch the video at http://blip.tv/file/522320. I had to update my flash player so the video wasn't choppy. To update the flashplayer in BT4 do:
apt-get remove flashplugin-nonfree
wget http://fpdownload.macromedia.com/get/flashplayer/current/install_flash_player_10_linux.tar.gz &&

tar xzf install_flash_player_10_linux.tar.gz && mkdir ~/.mozilla/plugins && mv libflashplayer.so ~/.mozilla/plugins

If you have problems with your attack after watching the video try decreasing the number of tasks to 10.

8.1.3 Using Medusa

Medusa is like hydra in that it is a login bruteforcer. So what are there differences? You can see for yourself at http://www.foofus.net/jmk/medusa/medusa-compare.html
The short version medusa supports somethings while hydra doesn't and vice versa. For example. If you need to crack ncpfs you need medusa. If you need ICQ you need hydra. There are more differences so please read the above webpage.

The following command is used to attack my D-Link DI-808HV DSL router.
medusa -h 172.19.0.1 -u "admin" -P wordlist.txt -M http
-h is the host you want to attack
-u is the username you want to try. If there is no username use ""
-P is the wordlist you want to use
-M is the module you want to use

Here is the list of modules included with BackTrack 3 Final:
ls /usr/local/lib/medusa/modules/

cvs.mod	mssql.mod	pop3.mod	rsh.mod	snmp.mod	vmauthd.mod
ftp.mod	mysql.mod	postgres.mod	smbnt.mod	ssh.mod	vnc.mod
http.mod	nntp.mod	rexec.mod	smtp-auth.mod	svn.mod	web-form.mod
imap.mod	pcanywhere.mod	rlogin.mod	smtp-vrfy.mod	telnet.mod	wrapper.mod

Here is the list of modules included with BackTrack 4 Final:
ls /usr/lib/medusa/modules/

cvs.mod	mssql.mod	pop3.mod	smbnt.mod	ssh.mod	web-form.mod
ftp.mod	mysql.mod	rexec.mod	smtp-vrfy.mod	telnet.mod	wrapper.mod
http.mod	nntp.mod	rlogin.mod	smtp.mod	vmauthd.mod	
imap.mod	pcanywhere.mod	rsh.mod	smtp.mod	vnc.mod	

Be sure to leave .mod off of the command line. The documentation at http://www.foofus.net/~jmk/medusa/medusa.html is lacking.

8.1.4 Using John the Ripper to crack a Cisco hash

there are two passwords in Cisco equipment: enable and telnet. Within the enable command there are two ways to store the password
1. enable password (aka type 7)
2. enable secret (aka type 5)

enable password is the original encryption algorithm used by Cisco. The algorithm was not released to the general public, however the algorithm was figured out and at least two decryption programs are available on the internet. The source code to decrypt a type 7 password is available on the internet. enable password is a very weak algorithm and should not be used.

enable secret is the replacement for enable password. enable secret takes the password and hashes it using the md5 hash. Hashes are one way algorithms. It is currently not possible to recover the password given the hash. You have to use a dictionary or brute force attack i.e. take every word in your dictionary, md5 hash it, and compare the dictionary hash to the cisco hash.

Once you have logged on to the router or switch you have to display the running configuration to get the hash. Type the following at the enable prompt:
show running-config

Look for enable secret 5 1sz0o$PYahL33gyTuHm9a8/UfmC1 in the display. This is a md5 hashed password. If you see: enable password 7 03003E2E05077C4F4007 then the password is stored in the weak password encryption.

Once you have the hash use a text editor and put the hash into the following format and save it a chash:
enable:1sz0o$PYahL33gyTuHm9a8/UfmC1:::::
Now you can use john to crack it
/usr/local/john/john /path/to/chash

8.2 Using Windows Tools

8.2.1 Using Brutus

Brutus is a windows application that does things similar to Hydra. You can find it at http://www.hoobie.net/brutus/

9 Cracking Applications

9.1 Cracking Oracle 11g (sha1)

I cannot do any better than sending you to http://freeworld.thc.org/thc-orakelcrackert11g/

9.2 Cracking Oracle passwords over the wire

Again The Hacker's Choice at http://freeworld.thc.org/thc-orakel/ has great documentation on this

9.3 Cracking Office passwords

If you need to remove a password from an excel spreadsheet (workbook or sheet) the best I found is Excel Password Remover 2009 which can be downloaded from http://www.straxx.com/excel/password.html

Instructions from the straxx website on how to use Excel Password Remover 2009:
To load this add-in in Excel: Open it the same way you do with Excel workbooks. Installing the add-in does not work for all users.
Loading this add-in in Excel gives you two extra menu-items on the "Tools"-menu (or equivalent in non-English versions of Excel). These are:
1. Unprotect workbook
2. Unprotect sheet

ElcomSoft has Advanced Office Password Breaker and Advanced Office Password Recovery. You can read about the differences between at http://blog.crackpassword.com/2009/05/frequently-asked-question-

advanced-office-password-recovery-or-advanced-office-password-breaker/. The short version is breaker only works on word and excel documents from 2003 and before. For anything else you need Advanced Office Password Recovery and it can take quite awhile to recover the password.

Goto http://www.gmayor.com/Remove_Password.htm for instructions on how to remove the password from Office documents.

9.4 Cracking rar passwords

The only free software to crack rar passwords is from http://www.crark.net. Version 3.2a adds CUDA support for Nvidia video card to increase cracking speed by up 15x. The source code is at http://www.crark.net/Cuda-OpenSrc.rar and binaries are available.

Next there is rarcrack. You can download the source code from http://sourceforge.net/projects/rarcrack/. You can read how to use rarcrack at http://www.mydigitallife.info/2009/01/06/how-to-recover-rar-7z-and-zip-password-with-rarcrack-in-linux/

You can also use Advanced Archive Password Recovery from http://www.elcomsoft.com/archpr.html to recover the password. Advanced Archive Password Recovery is not free. The advantage to Advanced Archive Password Recovery is that it supports a wide range of encryption and compression algorithms.

Finally cracking the password takes time. If you really need into that file you can just remove the password using winrar_unlock from http://www.mydigitallife.info/2008/11/30/winrar-unlock-free-download-to-unlock-locked-protected-rar-archive-support-sfx-exe/

9.5 Cracking zip passwords

You can use rarcrack to crack zip and 7z files. The source code is available from http://sourceforge.net/projects/rarcrack/.

fcrackzip is another tool to crack zip passwords. You can download the source code and windows binary from http://home.schmorp.de/marc/fcrackzip.html

Finally cracking the password takes time. If you really need into that file you should just try to remove the password.

9.6 Cracking pdf passwords

There are two types of pdf passwords user and owner. A user password is set when the author of the pdf wants the end user to enter a password to view the pdf. In other words the password has to be entered each time the pdf is opened. The owner password is set when the author wants to prevent the person viewing the pdf from changing the permissions of the restricted features. If both passwords are set either password can open the document. However only the owner password can change permissions.

You can use PDFCrack to crack password protected pdf files. The source code is available from http://pdfcrack.sourceforge.net/. PDF Crack is made for Linux however it should work for any POSIX compatible system. rubypdf created a windows binary of PDFCrack using Cygwin. You can read about it at http://sourceforge.net/projects/pdfcrack/forums/forum/575585/topic/2224035. If your computer has a multi-core processor you might want to apply the following patch to PDFCrack http://sourceforge.net/projects/pdfcrack/forums/forum/575585/topic/3364643. This patch adds support for multi-core processors and speeds up the cracking process and yes there is a windows binary available.

GuaPDF available from http://www.guapdf.com/ can also crack password protected pdf files. GuaPDF can remove the restrictions of any password protected pdf. This includes support for Adobe 9 with 256 bit AES and 128 bit RC4 encryption. The removal process is instantanous. Note that only standard pdf security is supported. Cracking of third-party plugins and/or e-books is not supported. GuaPDF is supported on Linux, Windows, Solaris, DOS, and Mac OS. It supports multi-core processors and NVIDIA GPU cards. GuaPDF will remove restrictions from small pdf files for free. For larger files or a version that supports distributed cracking the program you must pay for the program.

Elcomsoft also makes a pdf password recovery program. It can be found at http://www.elcomsoft.com/apdfpr.html. Advanced PDF Password Recovery can remove both the user and owner passwords as well as remove any restrictions.

You can also try to open the PDF and print it to another pdf creater to remove the restrictions.

Finally cracking the password takes time. If you really need into that file you should just try to remove the password.

10 Wordlists aka Dictionary attack

Strictly speaking a wordlist is a file that contains words or phrases from everyday life. The premade files below are wordlists. The other instructions in this section are not really wordlists as the resulting file every possible combination (depending on the options you give the program) of characters. I have not come across a word to describe these files so I am naming a file that contains every possible combination of characters a combination list. Not very original but it does clearly differentiate between the two types of lists. The resulting combination list can be used in place of a wordlist. Please note that when you generate a combination list the file will be huge. Using the 95 English characters to generate every possible combination of 10 characters the resulting file size will be 95^10 =
59,873,693,923,837,900,000 bytes
59,873,693,923,837,900 KB
59,873,693,923,838 MB
59,873,693,924 GB
59,873,694 TB
59,874 PB
60 EB

For 14 characters the file size would be 4,877 YB

10.1 Using John the Ripper to generate a wordlist

For Windows use:
C:\john\john-386.exe --stdout --incremental >wordlist.txt

For Linux use:
#/usr/local/john/john --stdout --incremental >wordlist.txt

The resulting output will be written to the wordlist.txt file.

If you know the maximum length of the password you can use --stdout=length and john output passwords of that length or less. For example --stdout=5 will generate words that are 5 characters long or shorter. Please note that the maximum length john supports by default is 8. If you need to generate a 9 character or longer wordlist you will have to download the source and change a line or two of code. Or you can use a different tool.

You can also have john direct its output to other programs. For example to have john feed random passwords to air-crack use:
john --incremental=All --stdout | aircrack-ng -b 00:11:22:33:44:55 -w --test.cap

10.2 Configuring John the Ripper to use a wordlist

If you have a wordlist (wordlist.txt) you want to try against NTLM hashes use the following command:

john -f:NT -w:wordlist.txt hash.txt

or you can edit the john.conf file to use your wordlist. So it would look like this:

[Options]
# Wordlist file name, to be used in batch mode
Wordlist = $JOHN/wordlist.txt
# Use idle cycles only
Idle = N
# Crash recovery file saving delay in seconds
Save = 600
# Beep when a password is found (who needs this anyway?)
Beep = N

10.3 Using crunch to generate a wordlist

in /pentest/password/crunch run it with a --help to see the options. It will ask for a minimum length, maximum length, character set etc.

cd /pentest/password
crunch 1 8 abcdefghijklmnopqrstuvwxyz0123456789
or
crunch 1 8 abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
the character set can be whatever you want.

The optional fourth parameter allows you specific a pattern. For example if you know that the last 2 characters of a 8 character password are 99 you can do:
crunch 8 8 abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 -t @@@@@@99
The @ characters will change while the 99 remains constant.

The optional fifth parameter allows you to specify a starting string. This is useful if you have to stop in the middle a generation. Just do:
tail output.txt
to see where the file left off and use -s start one character higher. For example:
crunch 8 8 abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
aaaaaeDo
aaaaaeDp
aaaaaeDq
aaaaaeDr
aaaaaeDs
aaaaaeDt
aaaaaeDu

aaaaaeDv
aaaaaeDw
aaaaaeDx
aaaaaeDy
Press Ctrl-C
crunch 8 8 abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789 -s
aaaaaeDz
aaaaaeDz
aaaaaeDA
aaaaaeDB
aaaaaeDC
aaaaaeDD

Use >output.txt to redirect the output from the screen to a file named output.txt

10.4 Generate a wordlist from a textfile or website

Sometimes users use words from their employeer's website. Their password could be the companies name or the name of the flagship product. I will show you how to download the website and convert it into a wordlist. These instructions are basically a text version of Pureh@te's flash video from http://blip.tv/rss/flash/533887. I had to update my flash player so the video wasn't choppy. To update the flashplayer in BT4 do:
apt-get remove flashplugin-nonfree
wget http://fpdownload.macromedia.com/get/flashplayer/current/install_flash_player_10_linux.tar.gz && tar xzf install_flash_player_10_linux.tar.gz && mkdir ~/.mozilla/plugins && mv libflashplayer.so ~/.mozilla/plugins

1. boot BackTrack and login as root
2. # mkdir /target
3. # cd /target
4. # wget -r http://targetwebsite.com
   wget -r will recursively download a website
5. # cd /pentest/password/wyl
6. # perl wyd.pl -n -o - possiblepasswords.txt -/target
7. # cd /root
8. # cat possiblepasswords.txt

You now have a large unsorted file containing duplicate words. You will have to sort and uniq the file before it is of any value. See section 15.7 on how to use sort and uniq.

10.5 Using premade wordlists

Xploitz and pureh@te have released their wordlists. The first two are torrents; the rest can be downloaded from the URL.

pureh@te's wordlist - http://www.h33t.com/details.php?id=178f55c67ca0f522831dbc67042a34983e6652f5

Xploitz's first wordlist - http://thepiratebay.org/tor/4017231

Xploitz's second wordlist part 1 - http://www.mediafire.com/?am2exxlnwma
Xploitz's second wordlist part 2 - http://www.mediafire.com/?4mzmdsbhhcn
Xploitz's second wordlist part 3 - http://www.mediafire.com/?4v9znjscgwt

Xploitz's second wordlist part 4 - http://www.mediafire.com/?dmj6yy3gw3x
Xploitz's second wordlist part 5 - http://www.mediafire.com/?5tyidngmztx
Xploitz's second wordlist part 6 - http://www.mediafire.com/?3utzg3jk1mb
Xploitz's second wordlist part 7 - http://www.mediafire.com/?dmwdvdrgsgb

Pureh@te has released another wordlist. It is 64 million words 8-63 characters and it was made from his other wordlist.
http://www.megaupload.com/?d=7RN6ZB2E

10.6 Other wordlist generators

You could try the wg perl script from http://freshmeat.net/projects/wg/
perl ./wg.pl -l 8 -u 64 -v
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789\`\~\!\@\#\$\%\^\&*
\(\)\-_\+\=\[\]\;\'\,\.\/\< \>\?\:\"\{\}\|\ > words.txt

This will generate a list of "words" (actually character strings) between 8 and 64 characters long (-l 8 -u 64) and output it to a text file named "words.txt". The \ characters are there to escape the bash command characters.

Siph0n has taken a C program a thread, converted it to python, and made a couple of enhancements. Here is the python source code:

```
f=open('wordlist', 'w')

def xselections(items, n):
if n==0: yield []
else:
for i in xrange(len(items)):
for ss in xselections(items, n-1):
yield [items[i]]+ss

# Numbers = 48 - 57
# Capital = 65 - 90
# Lower = 97 - 122
numb = range(48,58)
cap = range(65,91)
low = range(97,123)
choice = 0
while int(choice) not in range(1,8):
choice = raw_input('''
1) Numbers
2) Capital Letters
3) Lowercase Letters
4) Numbers + Capital Letters
5) Numbers + Lowercase Letters
6) Numbers + Capital Letters + Lowercase Letters
7) Capital Letters + Lowercase Letters
: ''')

choice = int(choice)
```

```
poss = []
if choice == 1:
poss += numb
elif choice == 2:
poss += cap
elif choice == 3:
poss += low
elif choice == 4:
poss += numb
poss += cap
elif choice == 5:
poss += numb
poss += low
elif choice == 6:
poss += numb
poss += cap
poss += low
elif choice == 7:
poss += cap
poss += low

bigList = []
for i in poss:
bigList.append(str(chr(i)))

MIN = raw_input("What is the min size of the word? ")
MIN = int(MIN)
MAX = raw_input("What is the max size of the word? ")
MAX = int(MAX)
for i in range(MIN,MAX+1):
for s in xselections(bigList,i): f.write(''.join(s) + '\n')
```

NOTE: When generating your own wordlists keep in mind that some programs (aircrack-ng) have a 2GB file size limit.

10.7 Manipulating your wordlist

So you now have a wordlist. What should you do with it? You need to manipulate it into something you can use. You can of course just use the wordlist without manipulating it, but you may be sacrificing performance and space. The first thing you should do is to combine, sort, and uniq all of the wordlists you have.

The following is from http://freeworld.thc.org/thc-hydra/README

uniq your dictionary files! this can save you a lot of time :-)
cat words.txt | sort | uniq > dictionary.txt

If you know that the target is using a password policy (allowing users only to choose password with a minimum length of 6, containing a least one letter and one number, etc. use the tool pw-inspector which comes along with the hydra package to reduce the password list:
cat dictionary.txt | pw-inspector -m 6 -c 2 -n > passlist.txt

If you are creating a wordlist specifically to crack WPA and WPA2 then the following command is what you should use as it will get rid of any words shorter than 8 characters and longer than 63:
cat dictionary.txt | pw-inspector -m 8 -M 63 > WPAwordlist.txt

The above paragraph is true for any wordlist and very good advice. Remember you want as many possible words as you can fit on your storage device. More is better as long as the list only contains unique words.

To see the number of lines (or words in this case) in the file:
wc -l wordlist.txt

Now that you have a sorted, uniq'd wordlist you need to look at it and see what is in it. Opening a large wordlist in vi or kwrite can take a lot of time and memory. I prefer to use head and tail to view the beginning and end of the wordlist.
To view the first 30 lines of wordlist.txt:
head -n 30 wordlist.txt
To view the last 30 lines of wordlist.txt:
tail -n 30 wordlist.txt

Say that the first 25 and last 23 lines are garbage and you want to delete them. Opening wordlist.txt in vi and kwrite might not work so good. You can use sed to change the file
To delete the first 25 lines of wordlist.txt:
sed '1,25d' wordlist.txt
To delete the last 23 lines of wordlist.txt:
sed -n -e :a -e '1,23!{P;N;D;};N;ba'

Next you can use a permutator on your wordlist to make it bigger. A permutator will add wordlists based on criteria you set. For example you want to a number to each word in your wordlist. Or you want to change all of the a's to @'s. You can get a permutator from:
http://www.backtrack-linux.org/forums/backtrack-howtos/689-wordlist-menu-tool-backtrack-4-final.html

You can also take your wordlist and run it through john the ripper and increase its size by about 49 times. Use:
john --wordlist=Wordlist.txt --rules --stdout >largelist.txt

If you are feeling a little lazy M1ck3y has created a script named Giga Wordlist Creator that can do all of this for you. You can download the script and find how to use it at (English Translation) http://translate.google.fr/translate?u=http%3A%2F%2Fwww.crack-wpa.fr%2Fforum%2Fviewtopic.php%3Fid%3D126&hl=fr&ie=UTF-8&sl=fr&tl=en

NOTE: When generating your own wordlists keep in mind that some programs (aircrack-ng) have a 2GB file size limit. There is also the issue of RAM. If you can keep the entire wordlist in RAM, the cracking will proceed that much faster. So keep your wordlists to a maximum of 2GB. If you have a 3GB wordlist and you want to break it into 1GB chucks do:
split -bytes=1024 m /tmp/dictionary_file_3GB /tmp/smaller_dictionary_file
please substitute 1024 m for whatever size you wish and change /tmp/ to the proper path you want to use.

It is also popular to leetify passwords. For example password becomes p@ssword, pa$$word, passw0rd, p@$$w0rd, etc. Here is a perl script written by Gitsnik and a modified version of Gitsnik's script and modified by robot

```
#!/usr/bin/env perl
```

```
use strict;
use warnings;

my %permution = (
"a" => [ "a", "A", "4", "@", "&"],
"b" => "bB8",
"c" => "cC",
"d" => [ "d", "D", "|)" ],
"e" => "eE3",
"f" => "fF",
"g" => "gG9",
"h" => "hH",
"i" => "iI!|1",
"j" => "jJ",
"k" => [ "k", "K", "|<" ],
"l" => [ "l", "L", "!", "7", "1", "|", "|_" ],
"m" => [ "m", "M", "/\\/\\" ],
"n" => [ "n", "N", "|\\|" ],
"o" => [ "o", "O", "0", "()" ],
"p" => "pP",
"q" => "qQ",
"r" => [ "r", "R", "|2" ],
"s" => "sS5\$",
"t" => "tT71+",
"u" => "uU",
"v" => [ "v", "V", "\\/" ],
"w" => ["w", "W", "\\/\\/" ],
"x" => "xX",
"y" => "yY",
"z" => "zZ2",
);

# End config

while (my $word = <>) {
chomp $word;
my @string = split //, lc($word);
permute(0, @string);
}

sub permute {
my $num = shift;
my @str = @_;
my $len = @str;

if ($num >= $len) {
foreach my $char (@str) {
print $char;
}
print "\n";
return;
```

```
}

my $per = $permution{$str[$num]};

if ($per) {
my @letters = ();
if (ref($per) eq 'ARRAY') {
@letters = @$per;
} else {
@letters = split //, $per;
}
$per = "";

foreach $per (@letters) {
my $s = "";
for (my $i = 0; $i < $len; ++$i) {
if ($i eq 0) {
if ($i eq $num) {
$s = $per;
} else {
$s = $str[0];
}
} else {
if ($i eq $num) {
$s .= $per;
} else {
$s .= $str[$i];
}
}
}
my @st = split //, $s;
permute(($num + 1), @st);
}
} else {
permute(($num + 1), @str);
}
}
```

A python script that leetifies passwords is included with BackTrack 4. It can be found in /pentest/passwords /cupp. It is easy to use.

1. boot BackTrack and login as root
2. open a terminal window
3. # cd /pentest/password/cupp
4. # ./cupp -i
5. answer the questions and wait until the program is fisished generating its file.

11 Rainbow Tables

11.1 What are they?

Rainbow tables are files that contain pre-computed hashes of passwords. This drastically cuts down on the time it takes to crack passwords. You will need a set of tables for each algorithm you want to break. Since LM hashes are so weak I do not waste the disk space to store them. The last time I checked the size of the LM hash rainbow tables for a password of 1 to 7 characters and a character set of:

[ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/] was 120GB in size. The table was generated by Hak5. This table is supposed to have a 99.9% success rate. You do not need a table with 8 to 14 characters as the password is split into 2 separate 7 character hashes.

NTLM tables are larger as the possible character set is:

[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/] and the password is not broken into smaller parts. Given this and most likely you have 6 or 7 character minimum password length; the size of the tables can be quite large. I have not found anywhere to download a reasonable set of NTLM rainbow tables. To me reasonable is from 1 to 14 characters using the above character set. A table using these specs will extremely large.

11.2 Generating your own

To generate you rainbow tables you need three things: what algorithm are you going to need, the software to generate the table, a really fast PC or FPGA or GPU. Let us start with the NTLM algorithm. The BackTrack software has rtgen available to generate the tables. If you want to generate tables using windows download Cain & Abel from http://www.oxid.it/projects.html as that contains the latest release of winrtgen. For more information about rainbow tables visit The Ethical Hacker Network at http://www.ethicalhacker.net/content/view/94/24/

A set of tables that use the loweralpha-numeric-symbol14 character set for a maximum of 8 character password and a 99.6972% probability of success will take 120 tables and 175GB of disk space. I used winrtgen to get these figures.

11.2.1 rcrack - obsolete

The current version of rcrack is 1.4 however version 1.3 and 1.4 only support windows. Version 1.2 is the only version that supports Linux. If you need to generate rainbow tables for any algorithms other than LM, MD5, and SHA1 you will have to recompile the RainbowCrack source code. The rtgen utility that comes with BackTrack 3 can only generate LM, MD5, and SHA1 tables. To recompile:

1. boot BackTrack and login as root
2. download the RainbowCrack source code from http://project-rainbowcrack.com/index.htm#download
3. Next download the algorithm patch using the link from the page above. This patch adds support for the NTLM, MD2, MD4 and RIPEMD160 algorithms
4. open a terminal prompt
5. # mkdir rc
6. # mv rainbowcrack-1.2-src.zip rc
7. # mv rainbowcrack-1.2-src-algorithmpatch.zip rc
8. # cd rc
9. # unzip rainbowcrack-1.2-src.zip
10. # unzip rainbowcrack-1.2-src-algorithmpatch.zip
11. # cp rainbowcrack-1.2-src-algorithmpatch/Hash* rainbowcrack-1.2-src/src/
12. # cd rainbowcrack-1.2-src
13. # make -f makefile.linux

14. # cp rcrack /pentest/password/rcrack/
15. # cp rt* /pentest/password/rcrack/
16. # cd /pentest/password/rcrack
17. # rtgen

Look for the hash algorithm section and examine the list available algorithms. You should now see NTLM, MD2, MD4 and RIPEMD160 have been added. If they aren't there start over and try again.

Replace the charset.txt file in /pentest/password/rcrack with the following:
charset configuration file for winrtgen v1.2 by Massimiliano Montoro (mao@oxid.it)
compatible with rainbowcrack 1.1 and later by Zhu Shuanglei (shuanglei@hotmail.com)

alpha = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
alpha-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ]
alpha-numeric = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
alpha-numeric-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
alpha-numeric-symbol14 = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=]
alpha-numeric-symbol14-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=]
alpha-numeric-all = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]
{}|\:;"'<>,.?/]
alpha-numeric-all-space = [ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]
{}|\:;"'<>,.?/]

loweralpha = [abcdefghijklmnopqrstuvwxyz]
loweralpha-space = [abcdefghijklmnopqrstuvwxyz]
loweralpha-numeric = [abcdefghijklmnopqrstuvwxyz0123456789]
loweralpha-numeric-space = [abcdefghijklmnopqrstuvwxyz0123456789]
loweralpha-numeric-symbol14 = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=]
loweralpha-numeric-symbol14-space = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+="]
loweralpha-numeric-all = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=~`[]{}|\:;"'<>,.?/]
loweralpha-numeric-all-space = [abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-_+=~`[]
{}|\:;"'<>,.?/]

mixalpha = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ]
mixalpha-space = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ]
mixalpha-numeric = [abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
mixalpha-numeric-space =
[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789]
mixalpha-numeric-symbol14 =
[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=]
mixalpha-numeric-symbol14-space =
[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=]
mixalpha-numeric-all =
[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]
{}|\:;"'<>,.?/]
mixalpha-numeric-all-space =
[abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()-_+=~`[]
{}|\:;"'<>,.?/]

byte = []
numeric = [0123456789]

numeric-space = [0123456789]

This new file adds support for many more character sets than the one that comes with rcrack or rcracki.

11.2.2 rcracki

rcracki is a modified version of rcrack which supports hybrid and indexed tables. The advantage of indexed tables is the size is reduced by 50% and lookup times are faster. Rainbowtables generated by rcrack and NOT compatible with rcracki. Rainbowtables generated by rcracki are NOT compatible with rcrack. You can convert rcracki rainbowtables to the old format using the rti2rto utility. Currently there is no rto2rti utility although one might be in the works. A hybrid table is a table that fits into 1 of 2 categories: 1) common passwords like password are tried first 2) the password is a combination of two character sets. For example password123 is a hybrid of the loweralpha and numeric character sets.

I spent almost 4 hours trying to get rcracki to compile. I seceded in compiling rotten by adding md5.cpp to the makefile.linux rtgen section. I added md5.cpp right before RainbowTableGenerate.cpp and then rtgen compiled fine. That took about 5 minutes to figure out and fix. I spent the rest of the time of rtdump. I just gave up. The linux version of rcracki is new so some things just don't work yet. Work is being done to get rcracki to work properly on linux but there is no ETA on when that will happen.

Here is how you are supposed to compile rcracki but it doesn't work yet:

1. boot BackTrack and login as root
2. download the rcracki source from http://www.freerainbowtables.com/phpBB3/viewtopic.php?&t=564
3. open a terminal window
4. # mkdir rcracki
5. # mv rcrackisrc.zip rcracki
6. # cd rcracki
7. # unzip rcrackisrc.zip
8. # make -f makefile.linux
9. # mkdir /pentest/password/rcracki
10. # cp rcrack /pentest/password/rcracki
11. # cp rtdump /pentest/password/rcracki
12. # cp rtgen /pentest/password/rcracki
13. # cp rtsort /pentest/password/rcracki
14. create the charset.txt using the charset.txt from rcrack section

11.2.3 rcracki - boinc client

The current version of rcrack is now a boinc client. BOINC is an open-source software platform for computing using volunteered resources. By using the boinc client PowerBlade (maintainer of rcracki) only has to compile a small binary for each platform. This frees his time to work on taking generating rainbow tables to the next level. There are versions of the client for Windows (32-bit and 64-bit), Linux (32-bit and 64-bit), Mac OS X.

All you have to do is download and install the correct boinc client for your system from http://boinc.berkeley.edu/download_all.php. Next point the boinc client to the project url of http://boinc.freerainbowtables.com/distrrtgen/

You need to create a separate account for chain generation. If you use the same email for your account at boinc.freerainbowtables.com and www.freerainbowtables.com, you will be able to view statistics about your

progress.

11.2.4 Generating a rainbow table

rtgen ntlm loweralpha-numeric-symbol14 1 8 0 20000 97505489 _XXX
rtgen is application
ntlm is the algorithm you want to rtgen to use
loweralpha-numeric-symbol14 is the character set you want to use
1 is the minimum password length
8 is the maximum password length
0 is the table index - do not worry about it
20000 is the chain length, higher numbers increase the probability success rate without increasing the table size much. However a larger number will cause rcrack or rcracki a larger amount of time to walk them thus increasing the time it takes to crack the hashes. 20000 is probably the largest you should use.
97505489 is the number of chains to store in the table. Think of this as the size of table you want to generate. 97505489 is the right size to generate a table if you are going to store them on a DVD. With 97505489 you can store 3 tables on 1 DVD. Use 40000000 if you are going to store the table on a CD. You can use whatever number you want as long as the resulting table is less than 2GB in size. The password cracking tools have issues with tables greater than 2GB.
_XXX is whatever you want appended to the filename. Since I know I need 120 tables I will start with _001 and end at _120

So to generate 3 tables you do:
rtgen ntlm loweralpha-numeric-symbol14 1 8 0 20000 97505489 _001
rtgen ntlm loweralpha-numeric-symbol14 1 8 0 20000 97505489 _002
rtgen ntlm loweralpha-numeric-symbol14 1 8 0 20000 97505489 _003

which will result in the following files be created:
ntlm_loweralpha-numeric-symbol14#1-8_0_20000x97505489_001.rt
ntlm_loweralpha-numeric-symbol14#1-8_0_20000x97505489_002.rt
ntlm_loweralpha-numeric-symbol14#1-8_0_20000x97505489_003.rt

Once all of the tables have been generated you have to sort them before you can use them:
rtsort ntlm_loweralpha-numeric-symbol14#1-8_0_20000x97505489_001.rt
rtsort ntlm_loweralpha-numeric-symbol14#1-8_0_20000x97505489_002.rt
rtsort ntlm_loweralpha-numeric-symbol14#1-8_0_20000x97505489_003.rt

Each command will take awhile to complete. The larger the file the longer it takes. It is recommended that you do this on a PC with at least 2GB of RAM (if you use 97505489 as the number of chains) so the entire file can be loaded into RAM which makes the sort faster.

11.3 WEP cracking

You do not need rainbow tables for WEP. See other postings for WEP cracking details.

11.4 WPA-PSK

Rainbow tables can be used to crack the Pre-Shared Keys of WPA. The Church of WIFI has a 40GB torrent of rainbow tables that use ~1,000,000 words for a total of approximately 40GB of hash tables for the top 1000 SSID's. The torrent is at http://rainbowtables.shmoo.com/

You will need a torrent client as BackTrack 4 does not come with one. I suggest bittorrent. Ktorrent is also available however to install it on BackTrack Ktorrent will add in 46 dependencies and bittorrent will only add 1 dependience. To install all you need to do is:
apt-get install bittorrent
btdownloadcurses --url http://rainbowtables.shmoo.com
/95896A255A82D1FE8B6A2BFFC098B735058B30D7.torrent

Make sure you have at least 175GB of free disk space before you try this (37GB for the lzma file, 42GB for the tar file, 42GB for the individual files, and approximately 100MB for each import into the final database). Of course you can delete the lzma file once your have the tar file, and then delete the tar file once you have untarred the files and directories.

11.4.1 airolib

to install the rainbow table in airolib for BT3 or BT4:

1. # lzma -d wpa_tables.tar.lzma
2. Goto to lunch. The decompression will take awhile depending on fast your processor is.
3. # tar -x wpa.tar
4. Go home or goto sleep and come back the next morning with a name for the database you are about to create. I am using dbname for this example.
5. # airolib-ng dbname --import cowpatty /path/to/file/you/want/to/import for example
   # airolib-ng dbname --import cowpatty /mnt/usb/wordlist/wpa_psk-h1kari_renderman/xaa-0
   /2WIRE044
   You will see
   Reading header...
   Reading...
   Updating references...
   Writing...
6. # airolib-ng dbname --stats
   There are 1 ESSIDs and 995759 passwords in the database. 995759 out of 995759 possible combinations have been computed (100%).

   ESSID Priority Done
   2WIRE044 64 100.0

Now the odds of your SSID being in this database are probably not good. You can check the SSID.txt file that came with the torrent for your SSID. If your SSID is not in the database you can add it. To add a SSID you need to create a text file that has all of the SSIDs you want to add. airolib won't accept a SSID from the command line. Then you import the file:
airolib-ng dbname --import essid newssid.txt
Reading file...
Writing...
Done.
cat newssid.txt
Test1
Test2
Test3
#airolib-ng dbname --stats
There are 5 ESSIDs and 995759 passwords in the database. 1991518 out of 4978795 possible combinations have been computed (40%).

ESSID Priority Done
2WIRE044 64 100.0
2WIRE047 64 100.0
Test1 64 0.0
Test2 64 0.0
Test3 64 0.0
Finally start batch processing to compute the hashes for the new SSIDs
airolib-ng test --batch
This will take a long time depending on your processor speed. The nice thing is once it is done the table can be used over and over thus speeding up future attacks.

To use the rainbow table do:
aircrack-ng -r dbname wpa.cap

11.4.2 pyrit

Pyrit is a GPU aware rainbow table generator geared specifically for WPA-PSK and WPA2-PSK password testing. Pureh@te has the definitive guide available at http://www.backtrack-linux.org/documents/BACKTRACK_CUDA_v2.0.pdf.

12 Distributed Password cracking

You can have a very fast computer but is not a match for 10 computer working on the problem. That is distributed processing. Each node in the cluster gets to work on a small part of the problem. There is a central node or server to coordinate the work.

12.1 john

The openwall wiki (http://openwall.info/wiki/john/parallelization has a great article on the various attempts to add distributed processing capabilities to john the ripper.

12.2 medussa (not a typo this is not medusa)

There is some well written documentation on how to compile and use medussa at http://www.krazyworks.com/distributed-password-cracking-with-medussa/. It appears that medussa hasn't seen any updates since 2004 but still appears to work fine. BackTrack 4 doesn't ship with gmp so you must do:
apt-get install libgmp3-dev
which will download the missing dependency and its dependencies. gmp is required to do high precision math. It has nothing to do with gimp. medussa supports unixcrypt. It also supports md5 (FreeBSD, Linux, cisco) and sharaw. MD5 is slow and sharaw works on SHA1 with no salting.

13 using a GPU

Cuda is an acronym for Compute Unified Device Architecture. NVIDIA developed cuda as a parallel computing architecture which allows users to use their graphics card (if supported by cuda) to process data instead of their CPUs. ATI also has hardware and software that you can use to accelerate your nongraphical applications. ATI's solution is called Stream. You are probably thinking why would I want my graphics card to process data. The answer is you probably don't except for specific circumstances, i.e. password cracking. CPUs are designed for general purpose number crunching. As encryption has become more common place

Intel and AMD have added instructions to their processors to make certain algorithms perform faster. Intel has added 6 instructions to some of their processors in an effort to increase the speed of the processing AES encryption and decryption. There are also 2 instructions for AES key expansion. These instructions, once used by software, will increase the speed of cracking of AES. However these instructions pale in comparison to a graphics card. A graphics card does nothing but compute large numbers and that is basically all encryption is, the computing of large prime numbers.

13.1 cuda - nvidia

The definitive guide was written by pureh@te and can be downloaded from http://www.backtrack-linux.org/documents/BACKTRACK_CUDA_v2.0.pdf. As the title says cuda is supported on NVIDIA cards. It is possible to run cuda applications on ATI cards however the results have been disappointing. If you have an ATI see the next section.

BackTrack 4 has several cuda enabled applications. Pyrit is probably the most used. See 11.4.2 pyrit on how to use pyrit.

13.2 Stream - ati

You can read about Stream at http://www.amd.com/us/products/technologies/stream-technology/Pages/stream-technology.aspx. BackTrack 4 doesn't have any applications that support Stream that I know of. I don't have the hardware to test any applications for Stream support.

14 example hash.txt

Administrator:500:CEEB0FA9F240C200417EAF50CFAC29C3:D280553F0103F2E643406517296E7582:::
a:1011:7584248B8D2C9F9EAAD3B435B51404EE:186CB09181E2C2ECAAC768C47C729904:::
b:1012:AC5BA6A944526699AAD3B435B51404EE:F07A9DFFFC2C5C7F9D9EBC83FD69D68E:::
c:1013:E7EED3F5C2C85B88AAD3B435B51404EE:6AA15B3D14492D3FA4AA7C5E9CDC0E6A:::
d:1014:8142C9E2CCAFA80EAAD3B435B51404EE:D43EB3C12D597DFE44A31859C586B2DC:::

Changes from version 0.1

- Added a section on plain-text.info
- Added a section on using john the ripper with a custom character list
- Added Xploitz's and pureh@te's wordlists

Changes from version 0.2

- Added sections on using Ophcrack
- Added sections on Cain and Abel under windows
- Fixed a typo
- New html format

Changes from version 0.3

- Moved some content to where it should have been
- Added a section on crunch
- Fixed typos
- Fixed a whole lot of html errors
- Slightly rearranged things to flow better

Changes from version 0.4

- Added a section on cached credentials
- Expanded the Novell section
- Fixed typos
- Fixed a couple of html errors
- Moved some things around
- New utility to dump passwords

Changes from version 0.5

- added wpa pw-inspector command
- added a wordlist manipulation section
- added usage of fgdump
- added rcracki section
- added a sample hash.txt to play with
- moved a few things around for a better flow

Changes from version 0.6

- added a section on generating a wordlist from a website
- added head, tail, and sed commands to wordlist manipulation
- added a section on Xhydra (pointing to Pureh@te's video)
- added a section on gsecdump
- added a section on medusa
- added a section on Cisco
- expanded the crunch section
- moved everything around in an effort to make things easier to find
- the dumbforce and knownforce are not finished

Changes from version 0.7

- Fixed john --incremental=All --stdout | aircrack-ng -b 00:11:22:33:44:55 -w --test.cap missing a - Thanks to roblad for pointing it out.

Changes from version 0.8

- updated the guide to support BT4-Pre-Final
- switched from transitional html to strict html
- added a section on cracking office passwords
- added a section on cracking rar passwords
- added a section on cracking zip passwords
- added a section on cracking pdf passwords
- added instructions for remote password dumping for fgdump
- point users to john the ripper wiki for dumbforce and knownforce usage

- convert text urls to links
- updated links
- update versions of software (wine and rcrack)
- fixed several spelling mistakes

Changes from version 0.9

- updated the guide to support BT4 Final
- fixed links to pureh@te's videos
- added instructions to update flash so you can watch videos clearly
- updated the john the ripper section to current versions
- added instructions for using rainbow tables for WPA cracking
- added two new leetifing scripts
- added a small cupp discussion
- added a section on pyrit
- added a section on distributed password cracking using john and medussa
- added a section on cuda and stream
- fixed typo in hash.txt

Changes from version 1.0

- removed references to old remote-exploit.org website