Note: To be viewed with a Monospaced, 9-point Font (i.e. Monaco, Courier)

```
 #### ####   #####   #####   #####  ######   #####   #####   DOCUMENTATION
### ### ### ### ### ### ### ### ### ### ### ### ### ### ###     by oleBuzzard
### ### ### ### ### ### ### ### ### ### ### ### ### ### ###
### ### ### ### ### ###     ###     ### ### ### ### ###        %%%     %   %%
### ### ### ####### ###     ###     ######  ####### ###       %  %%   %%  %  %
### ### ### ### ### ### ### ### ### ### ### ### ### ### ###   % % %     %  %%%%
### ### ### ### ### ### ### ### ### ### ### ### ### ### ###   %%  %     %  %  %
### ### ### ### ###  #####   #####  ### ### ### ###  #####     %%%  % %%% %  %
-[01-29-96]-----------------------------------------------------------------------
```

INTRODUCTION

FINALLY! A half-way decent UNIX 'passwd' cracker for the Macintosh. MACCRAC is a very well ported version of one of the PC world's best 'passwd' Crackers, CRACK V4.1. MACCRAC is great if you know how to use it, AND, more importantly, if you know what UNIX password cracking is about in the first place. Unfortunatley, the Mac underground have been SO long deprived of a decent UNIX passwd cracker, alot of us are quite a bit behind in the concept. That's what this tutorial is provided for. Hopefully after reading it, not only will you have an understanding of how to use MACCRAC, but also an increased understanding of what UNIX hacking is about in the first place.

PURPOSE OF CRACKING THE passwd

Traditionally stated, the purpose of hacking a UNIX is: to "get to ROOT." This refers to the ROOT account that every UNIX system has as part of it's Operating system. The ROOT is a 'Trusted User' account, THE most powerful account on a UNIX. If you can hack a ROOT you can utilize or exploit every function a UNIX is capable of. But to get to "ROOT" you have to have somewhere to start. For the purposes of this file, that somewhere is with the 'passwd' file.

WHAT'S THE passwd?

'passwd' is the common name of the file in which user account information is

stored on a UNIX system. You might consider it a comprehensive users list. The file contains the information for an accounts USERNAME, PASSWORD, USER NUMBER, GROUP, GECOS, HOME DIRECTORY, and SHELL. A single entry of a passwd file entry might look like this:

```
              PASSWORD        GROUP NUMBER    HOME DIRECTORY
             /               /               /
            /               /               /
kbahadur:8d34jSjs73hsb:2162:15:Ken Bahadur:/usr/users/kbahadur:/usr/bin/ksh
 \                      \            \                           \
  \                      \            \                           \
```

USERNAME USER NUMBER GECOS INFORMATION SHELL

Now take a look at the PASSWORD in this entry: 8d34jSjs73hsb. This is, in fact, NOT the password. It is, instead, the encrypted equivalent TO the password. As part of the UNIX Account Registration process, when a User designates a password, the UNIX takes the password, and (*this is important*) uses the other information from the account to generate an encrypted equivalent to the actual password. Why? Because as part of the UNIX operating system, users MUST have access to the 'passwd' file to be able to login. But if anyone who has an account can access the 'passwd' file, they can also see what everyone else's Password is. So, UNIX's security against this is to encrypt the password entry for each users account so that noone else will know what anyone elses password is. Unfortunaley/fortunatley (depending on who you are) the algorithm UNIX uses to perform this encryption has been known to Hackers for sometime. And so if

you can see this:

encrypted equivalent of pasword
/
kbahadur:8d34jSjs73hsb:2162:15:Ken Bahadur:/usr/users/kbahadur:/usr/bin/ksh

...you can use MACCRAC or any other of well over 50 'passwd' file crackers to "guess" the password to this account entry. "Guess?" You say? "How does that work?" It works like this:

GUESSING THE PASSWORD

First a UNIX 'passwd' file cracker takes an encrypted password equivalent (i.e.: 8d34jSjs73hsb) from an account entry in a UNIX 'passwd' file and holds it to be used as a Reference. From whichever account entry the encrypted equivalent was pulled, is the particular account the 'passwd' file

cracker will attempt to crack at that time.

Next the 'passwd' file cracker goes through a process of "guessing". In this

process a single word is pulled from a Dictionary file (more on Dictionaries later), encrypted utilizing the UNIX encryption algorithm (the one all us hackers know about), and compared, checking to see if the derived encrypted word matches the encrypted password equivalent used as a Reference.

If the encrypted word matches the Reference, the 'passwd' file cracker considers it an accurate guess, it then logs the information, and moves on to the next account. If the two do not match, the 'passwd' file cracker pulls another word from the Dictionary file and goes through the guessing process again. If the 'passwd' file cracker goes through every word in a Dictionary file and never matches the Reference, the entry is skipped, and the cracker moves on to the next account.

Now, as complicated as this may seem, it is all a relativley easy task for a

computer. As such, UNIX 'passwd' files are cracked on a regular basis. As a result of this a number of security and other measures now (potentially) exist to prevent unauthorized persons from accessing a UNIXes'passwd' file. This is the topic of the next section. To this point you should understand why UNIXes are hacked (to get to ROOT) and understand a little about 'passwd' files and their role in UNIX hacking. Got it?

GOT IT, NOW WHAT?

Ok, at this point you should be ready to try and find a UNIX 'passwd' file to crack, right? Wrong. You still have a couple of minor, requisite tasks to perform. First, (obviously) you'll need to find a UNIX to hack. In most cases, you've already got one in mind, but just in case you don't we'll take a look at a few. Also, once you've found a UNIX to hack, you'll need an account on that UNIX. There's no way to steal the 'passwd' file from a UNIX without first having an account on it (not true, you can always get a 'passwd' file from someone else, but ignore this because I'm contradicting myself). Once you've accomplished your requisites you can start trying to steal the 'passwd' file.

Step 1. Finding a UNIX to Hack

Seeing as how you're reading this file you probably already have a UNIX in mind. But, for the sake of clarity, heres what a common UNIX login screen looks like:

```
Ultrx v4.3 (rev .44)


login:
```

Other UNIX machines are: System V, BSD, Xenix, and AIX. Look for these names to be somewhere in the login screen. Knowing what type of UNIX you're using will aid you in hacking it.

Step 2. An account to start with

If you already have a UNIX account go to Step 3. If you do not already have an account, you need to get one. Either: trade for one, trash for one, get a legitimate one, or hack one out by hand. The first three options are probably the easiest. You can trade for UNIX accounts on IRC channels #hack or #phreak. You can trash for accounts in dumpsters and trashcans at most Colleges or Universities. You can buy legitimate accounts from any one of the rapidly increasing number of Internet Service Providers (they almost all use UNIX). But, of coure, as well know you're a hacker, and the only hing you wanna do is Hack an account. So be it. Here's a list of UNIX defaults.

NOTE These are NON-PASSWORDED accounts. They are common on System V, BSD, Xenix, and AiX. "These defaults are included in standard setup on various machines so the Sysadmin can log on for the first time." In some instances, negligent Admins will forget to change or delete these accounts. If so, you've got an account to start with. Remember, these are NON-PASSWORDED so if they work you shouldn't be prompted for a password. If a password is prompted for, try using the Account name for the password as well.

[Stolen from CoTNo #01]

root	bin	adm
makefsys	sysadm	sys
mountfsys	rje	sync
umountfsys	tty	nobody
checkfsys	somebody	setup
lp	powerdown	ingres
dptp	general	guest
daemon	gsa	user
trouble	games	help
nuucp	public	unix
uucp	test	admin
student	standard	pub
field	demo	batch
visitor	listen	network
uuhelp	usenet	sysinfo
cron	console	sysbin
who	root2	startup
shutdown	ncrm	new

Step 3. Stealing the passwd file

Once you've got your UNIX accpunt you can ATTEMPT to steal the 'passwd' file from it. I emphasize ATTEMPT because the 'passwd' file can be protected in a number of ways, or located in a number of different places. We will explore some common methods of exploiting the 'passwd' file.

-Common UNIX Hack-

This is probably THE easiest and most common UNIX hack. ogin in to your account and try typing this at the prompt:

```
 prompt   concatenate          Note on: 'booya>' is the name of the account
  /      /                     prompts   prompt on the machine I'm using in
booya> cat /etc/passwd                    these examples. The prompt on your
           /      \                        machine will be different. Also
      directory    filename                DON'T type 'booya>' with an entry.
```

'cat' is short for concatenate, a command used for reading and displaying files in standard output. '/etc' is the common directory for the password file on older UNIXes. 'passwd' is the common password filename on UNIXes. If

you entered: cat /etc/passwd and got a listing that looks like this (abbreviated):

```
kbahadur:IS3fhZdWX3JGU:2162:15:Ken Bahadur:/usr/users/kbahadur:/usr/bin/ksh
             \
            password intact
```

...then congrats! You've succesfully listed out (stolen) your first 'passwd' file. *Buffer* the entire contents to a text file, save it and jump down to the section: MACCRAC-ING.

If you got a listing that looks like this:

```
        password tokenized
         /
intruder:x:263:200:Jack Harmon:/usr/users/intruder:/bin/csh
```

or:

```
esvogt:PASSWORD HERE:2183:129:Novel,,,:/usr/users/advisor/esvogt:/usr/bin/ksh
           \
       password removed
```

or you got:

```
cat: cannot open /etc/passwd
```

Then the UNIX you are on is utilizing some other form of protection or may be using a different 'passwd'-ing process. Keep reading.

-AIX-

On AIX systems, an UNIX variation, the 'passwd' file is in a different place. On an AIX type:

```
booya> cat /etc/security/passwd
```

If this lists out a 'passwd' file with the (encrypted) password intact, then you've succesfully listed out (stolen) your first 'passwd' file. *Buffer* the entire contents to a text file and save it, and jump down to MACCRAC-ING. If not, keep reading.

-NIS/yp-

Some UNIXes use a system called Yellow Pages [taken from #hack/alt.2600 FAQ beta .013]:

> "NIS (Network Information System) is the current name for what was once known as yp (Yellow Pages). The purpose for NIS is to allow many machines on a network to share configuration information, including password data. NIS IS NOT DESIGNED TO PROMOTE SYSTEM SECURITY. If your system uses NIS you will have a very short /etc/passwd file that includes a line that looks like this:

```
+::0:0:::
```

"To view the real password type this command:"

```
booya> ypcat passwd
```

If 'ypcat' lists a password file with the (encrypted) password still intact, *buffer* the entire contents and go on to MACCRAC-ING, if not, keep reading.

-Password Shadowing-

Some systems use what is called password shadowing [again, taken from #hack/alt.2600 FAQ beta .013]:

"Password shadowing is a security system where the encrypted password field of /etc/passwd is replaced with a special token and the encrypted password is stored in a separate file which is not readable by normal system users.

"To defeat password shadowing on many (but not all) systems, write a program that uses successive calls to getpwent() to obtain the password file.

"Example:

```
-----------------------------------------------------------------CUT HERE
#include <pwd.h>
main()

struct passwd *p;
while(p=getpwent())
printf("%s:%s:%d:%d:%s:%s:%s\n", p->pw_name, p->pw_passwd,
p->pw_uid, p->pw_gid, p->pw_gecos, p->pw_dir, p->pw_shell);

-----------------------------------------------------------------CUT HERE
```

Now then, for those you who are unfamiliar with UNIX scripts and/or their implementation, follow these directions:

First Copy the above script (not including the CUT HEREs) into a Text file and save it as 'getp.c'. Next Login to your UNIX account and create a directory called 'executables'. (At the prompt) Type:

```
 prompt       directory name
  /             /
booya> mkdir executables
        /
   make directory
```

Now, use Fetch or some other FTP client to FTP into your account and Upload 'getp.c' into the directory 'executables'. Once you've done this, login to your account, and goto the 'executables' directory:

```
       change directory
        /
booya> cd executables
```

Type 'ls' to List the directory to make sure the file is there. If it is you can attempt to compile the 'getp.c' script. Almost all UNIX boxes have Compilers, it's just a matter of whether or not you have acces TO the Compiler. Typically you do. at the UNIX prompt Type:

```
prompt   compiler      executable
  \     /             /
booya> cc -o getp.c getfile
          /     \
    output      filename
    option
```

If you don't get an error you should be left with a file named 'a.out'. Type:

```
booya> a.out
```

If you get a listing with the (encrypted) password intact, *buffer* the contents to a text file and go on to MACCRAC-ING. if not, keep readin'.

If you got an error when you tried to compile the 'getp.c' script: 'cc: Command not found' then you either don't have that compiler or you don't have access to it. In either case, try compiling with the GNU C Compiler:

```
        gnu c compiler
        /
booya> gcc getp.c
              \
            filename
```

Again, you should be left with a file named 'a.out'. At the UNIX prompt type: a.out. If you get a password file with the (encrypted) password file intact, *buffer* the entire contents and go on to MACCRAC-ING. If not, keep reading.

-Last Resorts-

In some cases none of the above listed attacks may work. It might be because

you're running a newer version of UNIX like SunOS v5.4. Also it, may just be

that you don't have permissions to access the 'passwd' file for whatever reason. In the case of SunOs v5.4, v5.4 doesn't have those helpful v4.1.x bugs so well documented in the CERT Advicories. In this case your best bet may be to go pick up a book on UNIX (so you can know what you're doing), and

then goto the Bugtraq Archives:

http://www.eecs.nwu.edu/~jmyers/bugtraq/search.html

...and do a search for 'SunOS 5.4'. Any vulnerabilities in 5.4 (or any other system for that matter) may be found there.

In cases where you just don't have access to the 'passwd' file for whatever reason, you might try the 'Dumb User' Hack: Login to a UNIX using whatever account you have. Once you're logged in, at the prompt type:

```
      change directory up 1
           /
booya> cd ..
         ^
         Note space ' ' between 'cd' and '..'

booya> ls
        \
     lists contents of directory     accounts
                                     /     \
1031exch        dianafcr        jetski91    \   mikesotto       sanders
aa7bq           diane           jgroff       \  milton          saucy
aacker          digna           jhill         \ mjwright        sawgal
aardvark        dillon          jillk           mkansgen        sbarnes
acarr         / ditomaso        jimfinly        mmadison        sbray
     \       /
     accounts                    [ALL of these are accounts]

[etc...]
```

What this process does is give you the names of all the common accounts on the UNIX you're on. Buffer this list and print it out. Exit the UNIX (type: exit) and try to Hack back using these accounts with the Account name as the password. i.e.:

```
UNIX(r) System V Release 4.0 (arthur)

login: jetski91
Password: jetski91 -- would not be shown
Login incorrect     /
login: mkansgen    /
Password: mkansgen
Last login: Sat Jan 27 12:34:31 from slip212m.vinue.net
Sun Microsystems Inc.   SunOS 5.4       Generic July 1994
You have new mail.
Sat Jan 27 12:41:04 MST 1996
/usr/users/mkansgen
arthurmkansgen/usr/users/mkansgen%
```

This is the 'Dumb User' Hack. Because a user was 'dumb' enough use his account name for his password, it was easily hacked, and now that dummy's account is your's. If the Dumb User's account has more privileges than yours (i.e. Permission to read the 'passwd' file), go back through the previously described methods and attempt to get the 'passwd' file. If the account has no greater privileges, keep the account for later trading on #hack and try and hack another account with more privileges.

If you've tried everything and you still haven't succeed in stealing a 'passwd' file, goto bed and thank God you don't have more troubles in life.

MACCRAC-ING

At this point you should have a processable 'passwd' file. This file should contain account entries with the encrypted password intact, and it should be saved as a plain text file. If these are completed you can proceed with using MACCRAC.

Now to use MACCRAC there a couple of operating mechanics to go over. Remember MACCRAC is a ported version of an IBM program, and since this is a BETA, its still a little buggy, and frills free. Basically, there are four main components of MACCRAC:

MacCrac.FAT--This is the main MacCrac application which processes and crack's UNIX 'passwd' files.

MacCrac.Log--This is the file where all information generated during the process off cracking a UNIX 'passwd' file is stored.

DICTIONARY--This is a dictionary file containing words MACCRAC will use to try and crack a 'passwd' file.

passwd--This the file that contains the UNIX account information.

Important notes on the above:

MacCrac.FAT

MACCRAC REQUIRES that ALL FILENAMES MUST BE AS THEY ARE LISTED ABOVE! There will be no dialogs to ask you which DICTIONARY or 'passwd' file you wish to use. MACCRAC Will look ONLY for a Dictionary file called DICTIONARY and a UNIX 'passwd' called passwd, AND it will only look for them in the immediate folder it is in, so make sure these files are in the same folder with MACCRAC.

Dictionary

The DICTIONARY is a standard Word Processing Dictionary as used by say, Microsoft Word. MACCRAC's Dictionary is somewhat larger than most Word Processoing Dictionaries with a size 2,431k. But other than it's size, it's no different. Dictionary files consist of alphabetized words with one word per line (carriage return) and no spaces. Heres a short sample of a DICTIONARY file:

A
a

aa
aal
aalii
aam
Aani
aardvark
aardwolf

Now, at 2,413k, MACCRAC's Dictionary is fairly large...although certainly not the largest. I personally have seen Dictionary files as large as 4 gigabytes! But normally you won't need a Dictionary that big. In fact the DICTIONARY file that comes with MACCRAC should be more than adequate. But if

you would like to use a larger Dictionary or would like to use a Dictionary of say, Foreign Words, or Star Trek Terms, or Dog Names, then you can either

make them or, find them on the internet.

In using these Dictionary files, it's important to remember that what ever name they're called when you find them, they MUST be RENAMED to DICTIONARY, and placed in the same Folder as MACCRAC in order to be used. If the Dictionary file is not called DICTIONARY, or is not in the same Folder as MACCRAC, it will not/cannot be used.

As a final note on Dictionaries, there is a program called 'Word List Maker'. This is a Drag&Drop program which allows you to Drag two or more Dictionary files on to it, and it will combine them into a single Dictionary

AND delete all duplicate entries. This is great for making custom, or more extensive DICTIONARY files for MACCRAC to use. Keep in mind though, that the

larger the Dictionary, the slower the process.

passwd

Well the 'passwd' file is what we spent the majority of this Tutorial discussing, so I shouldn't need to go into it much here. The most important thing to say about the 'passwd' file at THIS point is that included with MACCRAC is a file called 'passwd'; DELETE IT! This is just a sample file included with MACCRAC probably for Development or Testing purposes. It will do you no good. Replace it with your newly acquired 'passwd' file, and make sure this newly acquired file is called: passwd. Also make sure it's in the same Folder with MACCRAC

LET'S DO IT

Well, f you have your 'passwd' file, and you have whatever Dictionay file you're going to use, and all of the files are correctly named and placed in the same Folder with MACCRAC, then I guess you're ready, so lets do it!

For the sake of speed, and because you won't be able to use your computer anyway, I suggest Restarting your Mac with Exensions Off (even if you have RamCharger or RamDoubler). Once you've restarted, Double click on the

MACCRAC icon. If this is your first time running MACCRAC, just go up to 'Crack' in the menubar and select: Start Cracking!. The first thing you'll probably notice is that once you've started a Cracking Session you can't do anything else. Thats because MACCRAC hogs the processor. I would suggest starting a session around 11:00 pm and letting it run all night. By morning,

it should have cracked at least 40-50 accounts.

If for some reason you want or need to stop a session before an entire 'passwd' file is cracked, the only way to do it is with COMMAND-OPTION-ESC. Don't worry, any cracks MACCRAC has cracked to that point will be saved.

If you've already started Cracking a 'passwd' file but had to quit, you can pickup where you left off by going up to the 'CRACK' menubar and dragging down to Settings. Once in Settings select 'Recover session from "Point File"'. Now you can 'Start Cracking!' where ever you let off.

OUTRO

If you've let it run long enough, you should have passwords. At this point you're on your way to geting to "ROOT". The topic of Hacking "root" on UNIX has been addressed by any of a number of well written, informative and readily available T-Philes on UNIX Hacking. At this point I suggest you pursue them as this file will not address that topic (remember, this is a Tutorial on MACCRAC)

I'd like to thank Disorder, Voyager and the rest of TNo Crew for their incite and assistance. That's it for this one. Look for more oleBuzzard's T-Philes on the World's Greatest Underground Mac Board...