Program for decoding cisco pwd

The follow is a little tool to decode simple cisco pwd.

cisco_hack_tool.c

```
/* This code is originally from a Bugtraq post by
   Jared Mauch  . I patched it with an improved
   translation table by Janos Zsako
   -Fyodor (fyodor@insecure.org) */


#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, 0x48, 0x53 , 0x55, 0x42
};

char pw_str1[] = " password 7 ";
char pw_str2[] = "enable password 7 ";
char pw_str3[] = "ip ftp password 7 ";
char pw_str4[] = " ip ospf message-digest-key 1 md5 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':
                bzero(passwd, sizeof(passwd));
                                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(!strncmp(line, pw_str3, strlen(pw_str3)))
            pw_pos = strlen(pw_str3);
        if(!strncmp(line, pw_str4, strlen(pw_str4)))
            pw_pos = strlen(pw_str4);

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

        bzero(passwd, sizeof(passwd));
                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 if (pw_pos == strlen(pw_str2))
                                fprintf(out, "%s", pw_str2);
            else if (pw_pos == strlen(pw_str3))
                fprintf(out, "%s", pw_str3);
            else if (pw_pos == strlen(pw_str4))
                fprintf(out, "%s", pw_str4);


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

— return to gimbo wiki home page

From:
https://wiki.gimbo.org/ - **wiki.gimbo.org**

Permanent link:
https://wiki.gimbo.org/doku.php?id=public:cisco_hack_pwd

Last update: **13.03.2022 05:55**