

How to activate development kit...

The serial number that we have been given is... sco005715 and the activation key starts with eyxt..... This key does not work ☹

My assumption is that the numbers at the end of the key are incorrect...

When any SCO program is installed 99% of all programs are installed correctly but a small number are encrypted and can only be decrypted using the correct key..

To decrypt a call is made to **/etc/brand** passing the key and the program name. If everything matches then the program is decrypted...

In the development kit one or more programs in each "Development system package" is encrypted. For example in the LEX package lex, ncform and nrform are encrypted...

```

BUSDEV  Yes      2272  BUS Cross development 1
HELP    Yes      138   Help utility and related

Enter the package(s) to list
or enter q to return to the menu: LEX

Press <RETURN> each time the bell rings

/usr/lib/lex                      ./usr/bin/lex
/usr/lib/lex/ncform              ./tmp/init.lex
/usr/lib/lex/nrform

1. Install one or more packages
2. Remove one or more packages
3. List the available packages
4. List the files in a package
5. Install a single file
6. Select a new set to customize
7. Display current disk usage
8. Help

Select an option or enter q to quit: _
```

SO..... You kindly gave me a disk image which contained the GNU GCC compiler. This will allow us to compile a simple program...

First try to install the development system. This will fail when trying to enter the key but all the include files and lib files will be installed ☺

Next type in the following code... I'm sorry but this is on my machine at work under VMWARE and the floppy disk just doesn't work anymore under windows 7. I'll try to get the actual code onto a disk from my home machine if I can...

```
# more pass2.c
#include <stdio.h>
#include <assert.h>
#include <string.h>
#include <sys/types.h>
#include <sys/file.h>
main()
{
    int count,count2, count3, count4,count5, count6;
    char letter1 = 97;
    char letter2 = 97;
    char letter3 = 97;
    char letter4 = 97;
    char letter5 = 97;
    char letter6 = 97;
    letter3=97;
    for (count3=0; count3 <26; count3 ++)
    {
        letter2=97;
        for (count2=0; count2 <26; count2 ++)
        {
            letter1=97;
            for (count=0;count <26;count++)
            {
```

```
                printf("/etc/brand sco005715 eyxt%c%c%c /usr/bin/lex /usr/lib/le
x /usr/lib/lex/ncform /usr/lib/lex/nrform /tmp/init.lex\n",letter3,letter2, lett
er1);
                printf("echo 'activation = eyxt%c%c%c'\n", letter3, letter2, let
ter1);
                letter1++;
            }
            letter2++;
        }
        letter3++;
    }
}
# _
```

Once that is done compile and run the a.out file..

redirecting the output to an .sh file

ie a.out >b.sh

This will generate a huge (in xenix terms) file like this...

```
# more b.sh
/etc/brand sco005715 eyxtaaa /usr/bin/cflow
echo 'activation = eyxtaaa'
/etc/brand sco005715 eyxtaab /usr/bin/cflow
echo 'activation = eyxtaab'
/etc/brand sco005715 eyxtaac /usr/bin/cflow
echo 'activation = eyxtaac'
/etc/brand sco005715 eyxtaad /usr/bin/cflow
echo 'activation = eyxtaad'
/etc/brand sco005715 eyxtaae /usr/bin/cflow
echo 'activation = eyxtaae'
/etc/brand sco005715 eyxtaaf /usr/bin/cflow
echo 'activation = eyxtaaf'
/etc/brand sco005715 eyxtaag /usr/bin/cflow
echo 'activation = eyxtaag'
/etc/brand sco005715 eyxtaah /usr/bin/cflow
echo 'activation = eyxtaah'
/etc/brand sco005715 eyxtaai /usr/bin/cflow
echo 'activation = eyxtaai'
/etc/brand sco005715 eyxtaaj /usr/bin/cflow
echo 'activation = eyxtaaj'
/etc/brand sco005715 eyxtaak /usr/bin/cflow
echo 'activation = eyxtaak'
/etc/brand sco005715 eyxtaal /usr/bin/cflow
--More-- (0%)
```

What is been done is create a sh script which contains a /etc/brand call for every different combination of the eyXXXX string....

You just need to run this shell script and leave it a while... Mine took a couple of hours... You will get lots of errors on the screen; but don't worry.

Once this has been completed you should be able to run the cracked /usr/bin/lex program..

You need to replace the /usr/bin/lex for each program that needs to be decrypted and create and run the sh file for each.. You could in theory do them all at the same time but when I did it I just started with CC and kept going until I could compile a program... Sorry I did not keep a list while doing it; but you will get a detailed message if a file needed to be /etc/brand'ed..... But from memory just cc will allow you to compile so that might be all you need...