shc - ?? Linux Shell ??????? Shell ?????? binary??????????????

?????

- Makeself
- Add a Binary Payload to your Shell Scripts
- shar

?????

```
# tar -xzf shc-3.8.7.tgz
# cd shc-3.8.7
# make
# ./shc -v -f match
```

? ??????????match ?? shell script??????? match.x?

????????? Shell??????????????????????????????? static library ???????

```
export CFLAGS=-static && ./shc -r -f myshell
```

???????

```
export CFLAGS=-static && ./shc -e 19/10/2011 -r -f PIAF-install-ec2.sh
```

???????? static ? dynamic library

```
file myshell.x
```

???????

SYNOPSIS

shc [-e date] [-m addr] [-i iopt] [-x cmnd]
[-l lopt] [-ACDhTv] -f script

DESCRIPTION

shc creates a stripped binary executable version of the script specified with -f on the command line.

The binary version will get a .x extension appended and will usually be a bit larger in size than the original ascii code. Generated C source code is saved in a file with the extension .x.c

If you supply an expiration date with the -e option the com-piled binary will refuse to run after the date specified. The message "Please contact your provider" will be displayed instead. This message can be changed with the -m option.

You can compile any kind of shell script, but you need to supply valid -i, -x and -l options.

The compiled binary will still be dependent on the shell specified in the first line of the shell code (i.e. #!/bin/sh), thus shc does not create completely independent binaries.

shc itself is not a compiler such as cc, it rather encodes and encrypts a shell script and generates C source code with the added expiration capability. It then uses the system compiler to compile a stripped binary which behaves exactly like the original script. Upon execution, the compiled binary will decrypt and execute the code with the shell -c option. Unfortunatelly, it will not give you any speed improvement as a real C program would.

shc's main purpose is to protect your shell scripts from modification or inspection. You can use it if you wish to distribute your scripts but don't want them to be easily readable by other people.

OPTIONS

The command line options are:

-e date
Expiration date in dd/mm/yyyy format [none]

-m message
message to display upon expiration ["Please contact your provider"]

-f script_name
File name of the script to compile

-i inline_option
Inline option for the shell interpreter i.e: -e

-x comand
eXec command, as a printf format i.e: exec(\\'%s\\',@ARGV);

-l last_option
Last shell option i.e: --

-r Relax security. Make a redistributable binary which executes on different systems running the same operating system.

-v Verbose compilation

-D Switch on debug exec calls

-T Allow binary to be traceable (using strace, ptrace, truss, etc.)

-C Display license and exit

-A Display abstract and exit

-h Display help and exit

ENVIRONMENT VARIABLES

CC C compiler command [cc]

CFLAGS
C compiler flags [none]

EXAMPLES

Compile a script which can be run on other systems with the trace option enabled:

example% shc -v -r -T -f myscript

BUGS

The maximum size of the script that could be executed once compiled is limited by the operating system configuration parameter _SC_ARG_MAX (see sysconf(2))

AUTHOR

Francisco Rosales <frosal@fi.upm.es>