NAME

crysh — aes256-cbc encrypted shell

SYNOPSIS

crysh

DESCRIPTION

The **crysh** utility accepts input that is encrypted using the AES256-CBC cipher, decrypts it, and executes the resulting commands.

Note: **crysh** never invokes another shell. That is, it does *not* use system(3), popen(3), or the like.

OPTIONS

crysh does not support any command-line options.

DECRYPTION DETAILS

crysh reads bytes from stdin, attempts to decrypt the data and subsequently execute the commands found in the input.

The data read from stdin is expected to consist of the literal string "Salted__" followed by exactly 8 bytes of salt followed by the encrypted commands. Commands to be executed by **crysh** can be generated using the enc(1), command as shown in the *EXAMPLES* section.

Decryption of the data is done using AES 256bit CBC mode with a SHA1 digest with keying material derived from the passphrase using the EVP_BytesToKey(3) function. The passphrase may be specified via the *CRYSH_PASSWORD* environment variable. If that variable is unset, **crysh** will prompt the user on the controlling tty for the password.

EXECUTION DETAILS

As a non-interactive shell **crysh** does not support very complex commands. The decrypted string is split up into words at whitespace (blanks and tabs). The only operators supported by **crysh** are:

; Multiple commands may be provided in the input by separating them using a semicolon. A command is only executed if the previous command returned a successful return code.

>file Standard output from a command may be redirected to a file.

2>file Standard error from a command may be redirected to a file.

>>file Standard output from a command may be redirected and appended to a file.

2>>file Standard error from a command may be redirected and appended to a file.

EXAMPLES

To generate valid input for **crysh**, feed it into enc(1).

With that in mind, the following commands illustrate use of the tool.

```
$ export CRYSH_PASSWORD="bacon"
$ echo "date" | \
        openssl enc -aes-256-cbc | crysh
Mon Dec  4 15:43:01 EST 2017
$ echo "date; whoami" | \
        openssl enc -aes-256-cbc | crysh
Mon Dec  4 15:43:59 EST 2017
jschauma
$ echo "date >/tmp/out; whoami" | \
        openssl enc -aes-256-cbc | crysh
jschauma
$ cat /tmp/out
Mon Dec  4 15:44:34 EST 2017
$ echo "date >>/tmp/out ; ls -l /nowhere" | \
```

```
        openssl enc -aes-256-cbc | crysh
ls: /nowhere: No such file or directory
jschauma
$ cat /tmp/out
Mon Dec  4 15:44:34 EST 2017
Mon Dec  4 15:46:25 EST 2017
$ echo "ls /nowhere 2>/dev/null ; date; whoami;" | \
        openssl enc -aes-256-cbc | crysh
# No output: ls(1) failed; date(1), whoami(1) are not executed
$ echo "foo" | crysh
crysh: Unable to decrypt input.
$ echo $?
128
$ unset CRYSH_PASSWORD
$ echo "ls" | openssl enc -aes-256-cbc | crysh | wc -l
Password:
        38
$
```

ENVIRONMENT

crysh honors the following environment variables:

CRYSH_PASSWORD
: The password used to derive the key material from. If unset, prompt the user on the tty for the password.

PATH
: The user's PATH, to allow the user to specify non-absolute command-names.

EXIT STATUS

crysh exits with the return status of the last command it executed.

If **crysh** was unable to decrypt the data or another error was encountered, it will return 128.

SEE ALSO

blowfish(3), EVP_BytesToKey(3), EVP_EncryptInit(3)

HISTORY

This program was first assigned as a stand-alone programming assignment for the class "Advanced Programming in the UNIX Environment" at Stevens Institute of Technology in the Fall of 2017.

BUGS

Well, let's see...