

**EXDECRYPT(8)**

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>SYNOPSIS</b>	<b>1</b>
<b>2</b>	<b>DESCRIPTION</b>	<b>2</b>
<b>3</b>	<b>EXAMPLE</b>	<b>3</b>
<b>4</b>	<b>EXIT STATUS</b>	<b>4</b>
<b>5</b>	<b>BUGS</b>	<b>5</b>
<b>6</b>	<b>SEE ALSO</b>	<b>6</b>
<b>7</b>	<b>COPYING</b>	<b>7</b>

## Chapter 1

# SYNOPSIS

**exdecrypt** STRING [STRING...]

## Chapter 2

# DESCRIPTION

Program decrypts one or many **STRING** value passed on command line. In the result program prints to **stdout** clear string value. The debug is configured in standard way as for all other binaries via *ndrxdebug.conf* or Common Config.

The encrypted value of the string can be made by **exencrypt(8)** utility.

Encryption keys which are used for string encryption are provided either by using built-in algorithm (username+hostname) hashed with SHA1. Or from vendor specific loaded plugin.

The encryption principles allows to encrypt the sensitive data and store on disk in PCI/DSS compatible way. And **exdecrypt** can be used for scripts to get in runtime the clear values needed.

In case of decrypt failure, logs needs to be analyzed. There could be problems with base64 string or invalid encryption key. In case decrypt process finds the binary zero (0x00 byte) in output string, error is generated automatically, because this value is forbidden in string decryption.

## Chapter 3

# EXAMPLE

Single string decryption:

```
$ exdecrypt AAAAC196L/d4Sj4OC8cSZh2492I=  
HELLO WORLD
```

Two string decryption:

```
$ exdecrypt AAAABbBtKbtIv9BXelSioTYi5xw= AAAABRaPhXxibr6ZktcZ6S7li50=  
HELLO  
WORLD
```

## Chapter 4

# EXIT STATUS

**0**      Success

**1**      Failure

## Chapter 5

# BUGS

Report bugs to [support@mavimax.com](mailto:support@mavimax.com)



## Chapter 6

## SEE ALSO

`exencrypt(8)`, `ex_env(5)`, `ndrxdebug.conf(5)`

## **Chapter 7**

# **COPYING**

© Mavimax, ltd.