hose

Name

`hose` — Control all manner of rubberhoses

A command within the Marutukku encryption system

Synopsis

hose [*`global-options`*] {*`command`*} [*`local-options`*] [*`args`*...]

GLOBAL OPTIONS

`-E`

Disable wait for entropy (useful for batch tests)

`-f`

Force through errors where possible

`-L`

Disable memory locking

`-q`

Quiet

`-Q`

Quick and quiet, enable `-d0`, `-ELQTW` and `-P0` options

`-T`

Disable reseting file time stamps to epoch

`-W`

Disable memory wiping (useful for batch tests)

`-d` *`level`*

Set debug level to ’level’

`-P` *`level`*

Set self-psychoanalysis rigour to ’level’

COMMANDS

aspectinfo

Dump informative info about aspect

attachextent

Attach extent

bindaspect

Binds aspect to device

changepass

Change keying for aspect

decryptaspect

Decrypt from Aspect to output

decryptfile

Conventional file decryption

dekeyaspect

Dekey aspect

detachextent

Detach previously attached extent

encryptaspect

Encrypt from input to Aspect

encryptfile

Conventional file encryption

example

Show example usage for command

help

General help or help on a particular command

info

Display configuration

keyaspect

Key aspect (needs an attached extent)

list

List available ciphers, commands or remaps

newaspect

Create new aspect for keymap

newextent

Create new extent

newkeymap

Create new keymap file

psycho

Visit the psychiatrist

remapinfo

Dump remap information

speeds

Test cipher speeds

sync

Sync hose daemon pending writes to disk

terminate

Terminate hose daemon

unbindaspect

Unbind aspect from device

wipe

Wipe file or extent

COMMAND DESCRIPTIONS

aspectinfo

aspectinfo [`-a` *`aspect`*][*`keymap`*]

`-a` *`aspect`*

Use aspect number ’aspect’

`Dump informative info about aspect`

EXAMPLE

Example 1. Example aspectinfo

```
  $   hose aspectinfo -a 0 maru.keymap

Agitating master key with cast-
cbc key generator over 123562 iterations...
Aspect 1 passphrase ("." to end):
Aspect 2 passphrase ("." to end):
Aspect 3 passphrase ("." to end):
Aspect 4 passphrase ("." to end):
Aspect 5 passphrase ("." to end):
Aspect 0:
        Lattice Cipher: cast-cbc
        Block Cipher:   idea-cbc
        Start:          0
        Blocks:         64
```

attachextent

attachextent [-B][-a *aspect*][-R *path*][*keymap* [*extent* [*device*]]]

-B

Disable pro-active block reallocation (bmap)

-a *aspect*

Use aspect number ’aspect’

-R *path*

Rendezvous with hosed AF_UNIX socket at ’path’

Attach extent

EXAMPLE

Example 2. Example attachextent

```
  $   hose attachextent -a 0 -R /tmp/rendezvous
```

bindaspect

bindaspect [-R][-a *aspect*]

-R *path*

Rendezvous with hosed AF_UNIX socket at ’path’

`-a` *`aspect`*

Use aspect number ’aspect’

```
Binds aspect to device
```

EXAMPLE

Example 3. Example bindaspect

```
  $   hose bindaspect -a 0 -R /tmp/rendezvous
```

changepass

changepass `[-a` *`aspect`*`][`*`keymap`*`]`

`-a` *`aspect`*

Use aspect number ’aspect’

```
Change keying for aspect
```

EXAMPLE

Example 4. Example changepass

```
  $   hose changepass -a 0 maru.keymap
```

decryptaspect

decryptaspect [`-a` *`aspect`*][`-o` *`file`*][`-s` *`blocks`*][*`keymap`* [*`extent`* [*`output`*]]]

`-a` *`aspect`*

Use aspect number ’aspect’

`-o` *`file`*

Output operation to ’file’

`-s` *`blocks`*

Size in ’blocks’

`Decrypt from Aspect to output`

EXAMPLE

Example 5. Example decryptaspect

```
  $   hose decryptaspect -a 0 -o maru.out
```

```
Agitating master key with cast-
cbc key generator over 123562 iterations...
Aspect 1 passphrase ("." to end):
Aspect 2 passphrase ("." to end):
Aspect 3 passphrase ("." to end):
Aspect 4 passphrase ("." to end):
Aspect 5 passphrase ("." to end):
decrypted 64 blocks from maru.extent to maru.out
```

decryptfile

decryptfile [`-3` *`cipher`*][`-i` *`file`*][`-o` *`file`*][`-V` *`iv`*][*`input`* [*`output`*]]

`-3` *`cipher`*

Cipher for block encryption/decryption

`-i` *`file`*

Take input from ’file’

`-o` *`file`*

Output operation to ’file’

`-V` *`iv`*

Use ’iv’ (in hex) as the initialisation vector

`Conventional file decryption`

EXAMPLE

Example 6. Example decryptfile

```
  $   hose decryptfile -3 idea-cbc -i maru.ciphertext -
V 0xadeadfedbabecafe -o maru.out

Passphrase:
```

dekeyaspect

dekeyaspect [`-a` *`aspect`*][`-R` *`path`*]

`-a` *`aspect`*

Use aspect number ’aspect’

`-R` *`path`*

Rendezvous with hosed AF_UNIX socket at ’path’

`Dekey aspect`

EXAMPLE

Example 7. Example dekeyaspect

```
  $   hose dekeyaspect -a 0 -R /tmp/rendezvous
```

detachextent

detachextent [`-R` *`path`*]

`-R` *`path`*

Rendezvous with hosed AF_UNIX socket at ’path’

`Detach previously attached extent`

EXAMPLE

Example 8. Example detachextent

```
  $   hose detachextent -R /tmp/rendezvous
```

encryptaspect

encryptaspect `[-B]` `[-a` *`aspect`*`]` `[-i` *`file`*`]` `[-s` *`blocks`*`]` `[`*`keymap`* `[`*`extent`* `[`*`input`*`]]]`

`-B`

Disable pro-active block reallocation (bmap)

`-a` *`aspect`*

Use aspect number ’aspect’

`-i` *`file`*

Take input from ’file’

`-s` *`blocks`*

Size in ’blocks’

`Encrypt from input to Aspect`

EXAMPLE

Example 9. Example encryptaspect

```
  $   hose encryptaspect -a 0 -i maru.plaintext maru.keymap

Agitating master key with cast-
cbc key generator over 123562 iterations...
Aspect 1 passphrase ("." to end):
Aspect 2 passphrase ("." to end):
Aspect 3 passphrase ("." to end):
Aspect 4 passphrase ("." to end):
Aspect 5 passphrase ("." to end):
encrypted 32 blocks from maru.plaintext to maru.extent
```

encryptfile

encryptfile [-3 *cipher*][-i *file*][-o *file*][-V *iv*][*input* [*output*]]

-3 *cipher*

Cipher for block encryption/decryption

-i *file*

Take input from ’file’

-o *file*

Output operation to ’file’

-V *iv*

Use ’iv’ (in hex) as the initialisation vector

```
Conventional file encryption
```

EXAMPLE

Example 10. Example encryptfile

```
  $   hose encryptfile -3 idea-cbc -i maru.plaintext -
o maru.ciphertext

Passphrase:
```

example

example [-m] [command]

`-m`

Minimal output

```
Show example usage for command
```

EXAMPLE

Example 11. Example example

```
  $   hose example newaspect

Example:
        ../hose/hose newaspect -2 cast-cbc -3 idea-cbc -a 0 -
s 64 -t 1 maru.keymap
```

help

help [-S] [commands | options | *command*]

-S

SGML output

General help or help on a particular command

EXAMPLE

Example 12. Example help

```
  $   hose help newkeymap

Usage: ../hose/hose [-EfLqQTW] [-d level] [-
P level] newkeymap [-1 cipher] [-A aspects] [-b bytes] [-
c blocks] [-D depth] [-r remap] [-s blocks] [keymap]
Description:
        Create new keymap file
Local options:
        -
1 cipher           Cipher for encryption/decryption of keys
        -A aspects          Max number of usable aspects
        -b bytes            Block size in bytes
        -c blocks           Largest aspect size in blocks
        -D depth            Depth of block key lattice
        -r remap            Use remap type 'remap'
        -s blocks           Size in ’blocks’
Global options:
        -E                  Disable wait for entropy (use-
ful for batch tests)
        -f                  Force through errors where possible
        -L                  Disable memory locking
        -q                  Quiet
        -Q                  Quick and quiet, enable -d0, -
ELQTW and -P0 options
        -
T                  Disable reseting file time stamps to epoch
        -W                  Disable memory wiping (use-
ful for batch tests)
        -d level            Set debug level to ’level’
        -P level            Set self-
psychoanalysis rigour to ’level’
Example:
        ../hose/hose newkeymap -1 cast-cbc -A 6 -b 8192 -c 32 -
r bmap -s 128 maru.keymap
```

info

info [`-l` *seconds*][`-I` *seconds*][`-x` *msec*][*keymap* [*extent* [*device*]]]

`-I` *seconds*

Autodetach after ’seconds’ of idleness

`-l` *seconds*

Autodetach after ’seconds’ since attach

`-x` *msec*

Use ’msec’ miliseconds between cipher state xors

Display configuration

EXAMPLE

Example 13. Example info

```
  $   hose info maru.keymap

        Major Version:  2
        Minor Version:  1
        Key Cipher:
                name cast-cbc
                  cipher_num    1
                  key_size      128 bits
                  block_size    64 bits
                  state/ksch    132 bytes
        Key Iterations: 123562
        Blocks:         128
        Block Size:     8192
        Lattice Depth:  32 (4194304k addressable bytes)
        Aspects:        6
        Remap Type:     bmap
        Checksum:       0xabf0bfb5
        Maru device:    /dev/maru0
        Maru extents:   maru.extent
        Maru IV/SALT:   maru.keymap
        Life time:      28800 (seconds)
        Idle time:      1800 (seconds)
        XOR cycle:      500 (mili seconds)
```

keyaspect

keyaspect `[-R]` `[-a` *`aspect`*`]`

`-R` *`path`*

Rendezvous with hosed AF_UNIX socket at ’path’

`-a` *`aspect`*

Use aspect number ’aspect’

`Key aspect (needs an attached extent)`

EXAMPLE

Example 14. Example keyaspect

```
  $   hose keyaspect -a 0 -R /tmp/rendezvous
```

list

list `[-m]` [ciphers | commands | remaps]

`-m`

Minimal output

`List available ciphers, commands or remaps`

EXAMPLE

Example 15. Example list

```
$   hose list ciphers

        name xor
          cipher_num    16
          key_size      256 bits
          block_size    0 bits (stream cipher)
          state/ksch    4 bytes
        name bcopy
          cipher_num    17
          key_size      256 bits
          block_size    0 bits (stream cipher)
          state/ksch    4 bytes
        name idea-cbc
          cipher_num    2
          key_size      128 bits
          block_size    64 bits
          state/ksch    432 bytes
        name cast-cbc
          cipher_num    1
          key_size      128 bits
          block_size    64 bits
          state/ksch    132 bytes
        name ssl-blowfish-cbc
          cipher_num    6
          key_size      448 bits
          block_size    64 bits
          state/ksch    8196 bytes
        name ssl-rc2-cbc
          cipher_num    12
          key_size      128 bits
          block_size    64 bits
          state/ksch    8196 bytes
        name ssl-rc4
```

```
  cipher_num    13
  key_size      256 bits
  block_size    0 bits (stream cipher)
  state/ksch    8196 bytes
name ssl-rc5-cbc
  cipher_num    15
  key_size      128 bits
  block_size    64 bits
  state/ksch    8196 bytes
name ssl-idea-cbc
  cipher_num    11
  key_size      128 bits
  block_size    64 bits
  state/ksch    8196 bytes
name ssl-des-cbc
  cipher_num    7
  key_size      64 bits (56 bits real)
  block_size    64 bits
  state/ksch    8196 bytes
name ssl-des-ede-cbc
  cipher_num    8
  key_size      128 bits (112 bits real)
  block_size    64 bits
  state/ksch    8196 bytes
name ssl-des-ede3-cbc
  cipher_num    9
  key_size      192 bits (168 bits real)
  block_size    64 bits
  state/ksch    8196 bytes
name ssl-desx-cbc
  cipher_num    10
  key_size      192 bits (168 bits real)
  block_size    64 bits
  state/ksch    8196 bytes
name ssl-cast-cbc
  cipher_num    14
  key_size      128 bits
```

```
          block_size    64 bits
          state/ksch    8196 bytes
        name rc16
          cipher_num    4
          key_size      256 bits
          block_size    0 bits (stream cipher)
          state/ksch    131080 bytes
```

newaspect

newaspect `[-2` *`cipher`*`][-3` *`cipher`*`][-a` *`aspect`*`][-O` *`block`*`][-s` *`blocks`*`][-t` *`time`*`][`*`keymap`*`]`

`-2` *`cipher`*

Cipher for generation of block keys

`-3` *`cipher`*

Cipher for block encryption/decryption

`-a` *`aspect`*

Use aspect number ’aspect’

`-O` *`block`*

Start block range at offset ’block’

`-s` *`blocks`*

Size in ’blocks’

`-t` *`time`*

Use ’time’ seconds of key cycle agitation

Create new aspect for keymap

EXAMPLE

Example 16. Example newaspect

```
  $   hose newaspect -2 cast-cbc -3 idea-cbc -a 0 -s 64 -t 1 maru.keymap

Generating 11128 pseudo-cryptographically random bytes for aspect 0 erasure
........................................
Generating 32 pseudo-cryptographically random bytes for aspect 0 key salt
........................................
Generating 104 pseudo-cryptographically random bytes for aspect 0 cycle
........................................
Generating 32 cryptographically random bytes for aspect 0 master key
........................................
Generating 32 cryptographically random bytes for aspect 0 info key
........................................
Generating 32 cryptographically random bytes for aspect 0 remap master key
........................................
Generating 104 pseudo-cryptographically random bytes for cycle salt
........................................

Agitating cast-cbc key generator state for 1 second...
123562 cast-cbc agitations (123562 per second)
Generating 4 pseudo-cryptographically random bytes for aspect information salt
```

```
........................................
Generating 64 pseudo-
cryptographically random bytes for primary lattice key salts
........................................
Generating 2048 pseudo-
cryptographically random bytes for subkey lattice IVs
........................................
Generating 8192 pseudo-
cryptographically random bytes for block whitener
........................................

Clearing key artifacts
```

newextent

newextent `[-1` *`cipher`*`][-w` *`rounds`*`][-s` *`blocks`*`][-b` *`bytes`*`][`*`keymap`*`]`
`[`*`extent`*`]`

`-1` *`cipher`*

Cipher for encryption/decryption of keys

`-b` *`bytes`*

Block size in bytes

`-s` *`blocks`*

Size in ’blocks’

`-w` *`rounds`*

Apply ’rounds’ worth of wiping

```
Create new extent
```

EXAMPLE

Example 17. Example newextent

```
  $   hose newextent -1 cast-cbc -w 0 -s 128 -b 8192

hose: Warning: creating extent using Unix file holes. Such ex-
tents are *not* crypto-deniable.
Extent creation complete (1048576 bytes)
```

newkeymap

newkeymap `[-1` *`cipher`*`][-A` *`aspects`*`][-b` *`bytes`*`][-c` *`blocks`*`][-D` *`depth`*`]` `[-r` *`remap`*`][-s` *`blocks`*`][`*`keymap`*`]`

`-1` *`cipher`*

Cipher for encryption/decryption of keys

`-A` *`aspects`*

Max number of usable aspects

`-b` *`bytes`*

Block size in bytes

`-c` *`blocks`*

Largest aspect size in blocks

`-D` *`depth`*

Depth of block key lattice

`-r` *remap*

Use remap type ’remap’

`-s` *blocks*

Size in ’blocks’

Create new keymap file

EXAMPLE

Example 18. Example newkeymap

```
  $   hose newkeymap -1 cast-cbc -A 6 -b 8192 -c 32 -r bmap -s 128 maru.keymap

Generating 64496 pseudo-cryptographically random bytes for keymap erasure
........................................
Maru keymap generation complete.
Saving Maru Keymap as "maru.keymap"
* MAKE AT LEAST TWO BACKUPS of this file. If a single bit sells out to the dark
  forces of entropy, your entire maru ciphertext extent will follow suit!
```

psycho

psycho

Visit the psychiatrist

EXAMPLE

Example 19. Example psycho

```
  $   hose -d 9 -P 9 psycho
```

```
hose: psychoanalysis: checking that all command options have help...
hose: psychoanalysis: checking that all options have commands that use them...
hose: psychoanalysis: sizeof (m_u64) == 8... passed
hose: psychoanalysis: sizeof (m_u32) == 4... passed
hose: psychoanalysis: sizeof (m_u16) == 2... passed
hose: psychoanalysis: sizeof (m_u8) == 1... passed
hose: psychoanalysis: sizeof (int) >= 4... passed
hose: psychoanalysis: hton8(0x12) == 0x12... passed
hose: psychoanalysis: hton16(0x1234) == 0x3412... passed
hose: psychoanalysis: hton32(0x12345678) == 0x78563412... passed
hose: psychoanalysis: hton64(0x1122334455667788) == 0x8877665544332211... passed
hose: psychoanalysis: MAX_PASSPHRASE >= MIN_PASSPHRASE... passed
hose: psychoanalysis: MAX_IV == MAX_CIPHER_BLOCK... passed
hose: psychoanalysis: MAX_CIPHER_BLOCK == 8... passed
hose: psychoanalysis: sizeof (maruPass) == MAX_PASSPHRASE... passed
hose: psychoanalysis: sizeof (maruKey) == MAX_KEY... passed
hose: psychoanalysis: sizeof (maruIV) == MAX_IV... passed
hose: psychoanalysis: sizeof (maruBlock) == MAX_CIPHER_BLOCK... passed
hose: psychoanalysis: blockAligned(maruCycle)... passed
hose: psychoanalysis: blockAligned(maruAspectInfo)... passed
hose: psychoanalysis: examining "/home/proff"... passed
hose: psychoanalysis: examining "/etc/mtab"... passed
hose: psychoanalysis: xor auto test vector in == out, ply = 1... passed
hose: psychoanalysis: xor auto test vector in != out, ply = 1... passed
```

```
hose: psychoanalysis: bcopy auto test vec-
tor in == out, ply = 1... passed
hose: psychoanalysis: bcopy auto test vec-
tor in != out, ply = 1... passed
hose: psychoanalysis: idea-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: idea-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: cast-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: cast-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-blowfish-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-blowfish-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-rc2-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-rc2-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-
rc4 auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-
rc4 auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-rc5-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-rc5-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-idea-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-idea-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-des-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-des-
cbc auto test vector in != out, ply = 1... passed
```

```
hose: psychoanalysis: ssl-des-ede-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-des-ede-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-des-ede3-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-des-ede3-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-desx-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-desx-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: ssl-cast-
cbc auto test vector in == out, ply = 1... passed
hose: psychoanalysis: ssl-cast-
cbc auto test vector in != out, ply = 1... passed
hose: psychoanalysis: rc16 auto test vec-
tor in == out, ply = 1... passed
hose: psychoanalysis: rc16 auto test vec-
tor in != out, ply = 1... passed
Passed the maru DSM, level 9
```

remapinfo

remapinfo [*keymap*]

```
Dump remap information
```

EXAMPLE

Example 20. Example remapinfo

```
  $   hose remapinfo maru.keymap

Agitating master key with cast-
cbc key generator over 123562 iterations...
Aspect 1 passphrase ("." to end):
Aspect 2 passphrase ("." to end):
Aspect 3 passphrase ("." to end):
Aspect 4 passphrase ("." to end):
Aspect 5 passphrase ("." to end):
Remap type: bmap        Dynamically distribute blocks to aspects
Block -> Aspect ownership map (* = collision):
0       ....................................................................
46      ..........................................................
```

speeds

speeds [-eS][-a *aspect*][*keymap* [*extent*]]

-e

Use entire maru encryption path for speed calculations

-S

SGML output

-a *aspect*

Use aspect number ’aspect’

```
Test cipher speeds
```

EXAMPLE

Example 21. Example speeds

```
  $   hose speeds

Cipher           | setkey/s |  512k/s | 1024k/s | 2048k/s | 4096k/s | 8192k/
=================+==========+=========+=========+=========+=========+=======
xor              | 28395307 |  688120 |  735121 |  756022 |  769100 |  76662
-----------------+----------+---------+---------+---------+-----
----+--------
bcopy            | 28817071 |  453326 |  587886 |  685740 |  746232 |  77912
-----------------+----------+---------+---------+---------+-----
----+--------
idea-
cbc         |  1801295 |    4339 |    4193 |    4166 |    4196 |    4224
-----------------+----------+---------+---------+---------+-----
----+--------
idea-cbc-
D       |    31245 |    4272 |    4150 |    4136 |    4188 |    4168
-----------------+----------+---------+---------+---------+-----
----+--------
cast-
cbc         |   462943 |   12883 |   13008 |   12980 |   13120 |   12888
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-blowfish-
cbc |     4048 |   14312 |   14369 |   14462 |   14444 |   14240
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-rc2-
cbc      |   124204 |    4211 |    4200 |    4226 |    4248 |    4248
```

```
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-
rc4          |   117164 |   23569 |   29444 |   32970 |   35716 |   36976
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-rc5-
cbc      |   294663 |   10822 |   10803 |   10914 |   10812 |   10800
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-rc5-cbc-
D    |   294667 |   18364 |   18599 |   18600 |   18472 |   18592
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-idea-
cbc     |  1332064 |    5902 |    6203 |    6282 |    6324 |    6408
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-idea-cbc-
D   |    31311 |    5863 |    6317 |    6390 |    6280 |    6216
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-des-
cbc      |   455926 |    5507 |    5512 |    5536 |    5524 |    5504
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-des-ede-
cbc  |   222911 |    1962 |    1954 |    1964 |    1956 |    1968
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-des-ede3-
cbc |   163273 |    1959 |    1948 |    1958 |    1928 |    1960
-----------------+----------+---------+---------+---------+-----
----+--------
ssl-desx-
cbc     |   456771 |    5508 |    5384 |    5420 |    5424 |    5400
```

```
-----------------+----------+----------+----------+----------+-----
----+--------
ssl-cast-
cbc      |   340860 |     9570 |     9515 |     9530 |     9624 |     9536
-----------------+----------+----------+----------+----------+-----
----+--------
rc16             |      424 |    30795 |    30680 |    30936 |    30864 |    3091
-----------------+----------+----------+----------+----------+-----
----+--------
```

sync

sync [`-R` *`path`*]

`-R` *`path`*

Rendezvous with hosed AF_UNIX socket at ’path’

```
Sync hose daemon pending writes to disk
```

EXAMPLE

Example 22. Example sync

```
  $   hose sync -R /tmp/rendezvous
```

terminate

terminate [`-R` *`path`*]

`-R` *`path`*

Rendezvous with hosed AF_UNIX socket at ’path’

`Terminate hose daemon`

EXAMPLE

Example 23. Example terminate

```
  $   hose terminate -R /tmp/rendezvous
```

unbindaspect

unbindaspect [`-a` *`aspect`*][`-R` *`path`*]

`-a` *`aspect`*

Use aspect number ’aspect’

`-R` *`path`*

Rendezvous with hosed AF_UNIX socket at ’path’

`Unbind aspect from device`

EXAMPLE

Example 24. Example unbindaspect

```
  $   hose unbindaspect -a 0 -R /tmp/rendezvous
```

wipe

wipe [-1 *cipher*][-b *bytes*][*extent*]

-1 *cipher*

Cipher for encryption/decryption of keys

-b *bytes*

Block size in bytes

Wipe file or extent

EXAMPLE

Example 25. Example wipe

```
  $   hose wipe -1 rc16 maru.extent
```

```
Generating 32 cryptographically random bytes for rc16 era-
sure key
........................................
```

```
Erasing maru.extent (and mir-
rors) with rc16(/dev/random): pass 1 192512/1048576
Erasing maru.extent (and mir-
rors) with rc16(/dev/random): pass 1 522240/1048576
Erasing maru.extent (and mir-
rors) with rc16(/dev/random): pass 1 843776/1048576
Erasing maru.extent (and mir-
rors) with rc16(/dev/random): pass 1 1048576/1048576
```

ENVIROMENTAL VARIABLES

MARU_PASSPHRASE

Use the contents of this variable instead of ever prompting for a passphrase.

MARU_PASSPHRASE_n

Use the contents of this variable instead of prompting for a passphrase for aspect_n. This variable is dominant over MARU_PASSPHRASE

AUTHORS

Julian Assange

COPYRIGHT