

UNIX

UNIX/C

#82#

C

automatic

C

func()

x

```
void func() {
    int x; // declares an integer on the stack
    ...
}
```

func()

heap

```
void func() {
    int *x = (int *) malloc(sizeof(int));
    ...
}
```

int * x

malloc()
NULL

#&Z\$ _ S^aUfi

malloc

NULL

man

malloc

man malloc

```
#include <stdlib.h>
```

```
...
```

```
void *malloc(size_t size);
```

stdlib.h

malloc

C

C

malloc()

malloc()

malloc

size_t

10

```
double *d = (double *) malloc(sizeof(double));
```

sizeof()

double

malloc()

sizeof()

C

8

double

malloc()

sizeof()

sizeof()

```
int *x = malloc(10 * sizeof(int));
```

```
printf("%d\n", sizeof(x));
```

10

sizeof()

4

32

8

64

C NULL

\$

sizeof()

sizeof()

```
int x[10];
printf("%d\n", sizeof(x));
```

40

malloc(strlen(s) + 1)

strlen()

1

sizeof()

malloc()

void

C

cast

malloc()

double

malloc()

#include <stdlib.h>

free()

```
int *x = malloc(10 * sizeof(int));
...
free(x);
```

malloc()

#include <stdlib.h>

malloc() free()

C

automatic

memory management

malloc()

new

garbage collector

strcpy(dst, src)

```
char *src = "hello";  
char *dst;          // oops! unallocated  
strcpy(dst, src); // segfault and die
```

segmentation fault

!=

```
char *src = "hello";  
char *dst = (char *) malloc(strlen(src) + 1);  
strcpy(dst, src); // work properly
```

strdup()

strdup man

buffer overflow

```
char *src = "hello";  
char *dst = (char *) malloc(strlen(src)); // too small!  
strcpy(dst, src); // work properly
```

malloc

[W06]

malloc

malloc()

uninitialized read

memory leak

free()

C

dangling pointer

free()

malloc()

double free

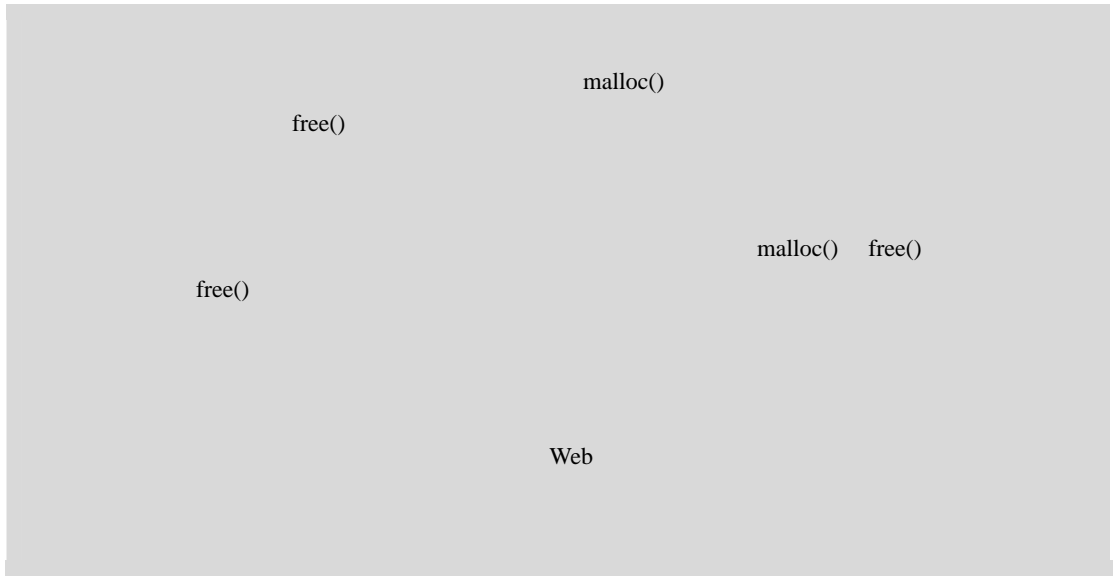
~~Xi/Wfi~~

free()

free()

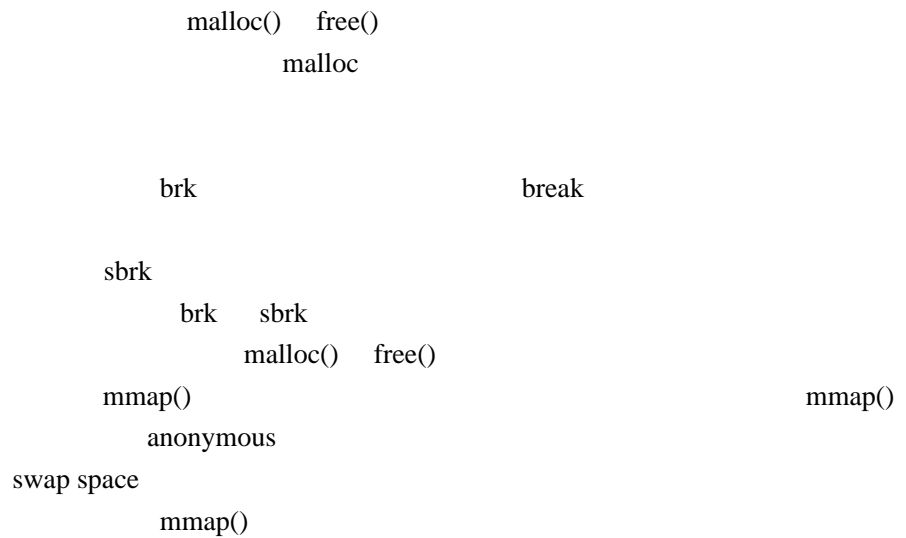
malloc()

invalid free



purify [HJ92] valgrind [SN05]

#87



#82(

calloc()

14.4

realloc()

realloc()

#82)

API

C

[KR88]

Stevens [SR05]

7

Novark

[N+07]

[HJ92] Purify: Fast Detection of Memory Leaks and Access Errors

R. Hastings and B. Joyce USENIX Winter 92

Purify

Purify

[KR88] The C Programming Language Brian Kernighan and Dennis Ritchie Prentice-Hall 1988

C

C

[N+07] Exterminator: Automatically Correcting Memory Errors with High Probability Gene Novark, Emery D. Berger, and Benjamin G. Zorn

PLDI 2007

C C ++

[SN05] Using Valgrind to Detect Undefined Value Errors with Bit-precision

J. Seward and N. Nethercote USENIX 05

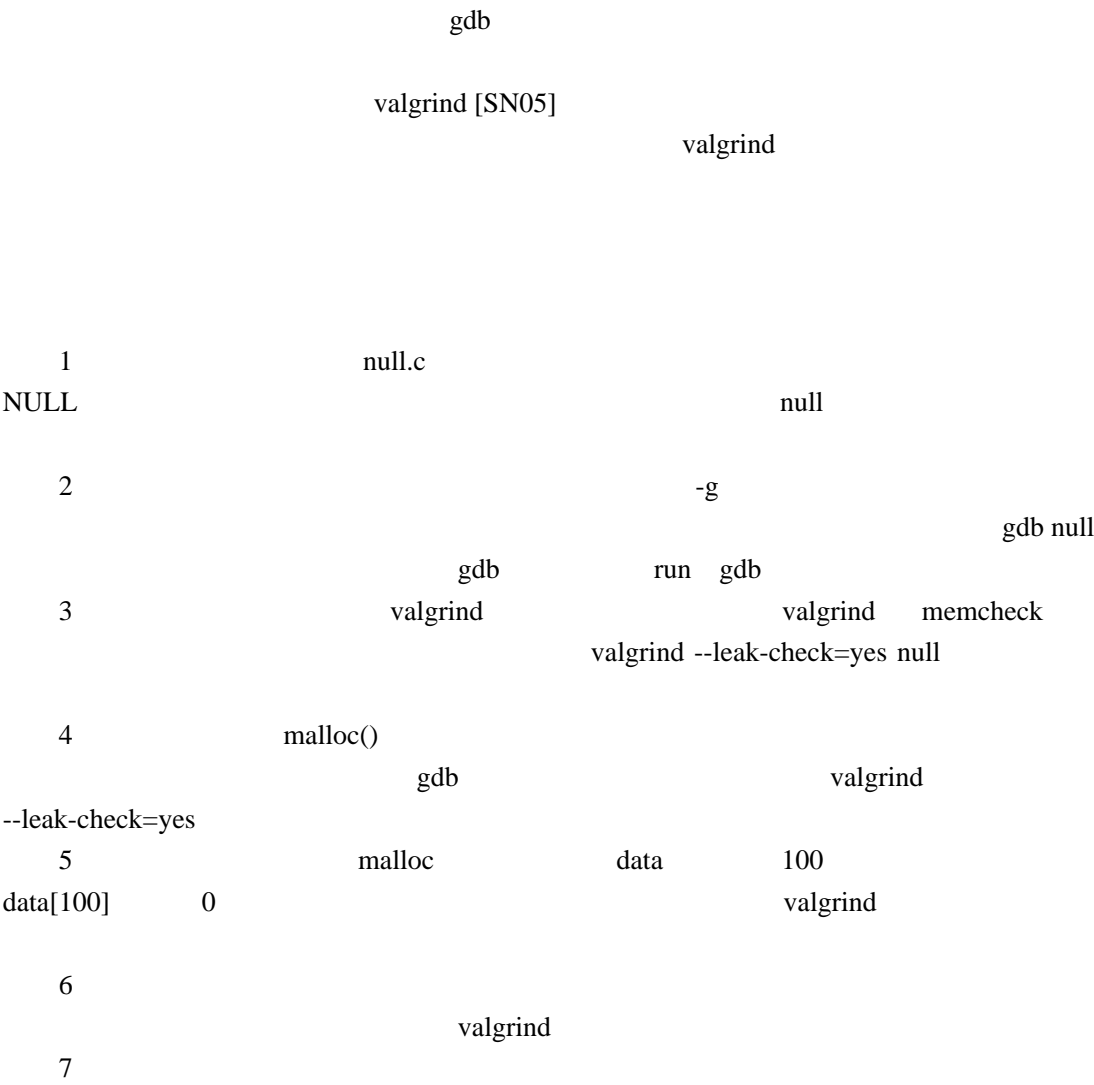
valgrind

[SR05] Advanced Programming in the UNIX Environment

W. Richard Stevens and Stephen A. Rago Addison-Wesley, 2005

C

[W06] Survey on Buffer Overflow Attacks and Countermeasures Tim Werthman



8
realloc()
realloc() valgrind

9 gdb valgrind
UNIX C