



sureSEC
SECURING THE SOURCE

Suresec security advisory 11
1th March 2006
CVE ID: CVE-2005-1126, CVE-2005-2752

Table of Contents

MacOSX vm_allocate() / malloc() integer overflow.....	2
About vm_allocate() / malloc():.....	2
Vulnerability summary:	2
Impact:	3
Affected versions:	3
Suggested Recommendations:	3
Credits:	3
About us:	3

MacOSX vm_allocate() / malloc() integer overflow

About vm_allocate() / malloc():

The vm_allocate function is responsible for allocating a new block of virtual memory for the calling process. Physical memory is not allocated until the pages are used by the process.

The allocated memory can be slightly larger than the requested size due to rounding up the page to be modulo of getpagesize().

The malloc() function is used to allocate blocks from the virtual memory allocation for use by the process.

Vulnerability summary:

The vm_allocate() mach syscall in the xnu / Darwin kernel contains an integer overflow¹ when calculating the size of the block of memory which is required.

The following code was responsible for this error:

```
#define PAGE_MASK (PAGE_SIZE - 1)
#define PAGE_SIZE vm_page_size
#define vm_map_round_page(x) (((vm_map_offset_t)(x) + \
PAGE_MASK) & ~((signed)PAGE_MASK))
```

...

```
map_size = vm_map_round_page(size);
if (map_addr == 0)
    map_addr += PAGE_SIZE;
```

As you can see from the code, the vm_map_round_page() macro, when passed a large value such as negative one (-1), will wrap back around to 0. The vm_allocate() call handles this situation by adding the size of a single page to size required. This leaves a single page allocated, and a pointer to this buffer is returned successfully. However an allocation considerably smaller than the length passed to the syscall has occurred.

1 <http://www.phrack.org/60/p60-0x0a.txt>

The `vm_allocate()` function is used by the `malloc()` call, from `libSystem` in user space, in order to allocate large amounts of memory. When a user supplied value is passed to the `malloc()` function it is therefore possible to wrap the value and cause an inconsistency in program state.

Impact:

This vulnerability exists in both `malloc()` calls in user space and `vm_allocate()` calls in the kernel. Due to this, a large number of applications in user space are exploitable for remote or local compromise of the machine. Also various system calls can be exploited in kernel space to escalate privileges to root.

Affected versions:

Early versions of Mac OSX Tiger (10.4) were affected by this vulnerability. It was silently fixed in the kernel however it was addressed in `libSystem` in the latest security patch APPLE-SA-2006-03-01 Security Update 2006-001.

Suggested Recommendations:

Apple have been informed about this vulnerability and installing the latest security update should mitigate any threat.

Credits:

Neil Archibald has been credited with discovering this vulnerability.

About us:

Suresec Ltd is a global service provider of Internet security solutions and consultancy with unmatched quality from our world class consultancy practice.

Our consultants have pioneered in the field of security research and have closely worked with leading software companies and service

providers to mitigate risks and fix a number of critical vulnerabilities, suresec also works closely with a number of open source companies to provide them with a source code auditing and technical consultancy. We have a strong team consultants spread across Europe, the United States of America And Australia specializing in security consulting.