

[UNIX] Userland Can Access Linux Kernel Memory (do_brk() Argument Bound Checking)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-12/0015.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 12/03/03

To: list@securiteam.com

Date: 3 Dec 2003 18:49:44 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

Userland Can Access Linux Kernel Memory (do_brk() Argument Bound Checking)

SUMMARY

Critical security bug has been discovered in the Linux kernel within do_brk() function that may lead to full compromise of vulnerable system. A successful exploit of this bug will allow a local user to have complete control over kernel memory space and gain uid 0 privileges. It has been discovered that the attackers of the recent compromise in Debian project's servers, exploited this vulnerability to gain root privileges.

DETAILS

Vulnerable Systems:

* Linux kernel versions 2.4.22 and previous.

Immune Systems:

* Linux kernel versions 2.4.23

The bug was discovered in September, and a fix was issued, but it 2.4.22 kernel package did not include the patch.

The physical memory of a x86 machine running one of the recent Linux

kernels is managed in a simplified flat memory model. Each user process may address a memory ranging from 0 up to TASK_SIZE bytes. Memory above this limit is not accessible to the user and contains kernel code with its data structures. User process is divided into logical sections, called virtual memory areas. The kernel keeps tracks and manages user process's virtual memory areas to provide proper memory management and memory protection faults handling. More details of Linux memory management are out of the scope of this article and can be found in [3].

The do_brk() is an internal kernel function that is called indirectly to manage process's memory heap (brk), growing or shrinking it accordingly. It is simplified version of mmap(2) system call that only handles anonymous mappings (i.e. not initialized data). The function lacks of bound checks of its parameters and may be exploited to create arbitrary large virtual memory area, exceeding user accessible memory limit. Thus, the kernel memory above this limit may become part of user process's memory as visible to the kernel memory manager.

Typical memory layout of user process may look like:

```
bash$ cat /proc/self/maps
08048000-0804c000 r-xp 00000000 03:02 207935 /bin/cat
0804c000-0804d000 rw-p 00003000 03:02 207935 /bin/cat
0804d000-0804e000 rwxp 00000000 00:00 0
40000000-40015000 r-xp 00000000 03:02 207495 /lib/ld-2.3.2.so
40015000-40016000 rw-p 00014000 03:02 207495 /lib/ld-2.3.2.so
40016000-40017000 rw-p 00000000 00:00 0
40020000-40021000 rw-p 00000000 00:00 0
42000000-4212f000 r-xp 00000000 03:02 319985 /lib/tls/libc-2.3.2.so
4212f000-42132000 rw-p 0012f000 03:02 319985 /lib/tls/libc-2.3.2.so
42132000-42134000 rw-p 00000000 00:00 0
bffffe000-c0000000 rwxp fffff000 00:00 0
```

The do_brk() function is called from within ELF and a.out loaders as well as from brk(2) syscall. These are three different vectors which may be used to exploit do_brk() bug. After successful exploitation process memory may contain a large memory mapping, i.e.:

```
080a5000-c891d000 rwxp 00000000 00:00 0
```

Impact:

Successful exploitation of do_brk() leads to full compromise of vulnerable system, including gaining full uid 0 privileges, possibility of kernel code and data structures modification as well as kernel-level (ring0) code execution.

Tested and successfully exploited kernel versions include:

- * 2.4.20-18.9 as shipped with RedHat 9.0
- * 2.4.22 (vanila)
- * 2.4.22 with grsecurity patch

There is no known reliable workaround for this vulnerability except. We recommend upgrading to the most recent kernel version (so far the 2.4.23 kernel) on all vulnerable systems.

Securiteam: [UNIX] Userland Can Access Linux Kernel Memory (do_brk() Argument Bound Checking)

Limiting maximum size of user process's data segment with ulimit -d command provides some workaround for exploit based on brk system call. However, there are at least two other attack vectors that can not be disabled without patching the system.

CVE Information:

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0961>>
CAN-2003-0961

References:

- [1] Intel Architecture Software Developer's Manual Volume 2 "Instruction Set Reference"
- [2] Intel Architecture Software Developer's Manual Volume 3 "System Programming Guide"
- [3] Daniel P. Bovet, Marco Cesati, "Understanding the Linux Kernel"

ADDITIONAL INFORMATION

The information has been provided by <mailto:ihaquer at isec.pl> Paul Starzetz and <mailto:cliph at isec.pl> Wojciech Purczynski.

The original article can be found at:

<http://isec.pl/vulnerabilities/isec-0012-do_brk.txt>
http://isec.pl/vulnerabilities/isec-0012-do_brk.txt.

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@securiteam.com
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====
=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind. In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.