

can be sometimes locally DoSed when running on particular types of hardware with certain versions of BIOS in specific m

# **[VulnWatch] Certain operating systems can be sometimes locally DoSed when running on particular types of hardware with certain versions of BIOS in specific multiboot configurations (and you thought XSS is too much?)**

*Source:* <http://www.derkeiler.com/Mailing-Lists/VulnWatch/2003-07/0050.html>

---

*From:* Michal Zalewski ([lcamtuf\\_at\\_ghettot.org](mailto:lcamtuf_at_ghettot.org))

*Date:* 07/24/03

Date: Thu, 24 Jul 2003 22:23:30 +0200 (CEST)

To: [full-disclosure@netsys.com](mailto:full-disclosure@netsys.com), [bugtraq@securityfocus.com](mailto:bugtraq@securityfocus.com), [vulnwatch@vulnwatch.org](mailto:vulnwatch@vulnwatch.org)

Yes, of course the subject line is silly... but in fact, the vulnerable combination actually occurs quite often. Still, I'm posting it here not because it's a very serious flaw, but because I find it amusing and unique. It's a CPU/BIOS/OS vulnerability, of sorts, and nobody's at fault, of course.

Thanks to Bulba for wasting his time on helping me figure out what's going on, and for a number of people for risking their lives testing this problem on their systems.

To the point. If your machine:

- is equipped with Pentium II or better,
- has a certain type of BIOS - tested and confirmed vulnerable (the list is definitely open and incomplete):

IBM ThinkPad X IZET9AWW 2.22 (09/2002)  
Dell Latitude CPx H\* revision A09  
Dell Latitude CPi A\* revision A15  
Compaq 686T2 v08.22.1999

Tested but not vulnerable:

Dell Latitude C800 revision A17  
Dell OptiPlex GX150 revision A10  
Dell Latitude C640 revision A08

...and either...

[VulnWatch] Certain operating systems can be sometimes locally DoSed when running on particular types of

can be sometimes locally DoSed when running on particular types of hardware with certain versions of BIOS in specific m

- dual boots between a fairly recent system that supports fast syscalls via SYSENTER (say, Windows XP) and a system that does not (say, Linux 2.4),

...or...

- had run a newer SYSENTER-enabled unstable/patched kernel, later downgraded to a stable version...

...then your system can be DoSed in a fairly ugly way by any of your users.

Pentium II introduced SYSENTER/SYSEXIT, a new, fast system call interface that is considerably more effective than the traditional entry method via INT or LCALL.

When you boot to a system that supports this mechanism, the system will configure certain MSRs (model-specific registers) of the CPU – primarily 0x174 (CS) and 0x176 (EIP) – to point to a specific handler code.

Once 0x174 is set, an invocation of SYSENTER opcode will cause the CPU to attempt to switch to the segment and address described in those registers. When 0x174 is zeroed, SYSENTER will simply fail, raising GPF.

Quite unfortunately, certain BIOSes do not zero those MSRs on reboot. It is not clear to me why the CPU does not reset those registers itself, even after a triple fault, but it does not. There seems to be no reasonable explanation for persistence of this setting, yet this behavior has been confirmed with several chips – Pentium II, Pentium III Katmai and Coppermine and others.

As a result, when a SYSENTER-enabled system is shut down and the machine is rebooted – but not powered down – the old setup is preserved. If a system that does not have a working SYSENTER support – as it is the case with all stable releases of Linux – is then booted up, the new system will continue to run with the "inherited" MSR settings. At this point, any user can issue a SYSENTER opcode to crash the system.

Note that those MSRs remain persistent on those boxes over subsequent warm boots, so the attack can be successful even after a very long period of time since the other system was last booted up.

Well, that's the story.

If you're concerned, you don't have to rewire your CPU or update your BIOS – the fix is to compile the following code and invoke it from your rc scripts after '/sbin/insmod msr':

```
-- sysleave.c --
```

[VulnWatch] Certain operating systems can be sometimes locally DoSed when running on particular types of

can be sometimes locally DoSed when running on particular types of hardware with certain versions of BIOS in specific m

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>

int main(void) {
    unsigned long long w = 0;
    int i = open("/dev/cpu/0/msr", O_WRONLY);
    if (i < 0) { printf("Cannot open MSR device (no module?).\n"); exit(1); }
    lseek(i, 0x174, SEEK_SET);
    if (write(i, &w, 8) < 0) { printf("MSR write error.\n"); exit(2); }
    printf("SYSENTER disabled.\n");
    return close(i);
}

-- EOF --
```

If you want to test your system, you can follow the guidelines posted at <http://lcamtuf.coredump.cx/bioses.txt> .

Cheers!

```
--
----- bash$ :(){ :|:&};: --
Michal Zalewski * [http://lcamtuf.coredump.cx]
    Did you know that clones never use mirrors?
----- 2003-07-24 16:48 --
```