

[NEWS] CPU/BIOS/OS Issue Allows Local Attacker to Cause a DoS Attack

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-07/0119.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 07/28/03

To: list@securiteam.com

Date: 28 Jul 2003 16:08:27 +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

Get Thawte's New Step-by-Step SSL Guide for Apache.

<http://ad.doubleclick.net/clk;5903117;8265118;i>

CPU/BIOS/OS Issue Allows Local Attacker to Cause a DoS Attack

SUMMARY

Some faulty BIOSs have been found to allow attacker to crash your computer (locally) in specific situations. This is due to improper implementation of newly (from Pentium II and above) introduced system calls (SYSENTER/SYSEXIT).

DETAILS

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

Securiteam: [NEWS] CPU/BIOS/OS Issue Allows Local Attacker to Cause a DoS Attack

* Dell Latitude C640 revision A08

..And either...

– 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 BIOSs do not zero those MSRs on reboot. It is not clear 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 since the other system was last booted up.

Well, that is the story.

If you are concerned, you do not 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':

Securiteam: [NEWS] CPU/BIOS/OS Issue Allows Local Attacker to Cause a DoS Attack

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

Recreation:

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

ADDITIONAL INFORMATION

The information has been provided by <<mailto:lcamtuf@ghettot.org>> Michal Zalewski.

=====

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.