Subject: SecurityFocus Article
Posted by Ed White on Thu, 11 May 2006 14:51:41 GMT
View Forum Message <> Reply to Message

A researcher of the french NSA discovered a scary vulnerability in modern x86 cpus and chipsets that expose the kernel to direct tampering.

The problem is that a feature called System Management Mode could be used to bypass the kernel and execute code at the highest level possible: ring zero.

The big problem is that the attack is possible thanks to the way X Windows is designed, and so the only way to eradicate it is to redesign it, moving video card driver into the kernel, but it seems that this cannot be done also for missing drivers and documentation!

I would like to know if OpenVZ barriers could be bypassed using this attack, or not. Maybe we will need a patch for the kernel, or for OpenVZ itself, or what?

Any hint is appreciated.

---------- ----------

The quest for ring 0

by Federico Biancuzzi
2006-05-10

Federico Biancuzzi interviews French researcher Loïc Duflot to learn about the System Management Mode attack, how to mitigate it, what hardware is vulnerable, and why we should be concerned with recent X Server bugs.
http://www.securityfocus.com/columnists/402

Subject: Re: SecurityFocus Article
Posted by Kirill Korotaev on Fri, 12 May 2006 11:06:18 GMT
View Forum Message <> Reply to Message

Ed White,
> A researcher of the french NSA discovered a scary vulnerability in modern x86 cpus and chipsets that expose the kernel to direct tampering.
>
> The problem is that a feature called System Management Mode could be used to bypass the kernel and execute code at the highest level possible: ring zero.
>
> The big problem is that the attack is possible thanks to the way X Windows is designed, and so the only way to eradicate it is to redesign it, moving video card driver into the kernel, but it seems that this cannot be done also for missing drivers and documentation!
>

> I would like to know if OpenVZ barriers could be bypassed using this attack, or not. Maybe we will need a patch for the kernel, or for OpenVZ itself, or what?
>
> Any hint is appreciated.
> ---------- ----------
>
> The quest for ring 0
>
> by Federico Biancuzzi
> 2006-05-10
>
> Federico Biancuzzi interviews French researcher Loïc Duflot to learn about the System Management Mode attack, how to mitigate it, what hardware is vulnerable, and why we should be concerned with recent X Server bugs.
> http://www.securityfocus.com/columnists/402
>
> _____

I read this article and my opinion is the following:
OpenVZ is not vulnerable.

---------- quote ----------
To carry out the general privilege escalation scheme, the attacker needs write access to various Programmed I/O registers and write access to the legacy video RAM range (0xA0000-0xbffff)
---------- quote ----------
none of these is available in OpenVZ VPS by default, so it should be impossible to exploit it from the VPS. OpenVZ doesn't give an access to any hardware by default at all.

Also, as the author states it is possible to lock SMRAM with D_LCK bit in SMRAM control register:
---------- quote ----------
It should also be noted that there is in the SMRAM control register a bit called D_LCK. If this bit is set, the SMRAM control becomes read only, and only a hard reset can clear the D_LCK bit. If this bit was set after the D_OPEN bit has been cleared, then it would be impossible to modify the default trusted SMI handler while in protected mode. The trouble is that on all the desktops I tested the D_LCK bit was cleared.
---------- quote ----------
So if one really thinks it is a problem, then it is possible to lock SMRAM and that's all. Not sure it will be fixed in mainstream somehow actually.

Kirill

Subject: Re: SecurityFocus Article
Posted by John Kelly on Wed, 17 May 2006 08:20:19 GMT
View Forum Message <> Reply to Message

On Thu May 11 2006, Ed White wrote:

> A researcher of the french NSA discovered a scary vulnerability in modern
> x86 cpus and chipsets that expose the kernel to direct tampering. The problem
> is that a feature called System Management Mode could be used to bypass the
> kernel and execute code at the highest level possible: ring zero.

Ring 0 kernel bad. KLOS good:

http://crpit.com/confpapers/CRPITV38Vasudevan.pdf