

[NT] Panda Internet Security/Antivirus+Firewall 2008 cpoint.sys Kernel Driver Memory Corruption

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2008-03/msg00016.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxxx>
 - *Date:* 10 Mar 2008 17:09:09 +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>

Panda Internet Security/Antivirus+Firewall 2008 cpoint.sys Kernel Driver
Memory Corruption

SUMMARY

The kernel driver cpoint.sys shipped with Panda Internet Security and Antivirus+Firewall 2008 contains a vulnerability in the code that handles IOCTL requests. Exploitation of this vulnerability can result in:

- 1) Local denial of service attacks (system crash due to a kernel panic),
or
- 2) Local execution of arbitrary code at the kernel level (complete system compromise)

The issue can be triggered by sending a specially crafted IOCTL request.

No special user rights are necessary to exploit the vulnerability.

DETAILS

The IOCTL call 0xba002848 of the cpoint.sys kernel driver shipped with Panda Internet Security/Antivirus+Firewall 2008 accepts user supplied input that doesn't get validated enough. In consequence it is possible to cause an out-of-bound write in kernel memory.

Disassembly of cpoint.sys (Windows Vista 32bit version):

```
[...]  
text:00012633 loc_12633:  
text:00012633 mov edx, 0BA002848h <-- (1)  
text:00012638 cmp ecx, edx  
text:0001263A ja loc_12946  
[...]  
text:00012640 jz loc_128BE  
[...]  
text:000128BE loc_128BE:  
text:000128BE cmp [ebp+IOCTL_INPUT_SIZE], 1008h <--  
(2)  
text:000128C5 jb loc_12A7D  
[...]  
text:000128CB mov esi, [ebp+IOCTL_INPUT_DATA] <-- (3)  
text:000128CE cmp dword ptr [esi], 3F256B9Ah <-- (4)  
text:000128D4 jnz loc_12A7D  
[...]  
text:000128FF xor eax, eax  
text:00012901 cmp [esi+8], eax <-- (5)  
text:00012904 jbe short loc_1291B  
[...]
```

- (1) Vulnerable IOCTL call
- (2) IOCTL input size check
- (3) The user supplied data is copied into esi
- (4) + (5) Minor input data checks

From this point there are two different vulnerable code paths. Both will

be described in the following:

Vulnerable code path 1:

```
[...]  
text:00012906 lea ecx, [esi+0Ch] <-- (6)  
[...]  
text:00012909 loc_12909:  
text:00012909 mov edx, [ecx] <-- (7)  
text:0001290B mov OVERWRITTEN_DATA[eax*4], edx <--  
(8)  
text:00012912 inc eax  
text:00012913 add ecx, 4  
text:00012916 cmp eax, [esi+8] <-- (9)  
text:00012919 jb short loc_12909  
[...]
```

- (6) Some user controlled data is copied into ecx

[NT] Panda Internet Security/Antivirus+Firewall 2008 cpoint.sys Kernel Driver Memory Corruption

- (7) The user controlled data is copied into edx
- (8) The user controlled data is copied (as dwords) at the memory location OVERWRITTEN_DATA
- (9) The size of the copied data (loop counter in eax) can be controlled by the user

This leads to an out-of-bound write in kernel memory.

Vulnerable code path 2:

```
[...]  
text:0001291B loc_1291B:  
text:0001291B xor eax, eax  
text:0001291D cmp [esi+10Ch], eax <-- (10)  
text:00012923 jbe loc_129B4  
[...]  
text:00012929 lea ecx, [esi+110h] <-- (11)  
[...]  
text:0001292F loc_1292F:  
text:0001292F mov edx, [ecx] <-- (12)  
text:00012931 mov OVERWRITTEN_DATA2[eax*4], edx <--  
(13)  
text:00012938 inc eax  
text:00012939 add ecx, 4  
text:0001293C cmp eax, [esi+10Ch] <-- (14)  
text:00012942 jb short loc_1292F  
[...]
```

- (10) Minor check of the user controlled data
- (11) Some user controlled data is copied into ecx
- (12) The user controlled data is copied into edx
- (13) The user controlled data is copied (as dwords) at the memory location OVERWRITTEN_DATA2
- (14) The size of the copied data (loop counter in eax) can be controlled by the user

This leads to an out-of-bound write in kernel memory.

In both cases it is possible to write an arbitrary amount of user controlled data into kernel memory. As the data that gets overwritten is in the data section of the cpoint.sys kernel driver it is possible to control adjacent data structures (e.g. some KEVENT structures). If these structures are overwritten with carefully crafted data it is possible to force the windows kernel into performing a memory corruption that leads to full control of the kernel execution flow.

Solution:

Hotfix for Panda Internet Security 2008:

<http://www.pandasecurity.com/homeusers/support/card?id=41337&idIdioma=2&ref=ProdExp>
<http://www.pandasecurity.com/homeusers/support/card?id=41337&idIdioma=2&ref=ProdExp>

Hotfix for Panda Antivirus+Firewall 2008:

<<http://www.pandasecurity.com/homeusers/support/card?id=41231&idIdioma=2&ref=ProdExp>>
<http://www.pandasecurity.com/homeusers/support/card?id=41231&idIdioma=2&ref=ProdExp>

Disclosure timeline:

- 2008/01/08 – Vendor notified using secure@xxxxxxxxxxxxxxxxxxxx
- 2008/01/13 – Vendor response with PGP key
- 2008/01/14 – Detailed vulnerability information sent to the vendor
- 2008/01/14 – Vendor acknowledges receipt of the information
- 2008/01/16 – Vendor confirms the vulnerability
- 2008/02/12 – Status update request sent to vendor
- 2008/02/15 – Vendor response stating that a hotfix was developed
- 2008/03/03 – Vendor sends time schedule for releasing the hotfix
- 2008/03/07 – Vendor releases hotfix
- 2008/03/08 – Full technical details released to general public

ADDITIONAL INFORMATION

The information has been provided by <<mailto:tk@xxxxxxxxxx>> Tobias Klein.

The original article can be found at:

<<http://www.trapkit.de/advisories/TKADV2008-001.txt>>

<http://www.trapkit.de/advisories/TKADV2008-001.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@xxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxx

=====

=====

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.