

## Re: FreeBSD Security Advisory FreeBSD-SA-05:01.telnet

**Source:** <http://www.derkeiler.com/Mailing-Lists/FreeBSD-Security/2005-04/0004.html>

---

**From:** Roberto ([roberto.trovo\\_at\\_redix.it](mailto:roberto.trovo_at_redix.it))

**Date:** 04/01/05

Date: Fri, 1 Apr 2005 09:29:48 +0200 (CEST)

To: "Colin Percival" <[cperciva@freebsd.org](mailto:cperciva@freebsd.org)>

> *Steve Kiernan wrote:*

>> *I was looking at this patch, but there seems to be an error in it:*

>>

>> *unsigned char slc\_reply[128];*

>> *+unsigned char const \* const slc\_reply\_eom =*

>> *&slc\_reply[sizeof(slc\_reply)];*

>> *unsigned char \*slc\_reply;*

>>

>> *Should the value for slc\_reply\_eom not be this instead?*

>>

>> *unsigned char const \* const slc\_reply\_eom = &slc\_reply[sizeof(slc\_reply)*

>> *- 1];*

>

> *No.*

>

>> *Considering the conditionals are the following:*

>>

>> *+ if (&slc\_reply[6+2] > slc\_reply\_eom)*

>> *+ return;*

>>

>> *.. and ..*

>>

>> *+ /\* The end of negotiation command requires 2 bytes. \*/*

>> *+ if (&slc\_reply[2] > slc\_reply\_eom)*

>> *+ return;*

>>

>> *If you don't subtract 1 from the sizeof(slc\_reply) or change the*

>> *conditional operators to >=, then you could try to write one byte past*

>> *the end of the buffer.*

>

> *The tests are written a bit oddly, but I'm fairly certain that they*

> *are correct. &slc\_reply[6+2] and &slc\_reply[2] are not the*

> *addresses of the last bytes which will be written; rather, they are*

> *the addresses of the byte after the last byte which will be written.*

>

FreeBSD–Security: Re: FreeBSD Security Advisory FreeBSD–SA–05:01.telnet

> *Taking the second example, if `slc_replay == slc_reply + 126`, then we*  
> *will have `&slc_replay[2] == slc_reply_eom`, but (looking at the code)*  
> *the two final bytes will be written into `slc_reply[126]` and*  
> *`slc_reply[127]`.*  
>  
> *Colin Percival*

Actually I've not read the code, but from these email it seems to me that someone could be confused by this code (at least Steve and I); for example refer to the address "`&slc_reply[128];`" when `slc_reply[127]` is the last element.

I do not want to be offensive in any way, what I want to say is that this code is clear to you (and the person who wrote it) but the next programmer that will reuse the code (because this is a open source) could make a mistake.

I think many bugs can derive from code not easy to understand.

This is only my opinion.

Kind Regards,  
Roberto

---

freebsd–security@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd–security>

To unsubscribe, send any mail to "freebsd–security–unsubscribe@freebsd.org"