
Subject: [PATCH 1/2] src/vzlist.c: Fix "cast discards qualifiers from pointer target type" warnings

Posted by [ldv](#) on Mon, 19 Feb 2007 23:35:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

src/vzlist.c (*_sort_fn, *_search_fn): Add const qualifier to casts from const pointer type

Signed-off-by: Dmitry V. Levin <ldv@altlinux.org>

src/vzlist.c | 20 ++++++++-----

1 files changed, 10 insertions(+), 10 deletions(-)

diff --git a/src/vzlist.c b/src/vzlist.c

index 15fc5d4..ed076b0 100644

--- a/src/vzlist.c

+++ b/src/vzlist.c

@@ -184,8 +184,8 @@ int none_sort_fn(const void *val1, const void *val2)

int leverage_sort_fn(const void *val1, const void *val2)

```
{
- struct Cla *la1 = ((struct Cveinfo *)val1)->la;
- struct Cla *la2 = ((struct Cveinfo *)val2)->la;
+ const struct Cla *la1 = ((const struct Cveinfo *)val1)->la;
+ const struct Cla *la2 = ((const struct Cveinfo *)val2)->la;
  int res;
```

```
  if ((res = check_empty_param(la1, la2)) == 2)
```

@@ -202,14 +202,14 @@ int leverage_sort_fn(const void *val1, const void *val2)

int id_sort_fn(const void *val1, const void *val2)

```
{
  int ret;
- ret = (((struct Cveinfo*)val1)->veid > ((struct Cveinfo*)val2)->veid);
+ ret = (((const struct Cveinfo*)val1)->veid > ((const struct Cveinfo*)val2)->veid);
  return ret;
}
```

int status_sort_fn(const void *val1, const void *val2)

```
{
  int res;
- res = ((struct Cveinfo*)val1)->status - ((struct Cveinfo*)val2)->status; if (!res)
+ res = ((const struct Cveinfo*)val1)->status - ((const struct Cveinfo*)val2)->status; if (!res)
  res = id_sort_fn(val1, val2);
  return res;
}
```

@@ -217,8 +217,8 @@ int status_sort_fn(const void *val1, const void *val2)

```
#define SORT_STR_FN(fn, name) \
int fn(const void *val1, const void *val2) \
{ \
```

```

- const char *h1 = ((struct Cveinfo*)val1)->name; \
- const char *h2 = ((struct Cveinfo*)val2)->name; \
+ const char *h1 = ((const struct Cveinfo*)val1)->name; \
+ const char *h2 = ((const struct Cveinfo*)val2)->name; \
    int ret; \
    if ((ret = check_empty_param(h1, h2)) == 2) \
        ret = strcmp(h1, h2); \
@@ -232,8 +232,8 @@ SORT_STR_FN(ip_sort_fn, ip)
#define SORT_UL_RES(fn, type, res, name, index) \
int fn(const void *val1, const void *val2) \
{ \
- struct type *r1 = ((struct Cveinfo *)val1)->res; \
- struct type *r2 = ((struct Cveinfo *)val2)->res; \
+ const struct type *r1 = ((const struct Cveinfo *)val1)->res; \
+ const struct type *r2 = ((const struct Cveinfo *)val2)->res; \
    int ret; \
    if ((ret = check_empty_param(r1, r2)) == 2) \
        ret = r1->name[index] > r2->name[index]; \
@@ -612,12 +612,12 @@ void usage()

int id_search_fn(const void* val1, const void* val2)
{
- return (*(int *)val1 - ((struct Cveinfo*)val2)->veid);
+ return (*(const int *)val1 - ((const struct Cveinfo*)val2)->veid);
}

int veid_search_fn(const void* val1, const void* val2)
{
- return (*(int *)val1 - *(int *)val2);
+ return (*(const int *)val1 - *(const int *)val2);
}

void print_hdr()
--
1.5.0.1.GIT

```

Subject: Re: [PATCH 1/2] src/vzlist.c: Fix "cast discards qualifiers from pointer target type" war

Posted by [kir](#) on Tue, 20 Feb 2007 05:52:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

Thanks, committed to git:

<http://git.openvz.org/?p=vzctl;a=commitdiff;h=fefbe762d120c1328f928f340d1c4a5661ceea5f>

That's actually the same as the last part of my patch to bug #434
(<http://bugzilla.openvz.org/434>)

Dmitry V. Levin wrote:

```
> src/vzlist.c (*_sort_fn, *_search_fn): Add const qualifier to casts from const pointer type
>
> Signed-off-by: Dmitry V. Levin <ldv@altlinux.org>
> ---
> src/vzlist.c | 20 ++++++++-----
> 1 files changed, 10 insertions(+), 10 deletions(-)
>
> diff --git a/src/vzlist.c b/src/vzlist.c
> index 15fc5d4..ed076b0 100644
> --- a/src/vzlist.c
> +++ b/src/vzlist.c
> @@ -184,8 +184,8 @@ int none_sort_fn(const void *val1, const void *val2)
>
> int leverage_sort_fn(const void *val1, const void *val2)
> {
> - struct Cla *la1 = ((struct Cveinfo *)val1)->la;
> - struct Cla *la2 = ((struct Cveinfo *)val2)->la;
> + const struct Cla *la1 = ((const struct Cveinfo *)val1)->la;
> + const struct Cla *la2 = ((const struct Cveinfo *)val2)->la;
> int res;
>
> if ((res = check_empty_param(la1, la2)) == 2)
> @@ -202,14 +202,14 @@ int leverage_sort_fn(const void *val1, const void *val2)
> int id_sort_fn(const void *val1, const void *val2)
> {
> int ret;
> - ret = (((struct Cveinfo*)val1)->veid > ((struct Cveinfo*)val2)->veid);
> + ret = (((const struct Cveinfo*)val1)->veid > ((const struct Cveinfo*)val2)->veid);
> return ret;
> }
>
> int status_sort_fn(const void *val1, const void *val2)
> {
> int res;
> - res = ((struct Cveinfo*)val1)->status - ((struct Cveinfo*)val2)->status; if (!res)
> + res = ((const struct Cveinfo*)val1)->status - ((const struct Cveinfo*)val2)->status; if (!res)
> res = id_sort_fn(val1, val2);
> return res;
> }
> @@ -217,8 +217,8 @@ int status_sort_fn(const void *val1, const void *val2)
> #define SORT_STR_FN(fn, name) \
> int fn(const void *val1, const void *val2) \
> { \
> - const char *h1 = ((struct Cveinfo*)val1)->name; \
> - const char *h2 = ((struct Cveinfo*)val2)->name; \
> + const char *h1 = ((const struct Cveinfo*)val1)->name; \
> + const char *h2 = ((const struct Cveinfo*)val2)->name; \
```

```

> int ret; \
> if ((ret = check_empty_param(h1, h2)) == 2) \
> ret = strcmp(h1, h2); \
> @@ -232,8 +232,8 @@ SORT_STR_FN(ip_sort_fn, ip)
> #define SORT_UL_RES(fn, type, res, name, index) \
> int fn(const void *val1, const void *val2) \
> { \
> - struct type *r1 = ((struct Cveinfo *)val1)->res; \
> - struct type *r2 = ((struct Cveinfo *)val2)->res; \
> + const struct type *r1 = ((const struct Cveinfo *)val1)->res; \
> + const struct type *r2 = ((const struct Cveinfo *)val2)->res; \
> int ret; \
> if ((ret = check_empty_param(r1, r2)) == 2) \
> ret = r1->name[index] > r2->name[index]; \
> @@ -612,12 +612,12 @@ void usage()
>
> int id_search_fn(const void* val1, const void* val2)
> {
> - return (*(int *)val1 - ((struct Cveinfo*)val2)->veid);
> + return (*(const int *)val1 - ((const struct Cveinfo*)val2)->veid);
> }
>
> int veid_search_fn(const void* val1, const void* val2)
> {
> - return (*(int *)val1 - *(int *)val2);
> + return (*(const int *)val1 - *(const int *)val2);
> }
>
> void print_hdr()
>
> -----
>

```
