Subject: mini howto: compiling nscd on VE
Posted by muzzol on Mon, 13 Jun 2011 10:24:00 GMT
View Forum Message <> Reply to Message

hi,

i've solved the problem about nscd on VE.

there's some threads talking about it:

forum.openvz.org/index.php?t=msg&goto=26259&

but those workarounds were not suitable for me so i went to recompilation way.

this was performed on a CentOS 5.6, adjust at your will.

basically what i've done is removing all references to LIBCAP on the code. i'ts a very ugly solution and you should know the implications of disabling capabilities on any binary.

that said, this is how i did it:

download your glibc source package. in my cas it was: glibc-2.5-58.src.rpm

install it and enter the apropiate dir: rpm -Uhv glibc-2.5-58.src.rpm
cd /usr/src/redhat

modify specs file:
mcedit SPECS/glibc.spec

add next line after %patchXXX -p1 stanzas, around line 740: echo "modify nscd and hit enter" ;read ZZZZZZZZ

it should look like this:
[...]
%patch245 -p1
%patch246 -p1
%patch247 -p1

echo "modify nscd and hit enter" ;read ZZZZZZZZ

Hack till glibc-kernheaders get updated, argh
mkdir -p override_headers/linux
[...]

now execute build command: rpmbuild -ba --target i686 SPECS/glibc.spec

at some point it will pause and you will see the message you just entered. don't hit enter yet: +
patch -p1 -s
+ echo modify nscd and hit enter
modify nscd and hit enter
+ read ZZZZZZZZZ

you must go to another terminal and modify two files: connections.c and selinux.c. both are on nscd subdir that is created on BUILD/glibc subdir. glibc subdir can vary, in my cas it was:
mcedit BUILD/glibc-2.5-20061008T1257/nscd/connections.c

mcedit BUILD/glibc-2.5-20061008T1257/nscd/selinux.c

on connections.c you have to remove these blocks of code. around line 2360:

```
#if defined HAVE_LIBAUDIT && defined HAVE_LIBCAP
  /* We need to preserve the capabilities to connect to the audit daemon.  */
  cap_t new_caps = preserve_capabilities ();
#endif
```

and around line 2395:

```
#if defined HAVE_LIBAUDIT && defined HAVE_LIBCAP
  /* Remove the temporary capabilities.  */
  install_real_capabilities (new_caps);
#endif
```

now on selinux.c you have to remove a 85 piece of code. be carefull to remove only the ifdef HAVE_LIBCAP definition:

```
# ifdef HAVE_LIBCAP

static const cap_value_t new_cap_list[] =

  { CAP_AUDIT_WRITE };

#  define nnew_cap_list (sizeof (new_cap_list) / sizeof (new_cap_list[0]))

static const cap_value_t tmp_cap_list[] =

  { CAP_AUDIT_WRITE, CAP_SETUID, CAP_SETGID };

#  define ntmp_cap_list (sizeof (tmp_cap_list) / sizeof (tmp_cap_list[0]))
```

```
cap_t
preserve_capabilities (void)
{
  if (getuid () != 0)
    /* Not root, then we cannot preserve anything.  */
    return NULL;

  if (prctl (PR_SET_KEEPCAPS, 1) == -1)
    {
      dbg_log (_("Failed to set keep-capabilities"));
      error (EXIT_FAILURE, errno, _("prctl(KEEPCAPS) failed"));
      /* NOTREACHED */
    }

  cap_t tmp_caps = cap_init ();
  cap_t new_caps = NULL;
  if (tmp_caps != NULL)
    new_caps = cap_init ();

  if (tmp_caps == NULL || new_caps == NULL)
    {
      if (tmp_caps != NULL)
        cap_free (tmp_caps);
```

```
    dbg_log (_("Failed to initialize drop of capabilities"));
    error (EXIT_FAILURE, 0, _("cap_init failed"));
  }

/* There is no reason why these should not work.  */
cap_set_flag (new_caps, CAP_PERMITTED, nnew_cap_list,
              (cap_value_t *) new_cap_list, CAP_SET);
cap_set_flag (new_caps, CAP_EFFECTIVE, nnew_cap_list,
              (cap_value_t *) new_cap_list, CAP_SET);

cap_set_flag (tmp_caps, CAP_PERMITTED, ntmp_cap_list,
              (cap_value_t *) tmp_cap_list, CAP_SET);
cap_set_flag (tmp_caps, CAP_EFFECTIVE, ntmp_cap_list,
              (cap_value_t *) tmp_cap_list, CAP_SET);

int res = cap_set_proc (tmp_caps);

cap_free (tmp_caps);

if (__builtin_expect (res != 0, 0))
  {
    cap_free (new_caps);
    dbg_log (_("Failed to drop capabilities"));
    error (EXIT_FAILURE, 0, _("cap_set_proc failed"));
```

```
    }


  return new_caps;
}

void
install_real_capabilities (cap_t new_caps)
{
  /* If we have no capabilities there is nothing to do here.  */
  if (new_caps == NULL)
    return;

  if (cap_set_proc (new_caps))
    {
      cap_free (new_caps);
      dbg_log (_("Failed to drop capabilities"));
      error (EXIT_FAILURE, 0, _("cap_set_proc failed"));
      /* NOTREACHED */
    }

  cap_free (new_caps);

  if (prctl (PR_SET_KEEPCAPS, 0) == -1)
    {
```

```
      dbg_log (_("Failed to unset keep-capabilities"));

      error (EXIT_FAILURE, errno, _("prctl(KEEPCAPS) failed"));

      /* NOTREACHED */

    }

}

# endif /* HAVE_LIBCAP */
```

now go back to previous terminal and hit enter. after some time you'll see on BUILD/glibc-2.5-20061008T1257/nscd/ the new binary. you don't have to wait rpmbuild to finish if you just want nscd. in fact you can modify the spec file to just compile nscd, but i recommend to touch only if necessary this file to avoid mistakes.

copy newly created nscd bin to anywhere you want. you can replace original nscd if you want just doing:
mv /usr/sbin/nscd /usr/sbin/nscd.orig
mv /tmp/nscd /usr/sbin/nscd

asuming you have new nscd on /tmp.

i hope this saves some time to people with this same problem.

Subject: Re: mini howto: compiling nscd on VE
Posted by curx on Mon, 13 Jun 2011 20:41:09 GMT
View Forum Message <> Reply to Message

please add an article to http://wiki.openvz.org

Bye,
 Thorsten

Subject: Re: mini howto: compiling nscd on VE
Posted by muzzol on Mon, 13 Jun 2011 21:50:41 GMT
View Forum Message <> Reply to Message

i've tried to create a page but i don't have permission.

must i contact any admin or wait any days to edit?