Bug 721570 - net-vpn/ocserv-1.0.1: test failures on arm64 due to stack smashing detection with LD_PRELOAD=libsocket_wrapper.so

Status: RESOLVED FIXED
Alias: None
Product: Gentoo Linux
Component: Current packages (show other bugs)
Hardware: ARM64 Linux
Importance: Normal normal (vote)
Assignee: Mike Gilbert
URL:
Whiteboard:
Keywords: TESTFAILURE
Depends on:
Blocks: ~~CVE-2020-12105~~
Show dependency tree

Reported: 2020-05-08 00:51 UTC by Sam James
Modified: 2020-05-12 17:29 UTC (History)
CC List: 3 users (show)

See Also: ~~CVE-2020-12023~~

Attachments	
test-suite.log (file_721570.txt,1.66 KB, text/plain) 2020-05-08 00:51 UTC, Sam James	Details
build.log (file_721570.txt,106.17 KB, text/plain) 2020-05-08 00:52 UTC, Sam James	Details
openconnect-8.06-get_cert_name-overflow.patch (openconnect-8.06-get_cert_name-overflow.patch,686 bytes, patch) 2020-05-08 14:02 UTC, Sergei Trofimovich (RETIRED)	Details | Diff
Add an attachment (proposed patch, testcase, etc.)	View All

Note
You need to log in before you can comment on or make changes to this bug.

Sam James AT Infra Dev Sec **2020-05-08 00:51:42 UTC** **Description**

Created attachment 636780 [details]
test-suite.log

```
FAIL: test-pass-group-cert
==========================

Testing local backend with username-password and certificate...
Connecting to obtain cookie (without certificate)... ok
Connecting to obtain cookie - group1 (with certificate)... Failure: Could not
connect with certificate!
FAIL test-pass-group-cert (exit status: 1)

FAIL: test-pass-group-cert-no-pass
==================================

Testing local backend with username-password and certificate...
Connecting to obtain cookie (without certificate)... ok
Connecting to obtain cookie - group1 (with certificate)... Failure: Could not
connect with certificate!
FAIL test-pass-group-cert-no-pass (exit status: 1)


----
Portage 2.3.99 (python 3.7.7-final-0, default/linux/arm64/17.0, gcc-9.3.0, glibc-
2.30-r8, 4.9.0-4-arm64 aarch64)
=================================================================
System uname: Linux-4.9.0-4-arm64-aarch64-with-gentoo-2.6
KiB Mem:   131544964 total, 115445756 free
KiB Swap:    3321056 total,   3321056 free
Timestamp of repository gentoo: Thu, 07 May 2020 22:35:12 +0000
sh bash 5.0_p17
ld GNU ld (Gentoo 2.33.1 p2) 2.33.1
ccache version 3.7.7 [disabled]
app-shells/bash:          5.0_p17::gentoo
dev-lang/perl:            5.30.1::gentoo
dev-lang/python:          2.7.18::gentoo, 3.6.10-r2::gentoo, 3.7.7-r2::gentoo,
3.8.2-r2::gentoo
dev-util/ccache:          3.7.7-r1::gentoo
dev-util/cmake:           3.14.6::gentoo
sys-apps/baselayout:      2.6-r1::gentoo
sys-apps/openrc:          0.42.1::gentoo
sys-apps/sandbox:         2.13::gentoo
sys-devel/autoconf:       2.13-r1::gentoo, 2.69-r4::gentoo
sys-devel/automake:       1.16.1-r1::gentoo
sys-devel/binutils:       2.33.1-r1::gentoo
sys-devel/gcc:            9.3.0::gentoo
sys-devel/gcc-config:     2.2.1::gentoo
sys-devel/libtool:        2.4.6-r6::gentoo
sys-devel/make:           4.2.1-r4::gentoo
sys-kernel/linux-headers: 5.4::gentoo (virtual/os-headers)
sys-libs/glibc:           2.30-r8::gentoo
Repositories:

gentoo
    location: /bound/portage
    sync-type: rsync
    sync-uri: rsync://rsync.gentoo.org/gentoo-portage
    priority: -1000
    sync-rsync-verify-metamanifest: yes
    sync-rsync-extra-opts:
    sync-rsync-verify-max-age: 24
    sync-rsync-verify-jobs: 1

ACCEPT_KEYWORDS="arm64"
ACCEPT_LICENSE="@FREE"
CBUILD="aarch64-unknown-linux-gnu"
CFLAGS="-O2 -pipe -march=native -fdiagnostics-show-option -frecord-gcc-switches"
CHOST="aarch64-unknown-linux-gnu"
CONFIG_PROTECT="/etc /usr/share/config /usr/share/gnupg/qualified.txt /var/bind"
CONFIG_PROTECT_MASK="/etc/ca-certificates.conf /etc/dconf /etc/env.d
/etc/fonts/fonts.conf /etc/gconf /etc/gentoo-release /etc/php/apache2-php7.4/ext-
active/ /etc/php/cgi-php7.4/ext-active/ /etc/php/cli-php7.4/ext-active/
/etc/revdep-rebuild /etc/sandbox.d /etc/terminfo /etc/texmf/language.dat.d
/etc/texmf/language.def.d /etc/texmf/updmap.d /etc/texmf/web2c"
CXXFLAGS="-O2 -pipe -march=native"
DISTDIR="/bound/distfiles"
EMERGE_DEFAULT_OPTS="--keep-going --jobs=3"
ENV_UNSET="DBUS_SESSION_BUS_ADDRESS DISPLAY GOBIN PERL5LIB PERL5OPT PERLPREFIX
PERL_CORE PERL_MB_OPT PERL_MM_OPT XAUTHORITY XDG_CACHE_HOME XDG_CONFIG_HOME
XDG_DATA_HOME XDG_RUNTIME_DIR"
FCFLAGS="-O2 -pipe -march=native"
FEATURES="assume-digests binpkg-docompress binpkg-dostrip binpkg-logs config-
protect-if-modified distlocks ebuild-locks fixlafiles ipc-sandbox merge-sync
multilib-strict network-sandbox news parallel-fetch pid-sandbox preserve-libs
protect-owned qa-unresolved-soname-deps sandbox sfperms strict unknown-features-
warn unmerge-logs unmerge-orphans userfetch userpriv usersandbox usersync xattr"
```

```
FFLAGS="-O2 -pipe -march=native"
GENTOO_MIRRORS="http://distfiles.gentoo.org"
INSTALL_MASK="/usr/share/doc/*/*.pdf /usr/share/man/*/*"
LANG="en_US.UTF-8"
LDFLAGS="-Wl,-O1 -Wl,--as-needed -Wl,--hash-style=gnu"
MAKEOPTS="-j40"
PKGDIR="/usr/portage/packages"
PORTAGE_CONFIGROOT="/"
PORTAGE_RSYNC_OPTS="--recursive --links --safe-links --perms --times --omit-dir-
times --compress --force --whole-file --delete --stats --human-readable --
timeout=180 --exclude=/distfiles --exclude=/local --exclude=/packages --
exclude=/.git"
PORTAGE_TMPDIR="/var/tmp"
USE="acl arm64 berkdb bzip2 cli crypt dri fortran gdbm iconv ipv6 libtirpc ncurses
nls nptl openmp pam pcre readline seccomp split-usr ssl tcpd unicode xattr zlib"
ADA_TARGET="gnat_2018" APACHE2_MODULES="authn_core authz_core socache_shmcb unixd
actions alias auth_basic authn_alias authn_anon authn_dbm authn_default authn_file
authz_dbm authz_default authz_groupfile authz_host authz_owner authz_user autoindex
cache cgi cgid dav dav_fs dav_lock deflate dir disk_cache env expires ext_filter
file_cache filter headers include info log_config logio mem_cache mime mime_magic
negotiation rewrite setenvif speling status unique_id userdir usertrack
vhost_alias" CALLIGRA_FEATURES="karbon sheets words" COLLECTD_PLUGINS="df interface
irq load memory rrdtool swap syslog" CPU_FLAGS_ARM="edsp thumb vfp vfpv3 vfpv4 vfp-
d32 aes sha1 sha2 crc32 v4 v5 v6 v7 v8 thumb2" ELIBC="glibc"
GPSD_PROTOCOLS="ashtech aivdm earthmate evermore fv18 garmin garmintxt gpsclock
greis isync itrax mtk3301 nmea ntrip navcom oceanserver oldstyle oncore rtcm104v2
rtcm104v3 sirf skytraq superstar2 timing tsip tripmate tnt ublox ubx"
INPUT_DEVICES="libinput" KERNEL="linux" LCD_DEVICES="bayrad cfontz cfontz633 glk
hd44780 lb216 lcdm001 mtxorb ncurses text" LIBREOFFICE_EXTENSIONS="presenter-
console presenter-minimizer" OFFICE_IMPLEMENTATION="libreoffice" PHP_TARGETS="php7-
2" POSTGRES_TARGETS="postgres10 postgres11" PYTHON_SINGLE_TARGET="python3_7"
PYTHON_TARGETS="python2_7 python3_7" RUBY_TARGETS="ruby24 ruby25" USERLAND="GNU"
VIDEO_CARDS="fbdev dummy v4l" XTABLES_ADDONS="quota2 psd pknock lscan length2
ipv4options ipset ipp2p iface geoip fuzzy condition tee tarpit sysrq steal rawnat
logmark ipmark dhcpmac delude chaos account"
Unset:  CC, CPPFLAGS, CTARGET, CXX, LC_ALL, LINGUAS, PORTAGE_BINHOST,
PORTAGE_BUNZIP2_COMMAND, PORTAGE_COMPRESS, PORTAGE_COMPRESS_FLAGS,
PORTAGE_RSYNC_EXTRA_OPTS
```

Sam James AT Infra Dev Sec **2020-05-08 00:52:33 UTC** **Comment 1**

Created attachment 636782 [details]
build.log

Mike Gilbert Dev **2020-05-08 02:09:16 UTC** **Comment 2**

I cannot reproduce this.

Please edit tests/test-pass-group-cert, and remove the redirection(s) to /dev/null. Then run the test script manually and post the output.

Sam James AT Infra Dev Sec **2020-05-08 02:27:42 UTC** **Comment 3**

(In reply to Mike Gilbert from comment #2)
> I cannot reproduce this.
>
> Please edit tests/test-pass-group-cert, and remove the redirection(s) to
> /dev/null. Then run the test script manually and post the output.

```
# ./test-pass-group-cert
Testing local backend with username-password and certificate...
Connecting to obtain cookie (without certificate)... Server '127.0.0.2' requested
Basic authentication which is disabled by default
Failed to obtain WebVPN cookie
ok
Connecting to obtain cookie - group1 (with certificate)... *** stack smashing
detected ***: <unknown> terminated
./test-pass-group-cert: line 40: 70583 Aborted
LD_PRELOAD=libsocket_wrapper.so /usr/sbin/openconnect --authgroup group1 -q
127.0.0.2:6551 --sslkey ./certs/user-group-key.pem -c ./certs/user-group-cert.pem -
u test --servercert=d66b507ae074d03b02eafca40d35f87dd81049d3 --cookieonly
Failure: Could not connect with certificate!

# ./test-pass-group-cert-no-pass
Testing local backend with username-password and certificate...
Connecting to obtain cookie (without certificate)... Server '127.0.0.2' requested
Basic authentication which is disabled by default
Failed to obtain WebVPN cookie
ok
Connecting to obtain cookie - group1 (with certificate)... *** stack smashing
detected ***: <unknown> terminated
./test-pass-group-cert-no-pass: line 41: 70603 Aborted
LD_PRELOAD=libsocket_wrapper.so /usr/sbin/openconnect --authgroup group1 -q
127.0.0.2:6555 --sslkey ./certs/user-group-key.pem -c ./certs/user-group-cert.pem -
u test --servercert=d66b507ae074d03b02eafca40d35f87dd81049d3 --cookieonly
Failure: Could not connect with certificate!

----
Note that running w/o LD_PRELOAD runs without crashing

# /usr/sbin/openconnect --authgroup group1 -q 127.0.0.2:6555 --sslkey ./certs/user-
group-key.pem -c ./certs/user-group-cert.pem -u test --
servercert=d66b507ae074d03b02eafca40d35f87dd81049d3 --cookieonly
Failed to connect to host 127.0.0.2
Failed to open HTTPS connection to 127.0.0.2
Failed to obtain WebVPN cookie
```

Mike Gilbert Dev **2020-05-08 02:43:15 UTC** **Comment 4**

This is probably not a bug in openconnect or ocserv, so I am unblocking the stable request.

Copying samba for net-libs/socket_wrapper, and toolchain for the stack smashing error. Can you provide any insight?

Sergei Trofimovich (RETIRED) Dev **2020-05-08 08:58:25 UTC** **Comment 5**

(In reply to Sam James (sec padawan) from comment #3)
> Connecting to obtain cookie - group1 (with certificate)... *** stack
> smashing detected ***: <unknown> terminated

Do you have a backtrace for that crash?

Sergei Trofimovich (RETIRED) Dev **2020-05-08 11:13:48 UTC** **Comment 6**

(In reply to Sergei Trofimovich from comment #5)
> (In reply to Sam James (sec padawan) from comment #3)
> > Connecting to obtain cookie - group1 (with certificate)... *** stack
> > smashing detected ***: <unknown> terminated
>
> Do you have a backtrace for that crash?

Got it on arm64-build.arm.dev.gentoo.org:

```
# ulimit -c unlimited
# ./test-pass-group-cert-no-pass
Testing local backend with username-password and certificate...
```

```
Connecting to obtain cookie (without certificate)... Server '127.0.0.2' requested
Basic authentication which is disabled by default
Failed to obtain WebVPN cookie
ok
Connecting to obtain cookie - group1 (with certificate)... *** stack smashing
detected ***: <unknown> terminated
./test-pass-group-cert-no-pass: line 41: 13204 Aborted                 (core
dumped) LD_PRELOAD=libsocket_wrapper.so /usr/sbin/openconnect --authgroup group1 -q
127.0.0.2:6555 --sslkey ./certs/user-group-key.pem -c ./certs/user-group-cert.pem -
u test --servercert=d66b507ae074d03b02eafca40d35f87dd81049d3 --cookieonly
Failure: Could not connect with certificate!

# file core
core: ELF 64-bit LSB core file, ARM aarch64, version 1 (SYSV), SVR4-style, from
'/usr/sbin/openconnect --authgroup group1 -q 127.0.0.2:6555 --sslkey ./certs/use',
real uid: 0, effective uid: 0, real gid: 0, effective gid: 0, execfn:
'/usr/sbin/openconnect', platform: 'aarch64'

# qfile /usr/sbin/openconnect
net-vpn/openconnect: /usr/sbin/openconnect

# gdb --quiet /usr/sbin/openconnect core
Reading symbols from /usr/sbin/openconnect...
Reading symbols from /usr/lib/debug//usr/sbin/openconnect.debug...
[New LWP 13204]
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib64/libthread_db.so.1".
Core was generated by `/usr/sbin/openconnect --authgroup group1 -q 127.0.0.2:6555 -
-sslkey ./certs/use'.
Program terminated with signal SIGABRT, Aborted.
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
50    ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) bt
#0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
#1  0x0000ffff95a19728 in __GI_abort () at abort.c:79
#2  0x0000ffff95a65b6c in __libc_message (action=<optimized out>,
fmt=fmt@entry=0xffff95b1bce8 "*** %s ***: %s terminated\n")
    at ../sysdeps/posix/libc_fatal.c:181
#3  0x0000ffff95ad526c in __GI___fortify_fail_abort
(need_backtrace=need_backtrace@entry=false, msg=msg@entry=0xffff95b1bcc0 "stack
smashing detected")
    at fortify_fail.c:33
#4  0x0000ffff95ad5220 in __stack_chk_fail () at stack_chk_fail.c:29
#5  0x0000ffff95ed7a40 in load_certificate (vpninfo=vpninfo@entry=0xaaaaaffb09140)
at gnutls.c:1356
#6  0x0000ffff95ed92e0 in openconnect_open_https
(vpninfo=vpninfo@entry=0xaaaaaffb09140) at gnutls.c:2158
#7  0x0000ffff95eb2258 in do_https_request (vpninfo=vpninfo@entry=0xaaaaaffb09140,
method=method@entry=0xffff95ede8a8 "POST",
    request_body_type=request_body_type@entry=0xffff95edfeb8 "application/xml;
charset=utf-8", request_body=request_body@entry=0xaaaaaffb02eb0,
    form_buf=form_buf@entry=0xffffe36cbf58, fetch_redirect=fetch_redirect@entry=0)
at http.c:1056
#8  0x0000ffff95ec3734 in cstp_obtain_cookie (vpninfo=0xaaaaaffb09140) at
auth.c:1312
#9  0x0000aaaadc89eda4 in main (argc=13, argv=0xffffe36cc408) at main.c:1599
```

Sergei Trofimovich (RETIRED) Dev **2020-05-08 13:49:55 UTC** Comment 7

```
valgrind did not find any problems. asan refused to work with LD_PRELOADED
libraries that manually load libc.so.6.

Resorting to tracking who stack canary on stack.

Before looking at the specifics here is how stack canary usually looks like on
arm64:

  // cat a.c
  void g(const char *);
  void f(void)
  {
     char b[128];
     g(b);
  }

$ aarch64-unknown-linux-gnu-gcc -O2 -c -S a.c; cat a.s
  ...
  f:
        stp     x29, x30, [sp, -176]!
        mov     x29, sp
        str     x19, [sp, 16]
        adrp    x19, :got:__stack_chk_guard
        add     x0, sp, 40
        ldr     x19, [x19, #:got_lo12:__stack_chk_guard]
        ldr     x1, [x19]
        str     x1, [sp, 168]
        mov     x1,0

        bl      g

        ldr     x1, [sp, 168]
        ldr     x0, [x19]
        eor     x0, x1, x0
        cbnz    x0, .L5
        ldr     x19, [sp, 16]
        ldp     x29, x30, [sp], 176
        ret
.L5:
        bl      __stack_chk_fail

It's a very verbose way to:
- load value by '__stack_chk_guard' absolute address into 'x19' register
- store canary copy on stack at '[sp +168]' address
- restore canary from stack and from address at 'x19' and compare it back.

A good hint here is that right after the store 'x1' value is zeroed out with 'mov
x1,0'.

Let's track life of a canary in 'load_certificate' and see where it gets corrupted:

I added 'gdb --args' to openconnect call in test-pass-group-cert-no-pass and
removed a few unused evals.

    LD_PRELOAD="libsocket_wrapper.so" gdb --args $OPENCONNECT --authgroup group1
...

  # bash -x ./test-pass-group-cert-no-pass
  ...
  Connecting to obtain cookie - group1 (with certificate)... +
LD_PRELOAD=libsocket_wrapper.so
  + gdb --args /usr/sbin/openconnect --authgroup group1 -q 127.0.0.2:6555 --sslkey
./certs/user-group-key.pem -c ./certs/user-group-cert.pem -u test --
servercert=d66b507ae074d03b02eafca40d35f87dd81049d3 --cookieonly

  (gdb) start
  (gdb) break load_certificate
  (gdb) continue
  Breakpoint 2, load_certificate (vpninfo=vpninfo@entry=0xaaaaaaae5140) at
gnutls.c:916
  Dump of assembler code for function load_certificate:
=> 0x0000ffffb7f88d08 <+0>:     sub     sp, sp, #0x200
   0x0000ffffb7f88d0c <+4>:     mov     x3, #0x14                       // #20
   0x0000ffffb7f88d10 <+8>:     mov     x2, #0x7                        // #7
   0x0000ffffb7f88d14 <+12>:    stp     x29, x30, [sp, #16]
   0x0000ffffb7f88d18 <+16>:    add     x29, sp, #0x10
   0x0000ffffb7f88d1c <+20>:    stp     x21, x22, [sp, #48]
   0x0000ffffb7f88d20 <+24>:    adrp    x21, 0xffffb7faf000
   0x0000ffffb7f88d24 <+28>:    stp     x19, x20, [sp, #32]
```

```
   0x0000ffffb7f88d28 <+32>:    mov     x19, x0
   0x0000ffffb7f88d2c <+36>:    adrp    x20, 0xffffb7f97000
   0x0000ffffb7f88d30 <+40>:    ldr     x0, [x21, #3992]
   0x0000ffffb7f88d34 <+44>:    stp     x23, x24, [sp, #64]
   0x0000ffffb7f88d38 <+48>:    add     x20, x20, #0x918
   0x0000ffffb7f88d3c <+52>:    ldr     x24, [x19, #1128]
   0x0000ffffb7f88d40 <+56>:    ldr     x1, [x0]
   0x0000ffffb7f88d44 <+60>:    str     x1, [sp, #504]
   0x0000ffffb7f88d48 <+64>:    mov     x1, #0x0                        // #0
   0x0000ffffb7f88d4c <+68>:    ldr     x23, [x19, #1120]
   0x0000ffffb7f88d50 <+72>:    mov     x1, x20
   0x0000ffffb7f88d54 <+76>:    mov     x0, x24
   0x0000ffffb7f88d58 <+80>:    stp     x25, x26, [sp, #80]
   0x0000ffffb7f88d5c <+84>:    mov     w26, #0x0                       // #0

Here is our canary store:

   0x0000ffffb7f88d40 <+56>:    ldr     x1, [x0]
   0x0000ffffb7f88d44 <+60>:    str     x1, [sp, #504]
   0x0000ffffb7f88d48 <+64>:    mov     x1, #0x0                        // #0

Absolute address of canary copy on stack is '[sp, #504]'. Let's find it out and
trace it's life:

  (gdb) break *0x0000ffffb7f88d48
  Breakpoint 3 at 0xffffb7f88d48: file gnutls.c, line 916.
  (gdb) continue
  Continuing.
  Breakpoint 3, 0x0000ffffb7f88d48 in load_certificate
(vpninfo=vpninfo@entry=0xaaaaaaaae5140) at gnutls.c:916

Looks like a valid canary in register, stack and global variable:

  (gdb) print (void*)$x1
  $3 = (void *) 0xb9b1883f29832000
  (gdb) print *(void**)($sp+504)
  $4 = (void *) 0xb9b1883f29832000
  (gdb) print *(void**)&__stack_chk_guard
  $6 = (void *) 0xb9b1883f2983200

Print stack address and add a watch point on it:

  (gdb) print (void*)($sp+504)
  $7 = (void *) 0xffffffffea
  (gdb) watch *(void**)0xffffffffeaa8
  Hardware watchpoint 3: *(void**)0xffffffffeaa8
  (gdb) continue
  Continuing.

  Hardware watchpoint 3: *(void**)0xffffffffeaa8

  Old value = (void *) 0xffffb7f8c2b4 <openconnect_open_https+572>
  New value = (void *) 0xc9d1f41373228800
  0x0000ffffb7f88d48 in load_certificate (vpninfo=vpninfo@entry=0xaaaaaaaae5140) at
gnutls.c:916
  916   in gnutls.c

  (gdb) bt
  #0  0x0000ffffb7f88d48 in load_certificate (vpninfo=vpninfo@entry=0xaaaaaaaae5140)
at gnutls.c:916
  #1  0x0000ffffb7f8c2e0 in openconnect_open_https
(vpninfo=vpninfo@entry=0xaaaaaaaae5140) at gnutls.c:2158

  (gdb) bt
  #0  get_cert_name (cert=0xaaaaaaaaea2f0, name=name@entry=0xffffffffea58 "",
namelen=<optimized out>) at gnutls.c:563
  #1  0x0000ffffb7f89408 in load_certificate (vpninfo=vpninfo@entry=0xaaaaaaaae5140)
at gnutls.c:1545
  ...


   557  static int get_cert_name(gnutls_x509_crt_t cert, char *name, size_t
namelen)
   558  {
   559          if (gnutls_x509_crt_get_dn_by_oid(cert,
GNUTLS_OID_X520_COMMON_NAME,
   560                                            0, 0, name, &namelen) &&
   561              gnutls_x509_crt_get_dn(cert, name, &namelen)) {
   562                  name[namelen-1] = 0;
   563                  snprintf(name, namelen-1, "<unknown>");
   564                  return -EINVAL;
   565          }
   566          return 0;
   567  }

  (gdb) disassemble
   0x0000ffffb7f88ab8 <+160>:   sub     x1, x1, #0x1
   0x0000ffffb7f88abc <+164>:   strb    wzr, [x20, x1]
=> 0x0000ffffb7f88ac0 <+168>:   bl      0xffffb7f5f590 <snprintf@plt>

I think it tells us that 'name[namelen-1] = 0;' corrupted the stack.

Probably 'name' value:
  (gdb) printf "%#x\n", (void*)$x20
  0xffffea58

Probably 'namelen-1' value:
  (gdb) printf "%#x\n", (void*)$x1
  0x54 # 84

But get_cert_name is ever called against 'name' with namelen=80.

Let's see if I can find if 'namelen' is inflated by 'gnutls_x509_crt_get_dn' when
it fails or somewhere else.
```

Sergei Trofimovich (RETIRED) Dev **2020-05-08 14:02:17 UTC** **Comment 8**

Created attachment 636836 [details, diff]
openconnect-8.06-get_cert_name-overflow.patch

openconnect-8.06-get_cert_name-overflow.patch shoud fix the overflow in net-vpn/openconnect

Sergei Trofimovich (RETIRED) Dev **2020-05-08 14:04:42 UTC** **Comment 9**

From http://man7.org/linux/man-pages/man3/gnutls_x509_crt_get_dn_by_oid.3.html :

```
"""
RETURNS
       GNUTLS_E_SHORT_MEMORY_BUFFER if the provided buffer is not long
       enough, and in that case the  buf_size will be updated with the
       required size. GNUTLS_E_REQUESTED_DATA_NOT_AVAILABLE if there are no
       data in the current index. On success 0 is returned.
"""
```

In our case 'namelen' is extended from 80 to 84 chars. Then corrupts one byte with 'name[namelen-1] = 0;'.

```
   557  static int get_cert_name(gnutls_x509_crt_t cert, char *name, size_t
namelen)
   558  {
   559          if (gnutls_x509_crt_get_dn_by_oid(cert,
GNUTLS_OID_X520_COMMON_NAME,
   560                                            0, 0, name, &namelen) &&
   561              gnutls_x509_crt_get_dn(cert, name, &namelen)) {
   562                  name[namelen-1] = 0;
   563                  snprintf(name, namelen-1, "<unknown>");
```

```
564                 return -EINVAL;
565         }
566         return 0;
567 }
```

Larry the Git Cow Dev **2020-05-12 16:02:53 UTC** **Comment 10**

The bug has been closed via the following commit(s):

https://gitweb.gentoo.org/repo/gentoo.git/commit/?id=27513d77015771f8604d9a21f388e9846c8c650a

```
commit 27513d77015771f8604d9a21f388e9846c8c650a
Author:     Mike Gilbert <floppym@gentoo.org>
AuthorDate: 2020-05-12 16:01:57 +0000
Commit:     Mike Gilbert <floppym@gentoo.org>
CommitDate: 2020-05-12 16:02:48 +0000

    net-vpn/openconnect: fix buffer overflow in get_cert_name

    Closes: https://bugs.gentoo.org/721570
    Signed-off-by: Mike Gilbert <floppym@gentoo.org>

 .../files/8.09-gnutls-buffer-overflow.patch        | 62 ++++++++++++++++++++++
 ...nect-8.09.ebuild => openconnect-8.09-r1.ebuild} |  3 ++
 2 files changed, 65 insertions(+)
```