LDAP injection via OneDev may leak some LDAP directory information

Low **robinshine** published **GHSA-5864-2496-4xjf** on May 29, 2021

Package

No package listed

Affected versions	Patched versions
<4.4.1	4.4.2

Description

Impact

What kind of vulnerability is it? Who is impacted?

If the LDAP external authentication mechanism is enabled in OneDev, the following code gets executed when a user tries to log in:

LdapAuthenticator.java

```
@Override
public Authenticated authenticate(UsernamePasswordToken token) {
        String fullName = null;
        String email = null;
        Collection<String> groupNames = null;
        Collection<String> sshKeys = null;

        Name userSearchBase;
        try {
                userSearchBase = new CompositeName().add(getUserSearchBase());
        } catch (InvalidNameException e) {
                throw new RuntimeException(e);
        }
        String userSearchFilter = StringUtils.replace(getUserSearchFilter(), "{0}", token.getUsername());
```

token.getUsername() is the username entered by the user in the login form. This gets added to userSearchFilter replacing the {0} placeholder, which is added in the External Authentication configuration. Then, userSearchFilter is used some lines below without further sanitization (except backslashes getting escaped):

NamingEnumeration results = ctx.search(userSearchBase, userSearchFilter, searchControls);
By manipulating the filter, an attacker can send forged queries to the application and explore the LDAP tree using Blind LDAP Injection techniques. In its simplest form, this vulnerability can be exploited by specifying a known username and adding an additional condition to the query, in a way that the obtained error message is different if the condition is met (e.g. trying to log in as admin)(mail=a*, assuming admin is an existing LDAP user, would tell us whether their mail attribute starts with the a character or not).

The specific payload depends on how the User Search Filter property is configured in OneDev (Administration > Security Setting > External Authentication). For my testing, I used a simple (&(uid={0})).

Impact

Information Disclosure (Blind LDAP Injection)

Patches

This issue was fixed in 4.4.2 by escaping special characters in user name

Credits

This issue was discovered and reported by the CodeQL static languages team.

Severity

Low

CVE ID

CVE-2021-32651

Weaknesses

CWE-90