

KYLIN-5421 LDAP user authentication exception

Root Cause

Kylin is not case sensitive when logging in, so users can log in successfully.

But when checking permissions, the username is case sensitive

So the user can log in, but the permission is wrong.

Dev Design

The current spring security ldap framework calls the LdapUserDetailsMapper through the LdapAuthenticationProvider to obtain authentication information and save the return as authentication object for external use.

Therefore, rewrite the LdapUserDetailsMapper expansion point to LdapCaseIgnoreUserDetailsContextMapper, in order to make the expansion point effective, in kylinSecurity.xml inject the object into the LdapUserDetailsService and userAuthProvider.

```
@Autowired
@Qualifier("userService")
private LdapUserService ldapUserService;

@Override
public UserDetails mapUserFromContext(DirContextOperations ctx, String username,
                                     Collection<? extends GrantedAuthority> authorities) {
    String dn = ctx.getNameInNamespace();
    this.logger.debug(LogMessage.format("Mapping user details from context with DN %s", dn))
    LdapUserDetailsImpl.Essence essence = new LdapUserDetailsImpl.Essence();
    essence.setDn(dn);
    Map<String, String> dnMap = ldapUserService.getDnMapperMap();
    String realName = dnMap.get(dn);
    logger.info(LogMessage.format("Ldap real name is %s", realName));
    String passwordAttributeName = "userPassword";
    Object passwordValue = ctx.getObjectAttribute(passwordAttributeName);
    if (passwordValue != null) {
        essence.setPassword(mapPassword(passwordValue));
    }
    essence.setUsername(realName);
    // Add the supplied authorities
    for (GrantedAuthority authority : authorities) {
        essence.addAuthority(authority);
    }

    return essence.createUserDetails();
}
```

The realname is obtained based on the dn resolution.

Root Cause

Kylin在登录时不区分大小写，所以用户可以成功登录。

但在检查权限时，用户名是区分大小写的

所以用户可以登录，但权限是错误的。

Dev Design

当前spring security ldap框架通过LdapAuthenticationProvider 调用LdapUserDetailsMapper 获取认证信息，并保存返回为authentication对象供外部使用。因此，重写LdapUserDetailsMapper拓展点为LdapCaseIgnoreUserDetailsContextMapper，为了使拓展点生效，在kylinSecurity.xml里将该对象注入到LdapUserDetailsService 和userAuthProvider 中。

```
@Autowired
@Qualifier("userService")
private LdapUserService ldapUserService;

@Override
public UserDetails mapUserFromContext(DirContextOperations ctx, String username,
                                      Collection<? extends GrantedAuthority> authorities) {
    String dn = ctx.getNameInNamespace();
    this.logger.debug(LogMessage.format("Mapping user details from context with DN %s", dn));
    LdapUserDetailsImpl.Essence essence = new LdapUserDetailsImpl.Essence();
    essence.setDn(dn);
    Map<String, String> dnMap = ldapUserService.getDnMapperMap();
    String realName = dnMap.get(dn);
    logger.info(LogMessage.format("ldap real name is %s", realName));
    String passwordAttributeName = "userPassword";
    Object passwordValue = ctx.getObjectAttribute(passwordAttributeName);
    if (passwordValue != null) {
        essence.setPassword(mapPassword(passwordValue));
    }
    essence.setUsername(realName);
    // Add the supplied authorities
    for (GrantedAuthority authority : authorities) {
        essence.addAuthority(authority);
    }

    return essence.createUserDetails();
}
```

根据dn解析得到realname。