LDAP authentication by binding user credentials | Tiki Wiki CMS Groupware :: Development
LDAP authentication by binding user credentials

Status
Open

Subject
LDAP authentication by binding user credentials

Version
2.x

Category
- Feature request

Feature
User Administration (Registration, Login & Banning)
External Authentication (LDAP, AD, PAM, CAS, etc)

Resolution status
New

Submitted by
stefricht

Lastmod by
stefricht

Rating
(0)

Description
Our Active Directory is configured not to allow arbitrary LDAP searches for unprivileged users. However, these users can successfully bind to AD's LDAP interface. This would be enough for authentication and we would not need a special account for checking authentication.

I therefore removed parts of the function fetchData in /lib/pear/Auth/Container/LDAP.php:

```
function fetchData($username, $password) { $this->log('Auth_Container_LDAP::fetchData() called.', AUTH_LOG_DEBUG); $err = $this->_prepare(); if ($err !== true) { return PEAR::raiseError($err->getMessage(), $err->getCode()); } $err = $this->_getBaseDN(); if ($err !== true) { return PEAR::raiseError($err->getMessage(), $err->getCode()); } // UTF8 Encode username for LDAPv3 if (@ldap_get_option($this->conn_id, LDAP_OPT_PROTOCOL_VERSION, $ver) && $ver == 3) { $this->log('UTF8 encoding username for LDAPv3', AUTH_LOG_DEBUG); $username = utf8_encode($username); } /* // make search filter $filter = sprintf('(&(%s=%s)%s)', $this->options['userattr'], $this->_quoteFilterString($username), $this->options['userfilter']); // make search base dn $search_basedn = $this->options['userdn']; if ($search_basedn != '' && substr($search_basedn, -1) != ',') { $search_basedn .= ','; } $search_basedn .= $this->options['basedn']; // attributes $searchAttributes = $this->options['attributes']; // make functions params array $func_params = array($this->conn_id, $search_basedn, $filter, $searchAttributes); // search function to use $func_name = $this->_scope2function($this->options['userscope']); $this->log("Searching with $func_name and filter $filter in $search_basedn", AUTH_LOG_DEBUG); // search if (($result_id = @call_user_func_array($func_name, $func_params)) === false) { $this->log('User not found',
```

AUTH_LOG_DEBUG); } elseif (@ldap_count_entries($this->conn_id, $result_id) >= 1) { // did we get some possible results? $this->log('User(s) found', AUTH_LOG_DEBUG); $first = true; $entry_id = null; do { // then get the user dn if ($first) { $entry_id = @ldap_first_entry($this->conn_id, $result_id); $first = false; } else { $entry_id = @ldap_next_entry($this->conn_id, $entry_id); if ($entry_id === false) break; } $user_dn = @ldap_get_dn($this->conn_id, $entry_id); // as the dn is not fetched as an attribute, we save it anyway if (is_array($searchAttributes) && in_array('dn', $searchAttributes)) { $this->log('Saving DN to AuthData', AUTH_LOG_DEBUG); $this->_auth_obj->setAuthData('dn', $user_dn); } // fetch attributes if ($attributes = @ldap_get_attributes($this->conn_id, $entry_id)) { if (is_array($attributes) && isset($attributes['count']) && $attributes['count'] > 0) { // ldap_get_attributes() returns a specific multi dimensional array // format containing all the attributes and where each array starts // with a 'count' element providing the number of attributes in the // entry, or the number of values for attribute. For compatibility // reasons, it remains the default format returned by LDAP container // setAuthData(). // The code below optionally returns attributes in another format, // more compliant with other Auth containers, where each attribute // element are directly set in the 'authData' list. This option is // enabled by setting 'attrformat' to // 'AUTH' in the 'options' array. // eg. $this->options['attrformat'] = 'AUTH' if (strtoupper($this->options['attrformat']) == 'AUTH') { $this->log('Saving attributes to Auth data in AUTH format', AUTH_LOG_DEBUG); unset ($attributes['count']); foreach ($attributes as $attributeName => $attributeValue) { if (is_int($attributeName)) continue; if (is_array($attributeValue) && isset($attributeValue['count'])) { unset ($attributeValue['count']); } if (count($attributeValue)<=1) $attributeValue = $attributeValue[0]; $this->log('Storing additional field: '.$attributeName, AUTH_LOG_DEBUG); $this->_auth_obj->setAuthData($attributeName, $attributeValue); } } else { $this->log('Saving attributes to Auth data in LDAP format', AUTH_LOG_DEBUG); $this->_auth_obj->setAuthData('attributes', $attributes); } } } @ldap_free_result($result_id); // need to catch an empty password as openldap seems to return TRUE // if anonymous binding is allowed */ $user_dn = $username; if ($password != "") { $this->log("Bind as $user_dn", AUTH_LOG_DEBUG); // try binding as this user with the supplied password if (@ldap_bind($this->conn_id, $user_dn, $password)) { $this->log('Bind successful', AUTH_LOG_DEBUG); // check group if appropiate if (strlen($this->options['group'])) { // decide whether memberattr value is a dn or the username $this->log('Checking group membership', AUTH_LOG_DEBUG); $return = $this->checkGroup(($this->options['memberisdn']) ? $user_dn : $username); $this->_disconnect(); return $return; } else { $this->log('Authenticated', AUTH_LOG_DEBUG); $this->_disconnect(); return true; // user authenticated } // checkGroup } // bind } // non-empty password // } while ($this->options['try_all'] == true); // interate through entries // } // get results // default $this->log('NOT authenticated!', AUTH_LOG_DEBUG); $this->_disconnect(); return false; }

If would be nice to have a checkbox in the LDAP part of the admin page for selecting this behaviour.

Importance
1 low

Priority
5

Demonstrate Bug (Tiki 19+)

Please demonstrate your bug on show2.tiki.org

Version: trunk ▼

Demonstrate Bug (older Tiki versions)

Please demonstrate your bug on show.tikiwiki.org

Version: 18.x ▼

Ticket ID
2226

Created
Monday 15 December, 2008 16:10:44 GMT-0000
by Unknown

LastModif
Tuesday 30 September, 2014 16:55:38 GMT-0000

Comments

vanillaxtrakt 09 Feb 09 21:00 GMT-0000

We had the same problem; our OpenLDAP is configured to reject anonymous binds. I was able to use a similar fix, except I had to set $user_dn differently in LDAP.php:

```
$user_dn = $this->options['userattr'] . "=" . $username . "," . $this->options['userdn'] .
"," . $this->options['basedn'];
```

A checkbox would be really nice. Actually, for us it would be essential. It's too much of a hassle to keep up with changes like this to the actual code of an application which are only going to be broken every time the application is upgraded. Because of this, we ended up choosing different wiki/CMS software.

Here are all of the changes I made:

```
--- LDAP.php.dist  2009-02-09 12:39:32.000000000 -0600

+++ LDAP.php    2009-02-09 12:42:04.000000000 -0600

@@ -563,7 +563,7 @@

             $username = utf8_encode($username);

         }



-        // make search filter

+ /*     // make search filter

         $filter = sprintf('(&(%s=%s)%s)',

                           $this->options['userattr'],

                           $this->_quoteFilterString($username),

@@ -659,6 +659,7 @@
```

```
                // need to catch an empty password as openldap seems to return TRUE

                // if anonymous binding is allowed

+ */             $user_dn = $this->options['userattr'] . "=" . $username . "," .
$this->options['userdn'] . "," . $this->options['basedn'];

                if ($password != "") {

                    $this->log("Bind as $user_dn", AUTH_LOG_DEBUG);



@@ -680,8 +681,8 @@

                        } // checkGroup

                    } // bind

                } // non-empty password

-            } while ($this->options['try_all'] == true); // interate through entries

-        } // get results

+ //         } while ($this->options['try_all'] == true); // interate through entries

+ //     } // get results

          // default

          $this->log('NOT authenticated!', AUTH_LOG_DEBUG);

          $this->_disconnect();
```

Attachments

filename	created	hits	comment	version	filetype

No attachments for this item

The original document is available at
https://dev.tiki.org/item2226-LDAP-authentication-by-binding-user-credentials