KLOXO SENSITIVE INFORMATION DISCLOSURE VULNERABILITY

SUMMARY:

Kloxo contains 2 vulnerabilities:

- The first one could allow an authenticated remote attacker (client or auxiliary) to get almost any info from DB, for example passwords of other users (including administrators), credentials for DB connection, etc.
- The second one allows assigning "admin" role to current client (After gathering credentials of user (reseller or admin) who has created current client).

Below I provided recommendations how these problems may be fixed.

TECHNICAL DETAILS:

Information Disclosure

This flaw is caused by insufficient validation of permission during processing "***action=getproperty***" web command. There is no validation that current user is "owner" of desired ***$object***. So an attacker can get value of any column from any DB table only by "name" (similar to ID - this column exists in all tables but it has text format).

So for proper fix we should add such validation.

In this report all listings are from **Kloxo 6.1.19** (in various versions lines numbers can be different). **Kloxo-MR 7.0.0.b-2015012701** has the same code so we should make the same changes.

The fixed code for Kloxo (**NL#** lines added):

Listing of /htmllib/lib/commandlinelib.php:

```
/* 186 */ function __cmd_desc_getproperty($param)
/* 187 */ {
/* 188 */    global $gbl, $sgbl, $login, $ghtml;
/* 189 */    if (isset($param['name']) && isset($param['class'])) {
/* 190 */        $name = $param['name'];
/* 191 */        $class = $param['class'];
/* 192 */        $object = new $class(null, 'localhost', $name);
/* 193 */        $object->get();
/* 194 */        if ($object->dbaction === 'add') {
/* 195 */            throw new lxException('object_doesnt_exist', 'name', $name);
/* 196 */        }
/* NL1 */        if (!$object->checkIfSomeParent($login->getClName())) {
/* NL2 */            throw new lxException("the_object_doesnt_exist_under_you", "", $object->nname);
/* NL3 */        }
/* 197 */    } else {
/* 198 */        $object = $login;
/* 199 */    }
/* … */
```

After this fix customer will be able to get only "own" info…

Risks of the fix:

- If there is any user story when client should have access to info what is not "under" him (for example regular customer "wants" to know some contact info of another regular customer) – he

will not able to do this. From security POV users should not be able to get such info but if it is needed – only this "needed" info should be allowed.

- If current user wants to get any info about himself he should request this info without '***name'*** and/or '***class'*** parameters (I believe that it's logical to do this without these parameters but maybe there is some "unusual" calls…).

Privilege escalation

Reseller cannot create privileged users (e.g. admins) but he can modify customers and can assign admin role to them (or he can create a new customer and after that assign admin role to him).

BTW I've noticed that there is proper validation if *subaction*=***information:***

/webcommand.php?login-class=client&login-name=***<username_of_reseller>***&login-password=***<password>***&action=**update**&class=**client**&name=***<user_created_by_reseller>***&v-**cttype**=**admin**&subaction=***information***

The code what implements this validation:

Listing of /htmllib/lib/client/clientcorelib.php:

```
/* 575 */ function updateInformation($param)
/* 576 */ {
/* 577 */    global $gbl, $sgbl, $login, $ghtml;
/* 578 */    if_demo_throw_exception('info');
/* 579 */    if (isset($param['cttype'])) {
/* 580 */       if (!$this->isAdmin()) {
/* 581 */          if ($this->getParentO()->isGt($param['cttype'])) {
/* 582 */             throw new lxException("parent_doesnt_have_privileges", 'cttype', '');
/* 583 */          }
/* 584 */       }
/* 585 */    }
/* … */
```

So we can use this code but we should put it in the function that will be executed every time not depend on ***subaction*** parameter. For example **commandUpdate()** of **ClientBase** class

The fixed code for Kloxo (**NL#** lines added):

Listing of /htmllib/lib/client/clientbaselib.php:

```
/* 1093 */ function commandUpdate($subaction, $param)
/* 1094 */ {
/* NL1 */    if_demo_throw_exception('info');
/* NL2 */    if (isset($param['cttype'])) {
/* NL3 */       if (!$this->isAdmin()) {
/* NL4 */          if ($this->getParentO()->isGt($param['cttype'])) {
/* NL5 */             throw new lxException("parent_doesnt_have_privileges", 'cttype', '');
/* NL6 */          }
/* NL7 */       }
/* NL8 */    }
/* 1095 */    switch($subaction) {
/* … */
```

Risks of the fix:

I don't see any serious risk here. But maybe it is better to put this code in other place but anyway this validation should be executed with any ***subaction*** parameter.