Talos Vulnerability Report

TALOS-2020-1177

phpGACL template multiple cross-site scripting vulnerabilities

JANUARY 27, 2021

CVE NUMBER

CVE-2020-13562, CVE-2020-13563, CVE-2020-13564

Summary

Multiple cross-site scripting vulnerabilities exist in the template functionality of phpGACL 3.3.7. A specially crafted HTTP request can lead to arbitrary JavaScript execution. An attacker can provide a crafted URL to trigger this vulnerability.

Tested Versions

phpGACL 3.3.7
OpenEMR 5.0.2
OpenEMR development version 6.0.0 (commit babec93f600ff1394f91ccd512bcad85832eb6ce)

Product URLs

http://phpgacl.sourceforge.net/

CVSSv3 Score

9.6 - CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:H

CWE

CWE-80 - Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS)

Details

phpGACL is a PHP library that allows developers to implement permission systems via a Generic Access Control List.

The latest version of this library has been found to be used in OpenEMR, as such the tests have been performed against an OpenEMR instance.

Across the whole codebase of phpGACL `smarty` is used for templating, however multiple variables are not escaped correctly when rendered, leading to cross-site scripting (XSS).

The following is an (incomplete) list of code paths that lead to XSS, caused by missing escape of the input parameters that can be injected by an attacker via GET or POST request. Note that other similar XSS code paths exist under the same `gacl/` subdirectory.

CVE-2020-13562 - phpGACL template action parameter cross-site scripting vulnerabiliy

In `admin/acl_admin.php`, the GET parameter `action` leads to an XSS:

```
...
if (isset($_GET['action'])) {
        $smarty->assign('action', $_GET['action']);    [1]
}

$smarty->assign('current','acl_admin');
$smarty->assign('page_title', 'ACL Admin');

$smarty->assign('phpgacl_version', $gacl_api->get_version() );
$smarty->assign('phpgacl_schema_version', $gacl_api->get_schema_version() );
$smarty->display('phpgacl/acl_admin.tpl');             [2]
?>
```

At [1] the `action` is fetched from the query string, and passed to `smarty`. The template used for rendering is `phpgacl/acl_admin.tpl` [2]:

```
...
[ <a href="group_admin.php?group_type=aro&return_page={$SCRIPT_NAME}?action={$action}&acl_id={$acl_id}">Edit</a> ]
...
```

The line above is executed twice in the template, which renders the `action` argument verbatim, leading to an XSS.

Exploit Proof of Concept

This issue has been reproduced by testing against OpenEMR, which ships the latest version of phpGACL.
The following request exploits the XSS in `acl_admin.php`, via the `action` argument:

```
http://openemr.dev/gacl/admin/acl_admin.php?action=\"><script>alert(1)</script>
```

CVE-2020-13563 - phpGACL template group_id parameter cross-site scripting vulnerabiliy

In `admin/assign_group.php`, the GET parameter `group_id` leads to an XSS when an invalid or no `action` is specified via POST:

```
...
//Get group name.
$group_data = $gacl_api->get_group_data($_GET['group_id'], $group_type);
$smarty->assign('group_name', $group_data[2]);

$smarty->assign('group_id', $_GET['group_id']);                                [1]
...
$smarty->assign('group_type', $group_type);
$smarty->assign('object_type', $object_type);
$smarty->assign('return_page', $_SERVER['REQUEST_URI'] );

$smarty->assign('current','assign_group_'. $group_type);
$smarty->assign('page_title', 'Assign Group - '. strtoupper($group_type));

$smarty->assign('phpgacl_version', $gacl_api->get_version() );
$smarty->assign('phpgacl_schema_version', $gacl_api->get_schema_version() );

$smarty->display('phpgacl/assign_group.tpl');                                  [2]
?>
```

At [1] the `group_id` is fetched from the query string, and passed to `smarty`. The template used for rendering is `phpgacl/assign_group.tpl` [2]:

```
...
        {include file="phpgacl/pager.tpl" pager_data=$paging_data link="?group_type=$group_type&group_id=$group_id&"}   [3]
          </td>
        </tr>
        <tr>
        ...
</table>
<input type="hidden" name="group_id" value="{$group_id}">   [4]
```

At [4] the `group_id` is rendered verbatim, leading to an XSS. The `group_id` is also passed to template `pager.tpl` via the “link” parameter [3], leading to another XSS:

```
...
      <a href="{$link}page=1">|&lt;</a> <a href="{$link}page={$paging_data.prevpage|escape:'html'}">&lt;&lt;</a>
...
      <a href="{$link}page={$paging_data.nextpage|escape:'html'}">&gt;&gt;</a> <a href="{$link}page=
{$paging_data.lastpageno|escape:'html'}">&gt;|</a>
```

Exploit Proof of Concept

This issue has been reproduced by testing against OpenEMR, which ships the latest version of phpGACL.
The following request exploits the XSS in `assign_group.php`, via the `group_id` argument:

```
http://openemr.dev/gacl/admin/assign_group.php?group_id="><script>alert(1)</script>
```

CVE-2020-13564 - phpGACL template acl_id parameter cross-site scripting vulnerabiliy

In `admin/acl_admin.php`, the GET parameter `acl_id` leads to an XSS when an invalid or no `action` is specified via POST:

```
...
        if (isset($_GET['acl_id'])) {
            $smarty->assign('acl_id', $_GET['acl_id'] );                       [1]
        }

        break;
}

//$smarty->assign('return_page', urlencode($_SERVER[REQUEST_URI]) );
if (isset($_GET['return_page'])) {
    $smarty->assign('return_page', $_GET['return_page']);
}
if (isset($_GET['action'])) {
    $smarty->assign('action', $_GET['action']);
}

$smarty->assign('current','acl_admin');
$smarty->assign('page_title', 'ACL Admin');

$smarty->assign('phpgacl_version', $gacl_api->get_version() );
$smarty->assign('phpgacl_schema_version', $gacl_api->get_schema_version() );
$smarty->display('phpgacl/acl_admin.tpl');                                      [2]
?>
```

At [1] the `acl_id` is fetched from the query string, and passed to `smarty`. The template used for rendering is `phpgacl/acl_admin.tpl` [2]:

```
...
[ <a href="group_admin.php?group_type=aro&return_page={$SCRIPT_NAME}?action={$action}&acl_id={$acl_id}">Edit</a> ]
...
[ <a href="group_admin.php?group_type=axo&return_page={$SCRIPT_NAME}?action={$action}&acl_id={$acl_id}">Edit</a> ]
...
<input type="hidden" name="acl_id" value="{$acl_id}">
...
```

In three different spots in the template, the `acl_id` argument is rendered verbatim, leading to an XSS.

Exploit Proof of Concept

This issue has been reproduced by testing against OpenEMR, which ships the latest version of phpGACL.
The following request exploits the XSS in `acl_admin.php`, via the `acl_id` argument:

```
http://openemr.dev/gacl/admin/acl_admin.php?acl_id="><script>alert(1)</script>
```

Timeline

2020-10-23 - Vendor Disclosure

2021-01-05 - Vendor Patched
2021-01-27 - Public Release

CREDIT

Discovered by Claudio Bozzato of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT TALOS-2020-1178

NEXT REPORT TALOS-2020-1180