Budapesti Műszaki Szakképzési Centrum

Neumann János Informatikai Technikum

***Szakképesítés neve:*** Informatikai rendszer- és alkalmazás-üzemeltető technikus

***száma:*** 5-0612-12-02

**VIZSGAREMEK**

Terror Háza Múzeum

Informatikai hálózati infrastruktúra kiépítése és konfigurálása  
Szerverek telepítése, konfigurálása, üzemeltetése

Agot Bence, Hadady Patrik, Törőcsik Ádám  
2/14.B

Budapest, 2022.

# Tartalomjegyzék

[Tartalomjegyzék 2](#_Toc134272706)

[Történet 4](#_Toc134272707)

[A tervezés 4](#_Toc134272708)

[Topológia 4](#_Toc134272709)

[IP címzés 4](#_Toc134272710)

[Megvalósítás 5](#_Toc134272711)

[Terror háza 5](#_Toc134272712)

[2. és 3. rétegbeli elemek 5](#_Toc134272713)

[Load Balancing 5](#_Toc134272714)

[HSRP 6](#_Toc134272715)

[Spanning Tree 7](#_Toc134272716)

[EtherChannel 7](#_Toc134272717)

[PAT 8](#_Toc134272718)

[DHCP 8](#_Toc134272719)

[Tunnel 8](#_Toc134272720)

[OSPF 8](#_Toc134272721)

[Forgalomirányítás telephelyek között (ISP) 9](#_Toc134272722)

[DNS 9](#_Toc134272723)

[Kertész Intézet 9](#_Toc134272724)

[12. Kerület 9](#_Toc134272725)

[Báltéri utca 9](#_Toc134272726)

[Szerverek 10](#_Toc134272727)

[Windows Szerver 10](#_Toc134272728)

[Exchange 10](#_Toc134272729)

[Linux Web Szerver 10](#_Toc134272730)

[Docker Compose: 11](#_Toc134272731)

[nginx image: 12](#_Toc134272732)

[nginx config: 13](#_Toc134272733)

[Wordpress config: 14](#_Toc134272734)

[Hálózat Programozás 15](#_Toc134272735)

[A program működése: 15](#_Toc134272736)

[Fájlok 15](#_Toc134272737)

[Program kódja: 16](#_Toc134272738)

[Program működése: 18](#_Toc134272739)

[Ábrajegyzék 19](#_Toc134272740)

# Történet

A Terror Háza Múzeum informatikai infrastruktúrát szeretne fejleszteni a látogatók vizuális látványának és a dolgozók hatékonyságának növelése érdekében. A múzeum vezetősége felkeresett minket, egy informatikai rendszer fejlesztésének céljából, melyben ötlettervét megmutatva munkához is kezdtünk. Felmértük a múzeum és a telephelyeinek jelenlegi kialakítását, majd hozzákezdtünk a tervezéshez.

# A tervezés

## Topológia

A tervezési munkálatokat megkezdtük. Felmértük a hálózat kiszolgálásának igényét, felmértük a telephelyeket. A következő döntésre jutottunk: a Terrorháza fő hálózatában 2. és 3. rétegi redundanciát fogunk használni a hálózat hibamentes működésének érdekében. A fő telephely az ISP-hez két HSRP-t használó router segítségével lesz összekötve.

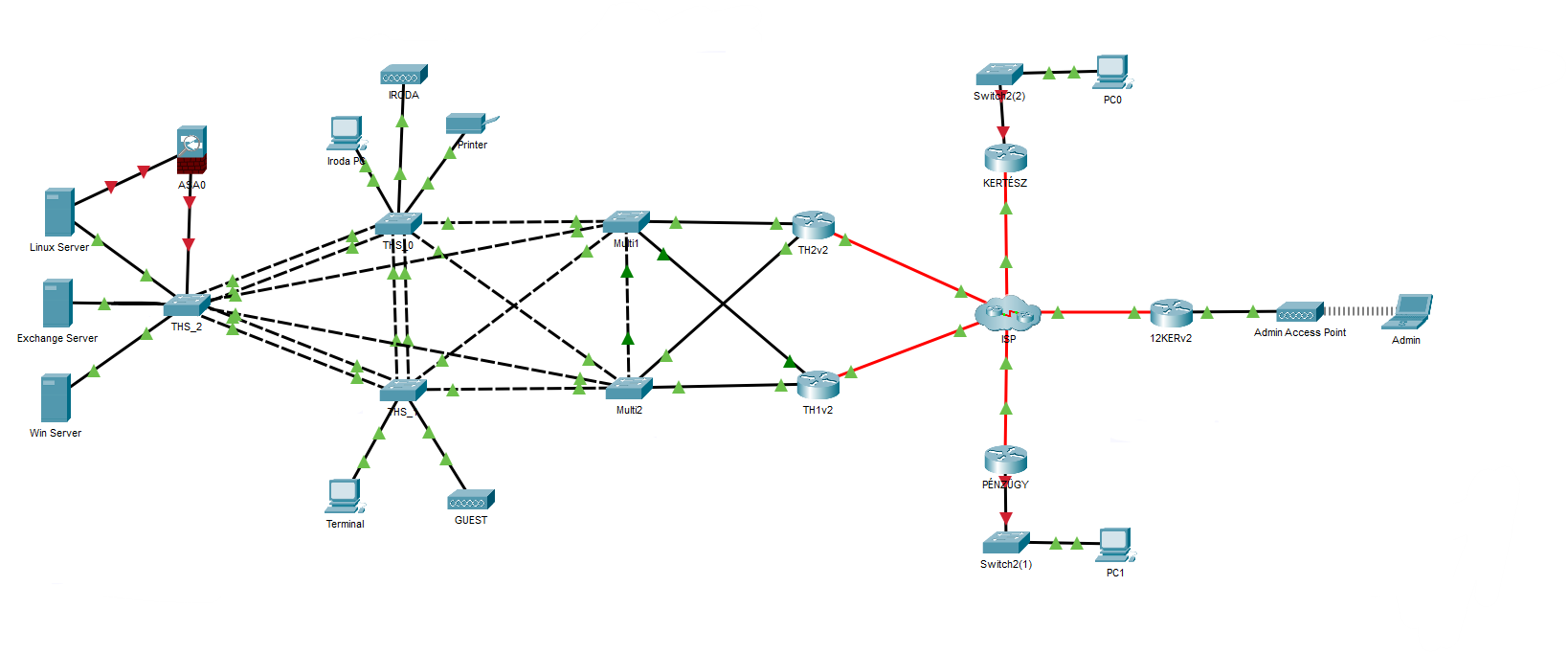
## IP címzés

A topológia felépítését követően, nekiláttunk a kiosztható IPv4-es és IPv6-os címek felosztásának és kiszámolásának. Elkészítettük a Fő telephely VLAN táblázatát és IP cím kiosztását.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Vlan** | **Name** | **Network** | **Default Router** | **DHCP** | **DHCP Server** | **DNS Server** |
| 10 | SZERVERTEREM | 192.168.10.0/24 | 192.168.10.1 |  |  | 8.8.8.8 |
| 20 | IRODA | 192.168.20.0/24 | 192.168.20.1 | 100-200 | 192.168.10.22 | 192.168.10.22 |
| 30 | WIFI | 192.168.30.0/24 | 192.168.30.1 | 10-254 | 192.168.10.22 | 192.168.10.22 |
| 40 | TERMINAL | 192.168.40.0/24 | 192.168.40.1 |  |  | 192.168.10.22 |
| 50 | WEB | 192.168.50.0/24 | 192.168.50.1 |  |  | 8.8.8.8 |
| 99 | MANAGEMENT | 192.168.99.0/24 | 192.168.99.1 | 10-254 | 192.168.10.22 | 192.168.10.22 |

# Megvalósítás

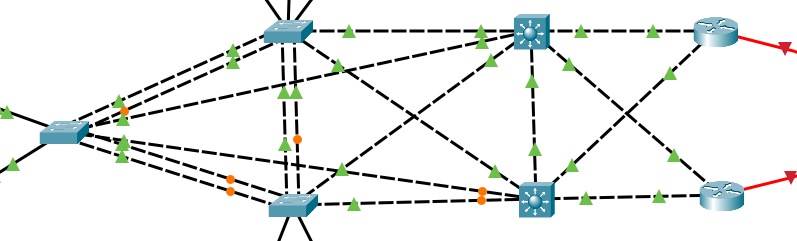
A topológia 4 fő elemből áll, mely a 4 telephely: ATerror Háza, Kertész intézet, 12. kerületi iroda, valamint a Báltéri utcai iroda. A topológia feléptése miatt a „Terror űrprogram” fantázia névvel lett illetve.



1. ábra A Terror Háza teljes hálózatának topológiája

## Terror háza

### 2. és 3. rétegbeli elemek

A Terror Háza hálózatában második és harmadik rétegű redundancia is jelen van. A második rétegben link agregationtalálható feszítőfával kiegészítve. A harmadik réteg switchei és routerei HSRP segítségével lehetővé teszik a load balancing létrejöttét.

2. ábra 2. és 3. rétegbeli kapcsolók

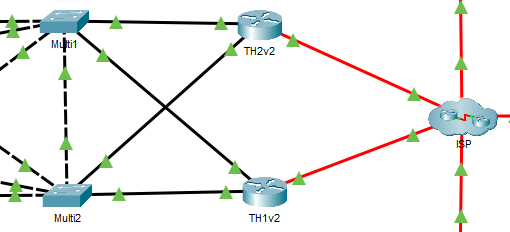
### Load Balancing

A load balancing egy olyan fogalom, mely meghatározza, hogy az adatforgalom több kijáratot igénybe véve csökkenti az egyes eszközök terheltségét, ezáltal kevésbe lassítva a hálózatot.

### HSRP

A képen szöveg látható

Automatikusan generált leírásA képen asztal látható

Automatikusan generált leírásA HSRPtöbb célt is szolgálhat. Az egyik megvalósítása lehet Failover konfiguráció, vagy használhatjuk akár load balancingra is. A jelen esetben mind2 jelen van, ugyanis a HSRP segítségével beállítottjuk, hogy bizonyos VLAN-okat adott router kezelje elsőként. Ezáltal normál üzemben elszeparálva a belső hálózat forgalmát a látogatók hálózati forgalmától.

3. ábra TH2 router HSRP konfigurációja

4. ábra TH1 router HSRP konfigurációja

5. ábra HSRP felépítése

### Spanning Tree

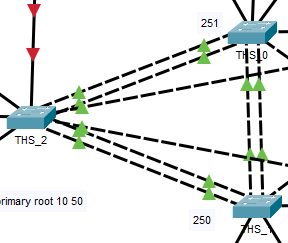
A képen asztal látható

Automatikusan generált leírásA feszítőfa a második réteg redundanciáját segíti elő mivel a körbe kötött kapcsolók nem alkothatnak hurkot. A hálózatban a Rapid PVST protokollt használtuk, hogy lehetséges legyen a VLAN-ok, és EtherChannel kezelése.

6. ábra feszítőfa információi

### EtherChannel

A képen szöveg látható

Automatikusan generált leírásAz EtherChannel avagy link agregation lehetővé tesz, hogy a switch-ek több kábellel is össze legyenek kötve amik képesek egymás szerepét átvenni, ha egyik üzemképtelenné válik ezzel biztosítva redundanciát, emellett itt is szóba hozható a load balancing. Mi a Cisco féle PAGP kapcsolatot alkalmazzuk 0

7. ábra EtherChannel jelenléte a topológiában

8. ábra EtherChannel konfigurációja

### PAT

A fő telephely belső hálózatáról kimenő összes forgalom PAT segítségével cím és port fordítást alkalmazunk, a belső hálózat védése érdekében, valamint cím spórolás céljából.

### DHCP

A DHCP fő lényege az, hogy a hálózatunkban dinamikusan kerüljenek az IP címek kiosztásra. Ezzel elősegítve a címütközések előfordulásának csökkentését. A cégen belül több IP tartományban is használunk DHCP szervert.

### Tunnel

A telephelyek összekapcsolásáhozl Tunnel rendszert alkalmazunk, ezáltalegy hálózatként tudjuk kezelni a több telephely megoldását.

### OSPF

A telephelyek között létrehozott Tunnelen keresztül OSPF forgalom irányítás van. Egy gyors és megbízható, ezen felül okos routing protocol.

## Forgalomirányítás telephelyek között (ISP)

### DNS

Az ISP-nél el van helyezve egy DNS szerver ami <insert IP address here> címen érhető el.

## Kertész Intézet

A telephely belső hálózata a 192.168.0.0/24. A helyi Windows szerver a 192.168.0.5-ös címet kapta a többi eszköz pedig a .10-.254-es címtartományból kap IP címet a forgalom irányítótól DHCP segítségével. A PAT a belső címeket a gigabit0/0/0 interfész 12.0.0.2-es címére fordítja.

## 12. Kerület

A Terror Háza hálózatának adminisztrációja innen történik VPN kapcsolton keresztül. Ez mind a 99-es MANAGEMENT vlan­-on keresztül történik.

## Báltéri utca

A telephely belső hálózata a 192.168.0.0/24. A helyi Windows szerver a 192.168.0.5-ös címet kapta a többi eszköz pedig a .10-.254-es címtartományból kap IP címet a forgalom irányítótól DHCP segítségével. A PAT a belső címeket a gigabit0/0/0 interfész 14.0.0.2-es címére fordítja.

# Szerverek

## Windows Szerver

A Windows szerver a 192.168.10.22-es címen érhető el, ami a 10-es VLAN része. A szerver a halózaban DNS, DHCP és Active Directory szolgáltatásokat, File megosztást, Windows Server backupot biztosít. Az operációs rendszerünk Windows Server 2022.

## Exchange

A Microsoft Exchange mail szerver a 192.168.10.19-es címen érhető el és egy Windows 2019 szerverről működik, ami a 10-es VLAN része.

## Linux Web Szerver

A szerver CentOS Stream 9-et használ és egy debian:stable és egy mariadb docker konténert futtat és így szolgáltatja a múzeum weboldalát. A szervert távolról SSH kapcsolattal lehet csak elérni, kulcs alapú hitelesítéssel a ’remote’ felhasználón keresztül. Csak a tartományi és távoli rendszergazdák kulcsait fogadja el a szerver. A felhasználó nem rendelkezik root jogosultságokkal, ha a konténerek kezelésére van szükség root felhasználóvá kell válni.

### Docker Compose:

A webes szolgáltatás egy letöltött mariadb és egy debian:stable image-ből készült konténerből áll. A konténerek beállításai az alábbi docker-compose.yaml fájlban találhatóak.

version: '3.1'

services:

  mariadb:

    image: mariadb

    restart: always

    environment:

      MYSQL\_DATABASE: wpdb

      MYSQL\_USER: dbuser

      MYSQL\_PASSWORD: dbpass

      MYSQL\_ROOT\_PASSWORD: dbroot

    volumes:

      - mariadb:/var/lib/mysql

  nginx:

    build:

      dockerfile: dockerfile

    command: bash -c 'service php7.4-fpm start && service nginx start && sleep infinity'

    restart: always

    ports:

      - 80:80

      - 443:443

    depends\_on:

      - mariadb

    links:

      - mariadb

    volumes:

      - nginx:/var/www/wordpress

volumes:

  mariadb:

  nginx:

### nginx image:

Az alábbi dockerfile felel az nginx image létrehozásáért, amit a docker compose használ a mariadb image-el együtt.

FROM debian:stable

RUN apt update

RUN apt upgrade -y

RUN apt install nginx php7.4 php7.4-fpm wget wordpress -y

COPY 'nginx.conf' '/etc/nginx/nginx.conf'

RUN wget 'https://wordpress.org/latest.tar.gz' -O '/root/wp.tar.gz'

RUN tar -xf '/root/wp.tar.gz' -C '/var/www/'

COPY 'wp-config.php' '/var/www/wordpress/wp-config.php'

COPY 'wordpress.crt' '/var/www/wordpress/wordpress.crt'

COPY 'wordpress.key' '/var/www/wordpress/wordpress.key'

RUN chown -R www-data:www-data '/var/www/wordpress/'

RUN ln -sf '/usr/share/zoneinfo/Europe/Budapest' '/etc/localtime'

### nginx config:

Az alábbi nginx.conf fájl bemásolásra kerül az nginx image-be létrehozásakor és az nginx szolgáltatás működését határozza meg.

user www-data www-data;

worker\_processes auto;

http {

        include mime.types;

        ssl\_session\_cache       shared:SSL:10m;

        ssl\_session\_timeout     10m;

        server {

                listen 80;

                listen [::]:80;

                listen 443 ssl;

                listen [::]:443 ssl;

                ssl\_certificate         /var/www/wordpress/wordpress.crt;

                ssl\_certificate\_key     /var/www/wordpress/wordpress.key;

                ssl\_protocols           TLSv1 TLSv1.1 TLSv1.2 TLSv1.3;

                ssl\_ciphers             HIGH:!aNULL:!MD5;

                root /var/www/wordpress;

                server\_name web.th.hu www.web.th.hu;

                index index.php index.html;

                location / {

                        try\_files $uri $uri/ =404;

                }

                location ~ \.php$ {

                        include snippets/fastcgi-php.conf;

                        fastcgi\_pass unix:/var/run/php/php7.4-fpm.sock;

                }

        }

}

### Wordpress config:

Az alábbi wp-config.php fájl bemásolásra kerül az nginx image-be létrehozásakor és a wordpress beállításait adja meg. A fájl a felcsatolt /var/www/wordpress mappába kerül, így az esetleges változások megmaradnak a konténer újraindítása után is.

<?php

/\*\* The name of the database for WordPress \*/

define( 'DB\_NAME', 'wpdb' );

/\*\* Database username \*/

define( 'DB\_USER', 'dbuser' );

/\*\* Database password \*/

define( 'DB\_PASSWORD', 'dbpass' );

/\*\* Database hostname \*/

define( 'DB\_HOST', 'mariadb' );

/\*\* Database charset to use in creating database tables. \*/

define( 'DB\_CHARSET', 'utf8' );

/\*\* The database collate type. Don't change this if in doubt. \*/

define( 'DB\_COLLATE', '' );

define('AUTH\_KEY',         'cQIZOl[ToY{gQ-j+\*c5DS^?RzbPtv\_2!~\*A2-0z+%q~H@teN$Yv\*HT[^|1c{c7-3');

define('SECURE\_AUTH\_KEY',  'RwzJIuO074G$yWh}xa[0|aW/@6${Byq4L2,X<|+P>l|lMu/j(iF%>sSloEVitI$8');

define('LOGGED\_IN\_KEY',    '3`+=3DUH#mAj-ebw&rpR4:lSll}NLRof>M}RHkub=)@)<qU|o6+V]sBmE>[AXnxx');

define('NONCE\_KEY',        'S%8p1.#Una$EnF/7+Ob-LeKtKP\*|yC%9}s<WDJ^CKA\*aE>7\*?`/ {;&~SO26n.Ej');

define('AUTH\_SALT',        '.v?P{rO[Ac&b}LLp\_KwxpR-eOt~:XI^6$z+(VutQAj}#XZA\*ROHuR|Pfz[v{57Y;');

define('SECURE\_AUTH\_SALT', '+0n+-E6=pKK]w-wSAPCrGg)Jge+-3pPO)/dJy&]9Y(.s$sh19g-z;@hdcu|u&%<!');

define('LOGGED\_IN\_SALT',   'wKA{Is>;2IDV2.<YM(f-l/ha\_HeSfOude$$-?UE%o~+M((|FW|rR\*/dp%aTP[NGJ');

define('NONCE\_SALT',       'iK<:\*G|n`A89j:=]7Qw^hhH2a6#:Xye-;<|9p?j@.4\*<A-lC#1h WyFyWMV%]W[c');

$table\_prefix = 'wp\_';

define( 'WP\_DEBUG', false );

/\*\* Absolute path to the WordPress directory. \*/

if ( ! defined( 'ABSPATH' ) ) {

        define( 'ABSPATH', \_\_DIR\_\_ . '/' );

}

/\*\* Sets up WordPress vars and included files. \*/

require\_once ABSPATH . 'wp-settings.php';

# Hálózat Programozás

A program célja az IP cím alapján megadott eszközökön lecserélni a megadott SSH felhasználó jelszavát egy új véletlenszerűen generáltra. Ezt havi rendszerességgel teszi meg a múzeum egyik Linux eszközéről.

A program kódja megtekinthető a [GitHub](https://github.com/Toarexer/2022-2023_Vizsgaremek/blob/main/netmiko/2/repass.py)-on.

### A program működése:

1. A python script elindítható argumentumok nélkül mely esetben a hosts.csv fájlban található összes eszközbe megpróbál majd bejelentkezni és leszót módosítani.

Amennyiben a programnak megadunk argumentumokat akkor feltételezi, hogy mind érvényes IP címek és csak azoknak eszközöknek próbálja meg a jelszavát lecserélni, amiknek a címe szerepel az argumentumok között.

1. Ha a hosts.csv vagy errors.csv fájl nem létezik a program létrehozza őket.
2. Beolvassa a hosts.csv fájlt és kiolvassa belőle az eszközök adatait. Ha egy sort nem tudott értelmezni, azt kiírja az errors.csv fájlba.
3. A program ez után szortírozza az eszközöket és egyesével végrehajtja a következőket:
4. Generál egy új 16 karakter hosszú jelszót.
5. Bejelentkezik az eszközre a régi jelszó segítségével.
6. Lecseréli a régi jelszót az újonnan generáltra.
7. Menti az eszköz konfigurációját, majd kilép belőle.
8. A hosts.csv fájlt átnevezi hosts.prev.csv-nek.
9. Kiírja az új adatokat egy új hosts.csv fájlba.

### Fájlok

* A hosts.csv a következő adatokat tárolják pontosvesszővel elválasztva:

1. IP cím
2. Port
3. Felhasználónév
4. Jelenlegi jelszó
5. Privilegizált mód jelszava
6. Sikeres volt-e a legutolsó jelszó csere

* A hosts.prev.csv mindig az utolsó futás előtti állapotát tárolja a hosts.csv fájlnak.
* Az errors.csv azokat a sorokat tárolja, amiket a program a hosts.csv fájlból olvasott be, de nem tudta azokat formázási hiba miatt értelmezni. A fájl folyamatosan gyűjti az újabb hibákat anélkül, hogy törölne belőle bármit is, ezért manuálisan kell a fájlt üríteni vagy törölni.

### Program kódja:

#!/usr/bin/env python3

from hashlib import sha256

from netmiko import ConnectHandler

from os import chdir, rename

from os.path import dirname, exists

from random import randint

import sys

def readcsv(path: str, errpath: str):

    arr = []

    with open(path, "r") as f:

        with open(errpath, "a") as e:

            for line in [x.removesuffix("\n") for x in f.readlines()]:

                args = line.split(";")

                try:

                    arr.append(

                        {

                            "ip": args[0],

                            "port": int(args[1]),

                            "user": args[2],

                            "pass": args[3],

                            "secret": args[4],

                            "success": bool(args[5])

                        }

                    )

                except:

                    print("Failed to parse: " + line, file=sys.stderr)

                    e.write(line + "\n")

    return arr

def writecsv(path: str, hosts: list):

    open(path, "w").close()

    with open(path, "a") as f:

        for host in hosts:

            print(host["ip"],

                  host["port"],

                  host["user"],

                  host["pass"],

                  host["secret"],

                  host["success"],

                  sep=";", file=f)

chdir(dirname(\_\_file\_\_))

csvfile = "hosts.csv"

errfile = "errors.txt"

if not exists(csvfile):

    open(csvfile, "w").close()

if not exists(errfile):

    open(errfile, "w").close()

hosts = readcsv(csvfile, errfile)

for host in filter(lambda h: (len(sys.argv) == 1) or (h["ip"] in sys.argv[1:]), hosts):

    args = {

        "device\_type": "cisco\_ios",

        "ip": host["ip"],

        "port": host["port"],

        "username": host["user"],

        "password": host["pass"],

        "secret": host["secret"],

    }

    newpass = sha256(str(randint(0, 0xFFFFFFFF)).encode()).hexdigest()[:16]

    commands = [

        "username " + host["user"] + " secret " + newpass,

        "do write"

    ]

    print("Trying " + host["ip"])

    try:

        with ConnectHandler(\*\*args) as nc:

            nc.enable()

            nc.send\_config\_set(commands)

        print("Successfully set password for " + host["ip"])

        host["pass"] = newpass

        host["success"] = True

    except:

        print("Failed to set password for " + host["ip"], file=sys.stderr)

        host["success"] = False

rename(csvfile, csvfile.replace(".csv", ".prev.csv"))

writecsv(csvfile, hosts)

### Program működése:

# Ábrajegyzék

[1. ábra A Terror Háza teljes hálózatának topológiája 4](#_Toc133664592)

[2. ábra 2. és 3. rétegbeli kapcsolók 4](file:///C:\GitHub\2022-2023_Vizsgaremek\dokumentáció.docx#_Toc133664593)

[3. ábra TH2 router HSRP konfigurációja 5](file:///C:\GitHub\2022-2023_Vizsgaremek\dokumentáció.docx#_Toc133664594)

[4. ábra TH1 router HSRP konfigurációja 5](file:///C:\GitHub\2022-2023_Vizsgaremek\dokumentáció.docx#_Toc133664595)

[5. ábra HSRP felépítése 5](file:///C:\GitHub\2022-2023_Vizsgaremek\dokumentáció.docx#_Toc133664596)

[6. ábra feszítőfa információi 6](file:///C:\GitHub\2022-2023_Vizsgaremek\dokumentáció.docx#_Toc133664597)

[7. ábra EtherChannel jelenléte a topológiában 6](file:///C:\GitHub\2022-2023_Vizsgaremek\dokumentáció.docx#_Toc133664598)

[8. ábra EtherChannel konfigurációja 6](file:///C:\GitHub\2022-2023_Vizsgaremek\dokumentáció.docx#_Toc133664599)