IoT Hálózati Forgalomanalizáló és Biztonsági Auditáló Szoftver

1. Bevezetés és Célkitűzés

Az Internet of Things (IoT), azaz a Dolgok Internetének terjedésével egyre több eszköz csatlakozik az otthoni és vállalati hálózatokhoz. Kamerák, okosizzók, routerek és termosztátok kommunikálnak folyamatosan, gyakran megfelelő biztonsági védelem nélkül.

Jelen projekt célja egy **Java alapú hálózati diagnosztikai eszköz** fejlesztése volt, amely képes:

1. Felderíteni egy adott hálózati eszközt IP cím alapján.
2. Ellenőrizni az eszköz elérhetőségét.
3. Portszkenneléssel (Port Scanning) feltérképezni a futó szolgáltatásokat.
4. Biztonsági kockázatelemzést végezni a nyitott portok alapján.

A projekt költséghatékony megvalósítása érdekében a drága célhardvereket egy **Android okostelefonnal szimuláltuk**, amelyen egy webszerver futtatásával modelleztük egy sebezhető IoT eszköz viselkedését.

2. Alkalmazott Technológiák és Módszertan

2.1 Szoftveres környezet

A fejlesztés **Java** nyelven történt, mivel a nyelv beépített java.net csomagja kiváló támogatást nyújt az alacsony szintű hálózati műveletekhez (Socket programozás).

A program működése négy fő fázisra bontható:

1. **ICMP/Ping vizsgálat:** Az InetAddress.isReachable() metódussal ellenőrizzük, hogy a célpont válaszol-e.
2. **TCP Connect Scan:** A program megpróbál TCP kapcsolatot létesíteni (3-way handshake) egy előre definiált portlistán.
3. **Timeout kezelés:** A gyors működés érdekében 200 ms-os időkorlátot (timeout) vezettünk be, így a zárt portok nem lassítják le a futást.
4. **Logikai kiértékelés:** A nyitott portok alapján a szoftver következtetéseket von le (pl. 8080-as port esetén webszerver jelenléte).

2.2 Hardveres szimuláció (IoT Eszköz)

Valós hardver hiányában egy Android okostelefont használtunk a célpont szerepében.

- **Eszköz:** Okostelefon Wi-Fi kapcsolattal.
- **Szoftver:** *Simple HTTP Server* (mobil alkalmazás).
- **Konfiguráció:** A telefon a **8080-as porton** indított egy titkosítatlan HTTP szolgáltatást, amely tökéletesen szimulálja egy rosszul beállított router vagy IP kamera adminisztrációs felületét.

3. A Megvalósítás Részletei

A program "agyát" a Socket osztály alkotja. A szkennelés nem "brute-force" módon 1-től 65535-ig történik, hanem egy célzott listát vizsgál, amely a leggyakoribb IoT sebezhetőségeket tartalmazza:

- **Port 21, 22:** FTP és SSH (fájlátvitel és távoli vezérlés).
- **Port 23:** Telnet (elavult, titkosítatlan protokoll – magas kockázat).
- **Port 80, 8080:** HTTP webszerverek (gyakori admin felületek).
- **Port 554:** RTSP (Real Time Streaming Protocol – kamerákhoz).

A program a felhasználótól bekéri az IP címet, majd szekvenciálisan végrehajtja a vizsgálatokat, és a konzolon strukturált formában megjeleníti az eredményt.

4. Tesztelés és Eredmények

A tesztelést egy izolált helyi Wi-Fi hálózaton (LAN) végeztük.

Tesztkörnyezet adatai:

- **PC IP címe:** (A futtató gép)
- **Célpont (IoT eszköz) IP címe:** pl 192.168.0.105 (Példa cím)
- **Megnyitott szolgáltatás:** Simple HTTP Server a 8080-as porton.

Futtatási napló (Log):

Plaintext

```
==========================================
    IoT Forgalomanalizáló & Portszkenner
==========================================
Add meg a célpont IP címét: pl 192.168.0.105

[1] Eszköz elérhetőségének ellenőrzése...
✓ Az eszköz elérhető a hálózaton.

[2] Kritikus portok vizsgálata...
.....

Találat! Nyitott portok:
 -> [NYITVA] Port 8080

[3] Biztonsági értékelés és jelentés...
 [INFO] Port 8080/8000: Alternatív webszerver.
      -> Ez valószínűleg a tesztelésre használt telefonod webszervere.
      -> Éles környezetben ez gyakran routerek admin felülete.

==========================================
```

Elemzés befejezve.

Eredmények kiértékelése: A teszt sikeres volt. A szoftver:

1. Helyesen azonosította, hogy a pl 192.168.0.105-ös eszköz aktív.
2. Megtalálta a 8080-as nyitott kaput, miközben a többi zárt portot (pl. 23, 554) figyelmen kívül hagyta.
3. Helyes biztonsági figyelmeztetést adott, jelezve, hogy egy webszerver fut az eszközön.

Vizuális ellenőrzésként a PC böngészőjében is megnyitottuk a pl http://192.168.0.105:8080 címet, ahol az oldal sikeresen betöltődött, igazolva a Java program találatát.

5. Biztonsági Következtetések

A projekt rávilágított az IoT eszközök alapvető sebezhetőségére. Ha egy támadó (vagy a mi szoftverünk) képes másodpercek alatt felderíteni egy nyitott portot a helyi hálózaton, akkor könnyen hozzáférhet az eszköz adminisztrációs felületéhez, ha az nincs jelszóval védve.

Javaslatok a védekezésre:

- Ismeretlen portok lezárása a tűzfalon.
- WPA2/WPA3 titkosítás használata a Wi-Fi hálózaton.
- Alapértelmezett portok (pl. 8080) megváltoztatása.
- Erős jelszavak használata az admin felületeken.