

Angry IP Scanner - Projektterv

Tartalom

1. Bemutató	2
2. A jelenlegi program szerkezete	2
3. A program fő funkciói	6
3.1 Use-Case	6
3.2 Jelenlegi funkciók bemutatása	6
Főképernyő	6
Fetchers	7
Preferences	8
Hálózat felderítése	10
Statisztika	11
Exportálás	11
4. A projekt átvétele	12
5. A tervezett változtatások	12
5.1 CSV szeparátor karakter	12
5.2 Minimize to tray	12
5.3 Mentés a hálózat felderítés alatt is	12
5.4 Port szkennelés progress bar	13
5.5 Szkennelés mentése, betöltése	13
5.6 IP címek fordított sorrendben	13
5.7 Lokalizáció	13

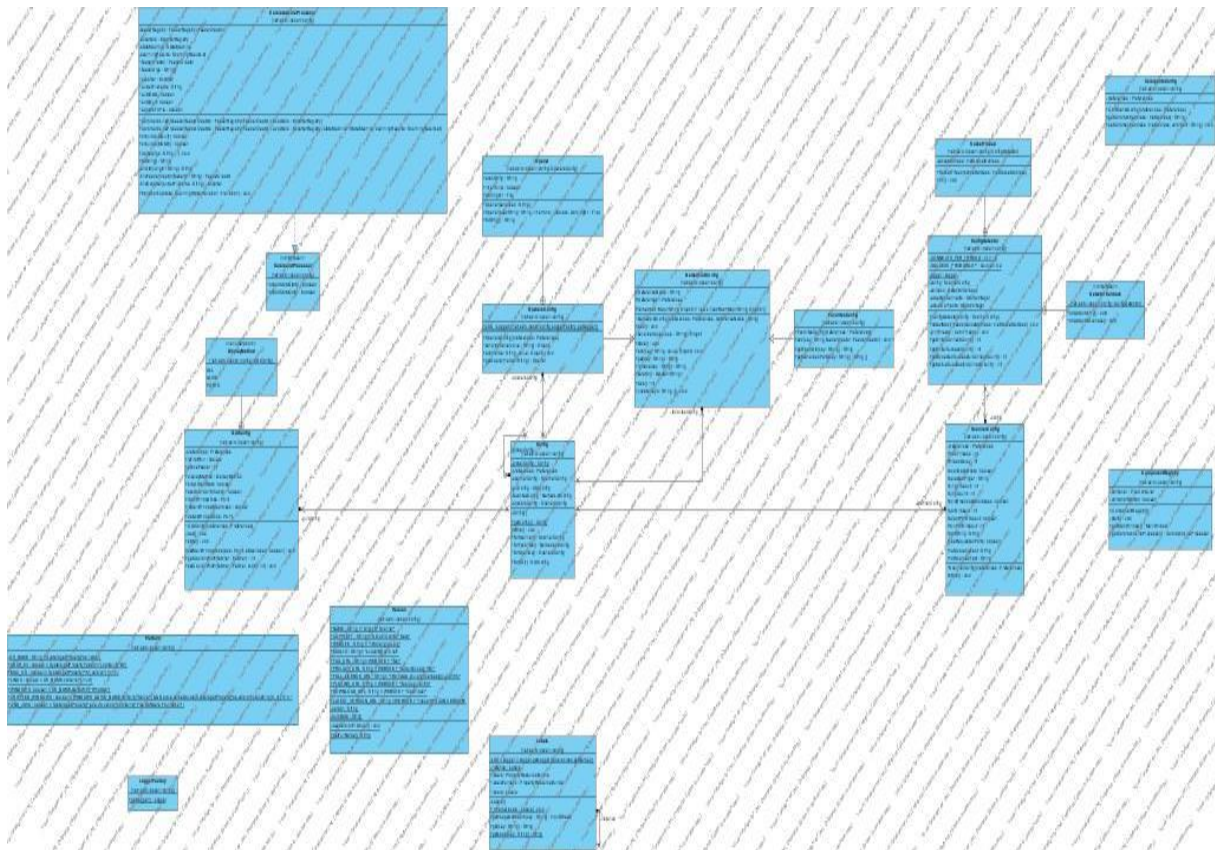
1. Bemutató

Az Angry IP Scanner egy nyílt forráskódú, *hálózat felderítő* eszköz. Képes megvizsgálni, hogy a megadott IP cím tartományban a hálózaton mely IP címek vannak használatban, meghatározni az eszközök MAC címét, webszervert felismerni és a NetBIOS információkat megadni. Az eredményeket több formátumban is lehetséges elmenteni.

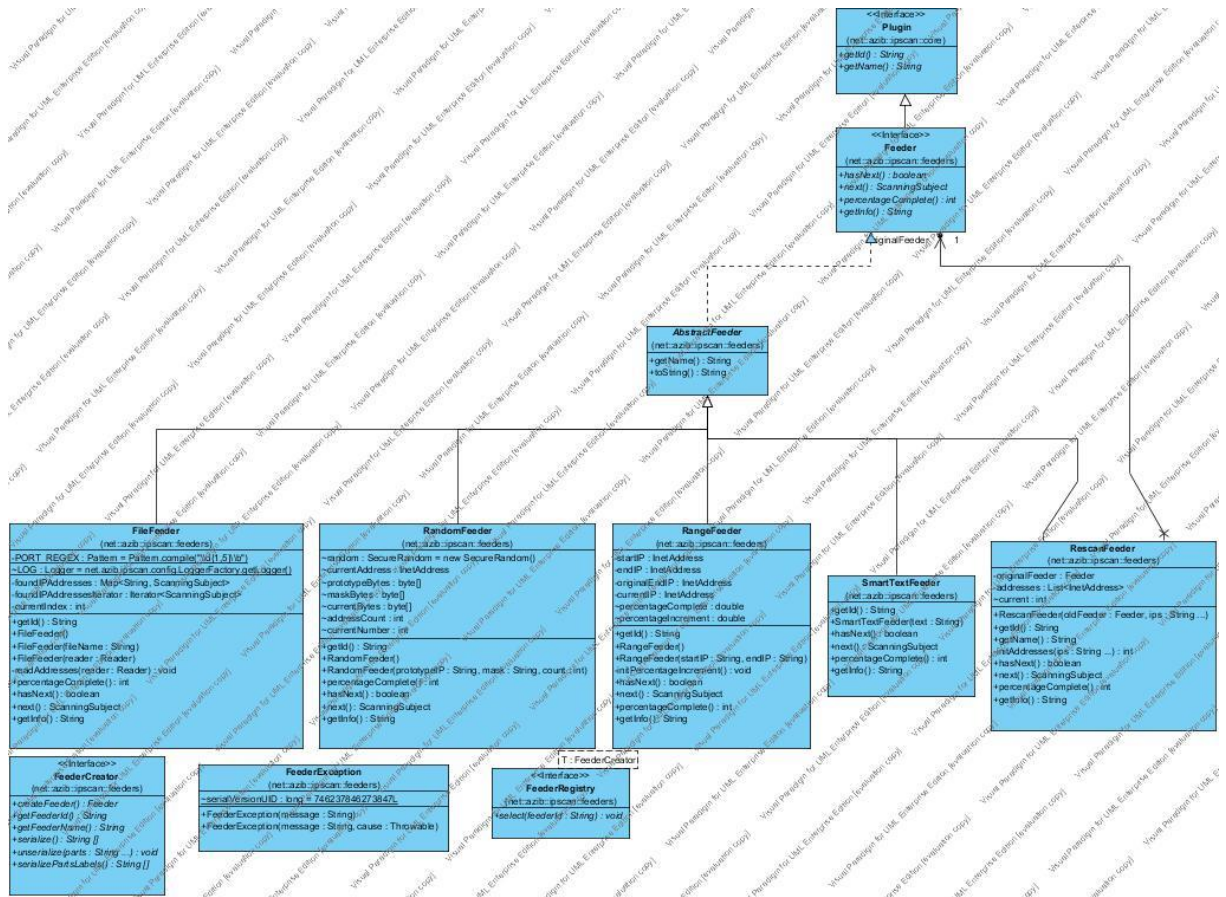
A projekt célja az eszköz bővítése néhány, az Open Source közösség által kért, hasznosnak vélt funkcióval.

2. A jelenlegi program szerkezete

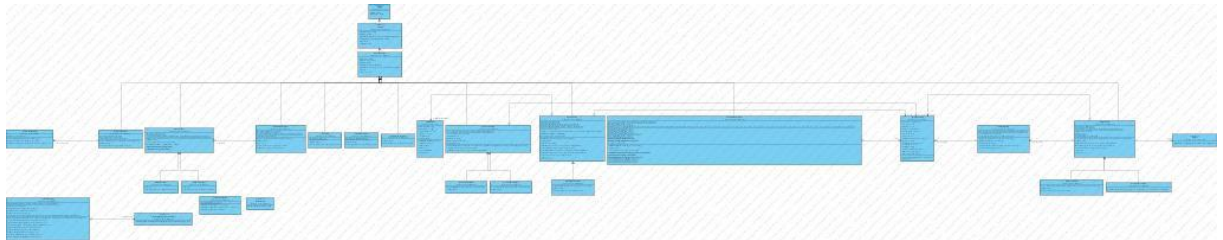
A program szerkezetéről class illetve package diagram készült. A class diagrammok packagenként jelenítik meg az osztályokat és azok kapcsolatait a jobb átláthatóság miatt. A programról EK diagram nem készült mivel nem tartozik hozzá adatbázis.



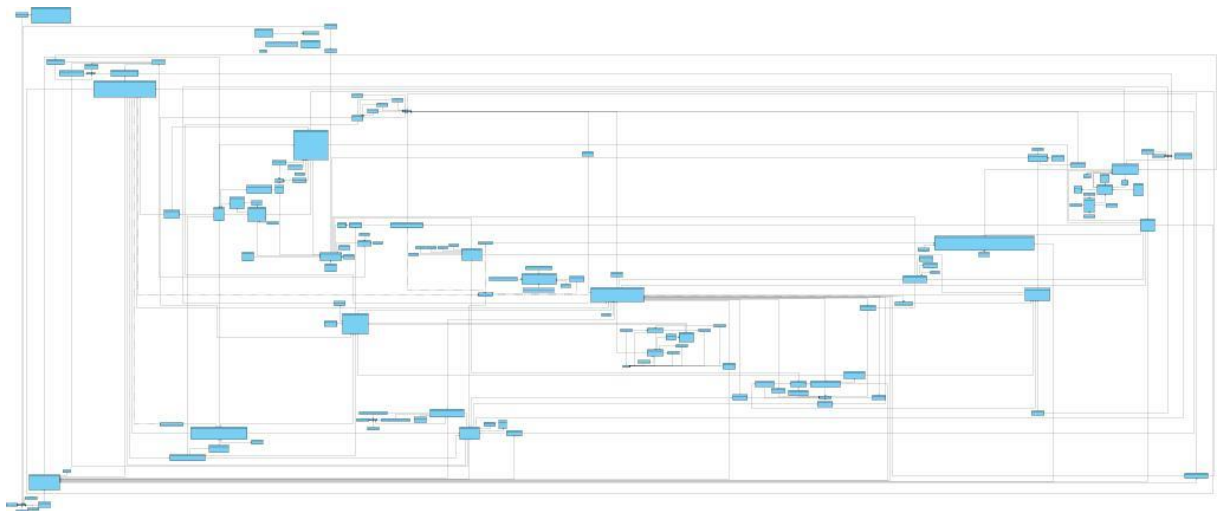
1. ábra A net.azib.ipscan.config package-hoz tartozó class diagram



4. ábra A net.azib.ipscan.feeders packagehoz tartozó class diagram



5. ábra A net.azib.ipscan.fetchers packagehoz tartozó class diagram

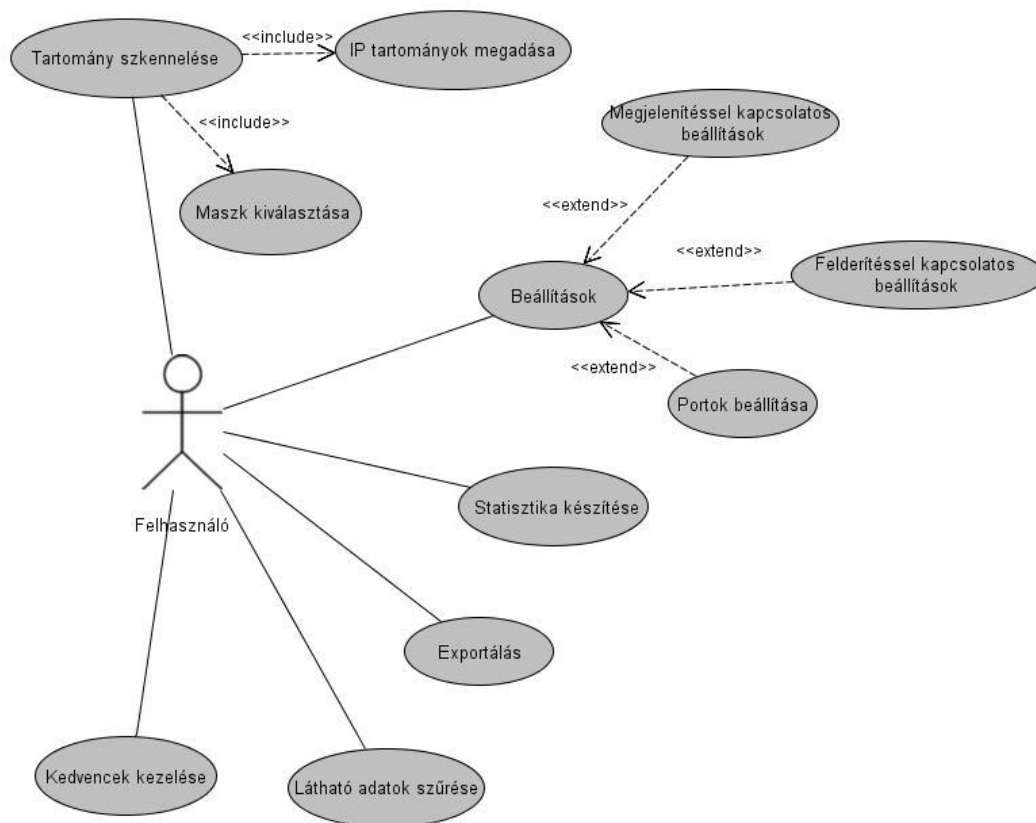


6. ábra A net.azib.ipscan.gui packagehoz tartozó class diagram

3. A program fő funkciói

A program fő funkciói usecase diagramon illetve a programról készült screenshotokkal is bemutatásra kerülnek.

3.1 Use-Case



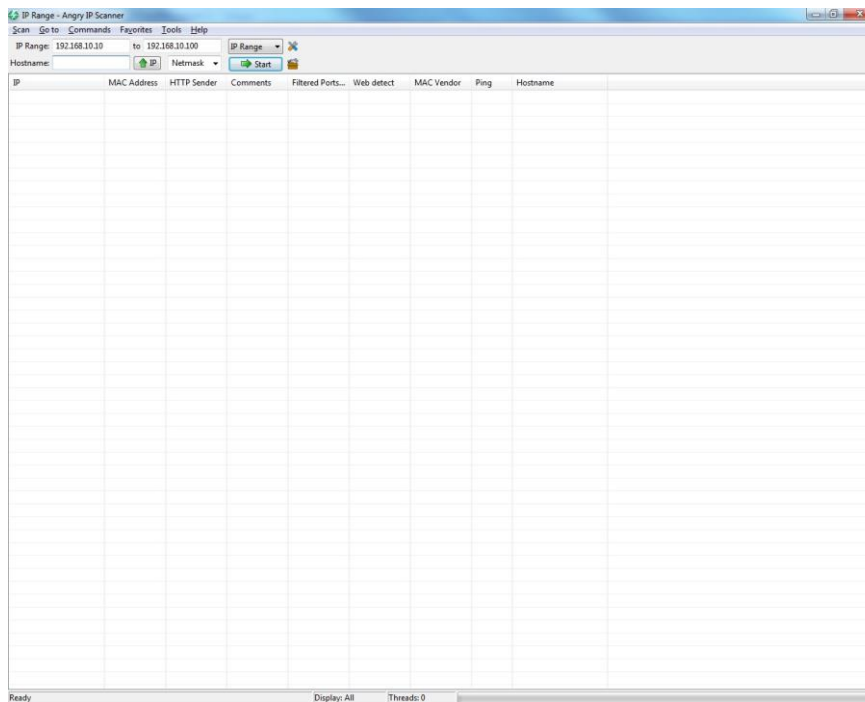
9. ábra Usecase diagram

3.2 Jelenlegi funkciók bemutatása

Főképernyő

A program indítása után az alábbi képernyő fogad. A szükséges adatok megadása után akár rögtön indítható a hálózat felderítése. Látható egy táblázat melyben a szkennelés eredménye lesz megtekinthető, illetve a menüből elérhetőek a program további funkciói is.

A Start gomb melletti ikonral a beállítási lehetőségek jelennek meg, a felette lévő ikonra kattintva pedig kiválasztható, hogy mely adatok szerepljenek a táblázatban. Az IP gombra kattintva az adott gép IP címét jeleníti meg kezdő és vég IP címként. A táblázat fejlécében szereplő adatok szerint rendezni lehet az eredmény listát.



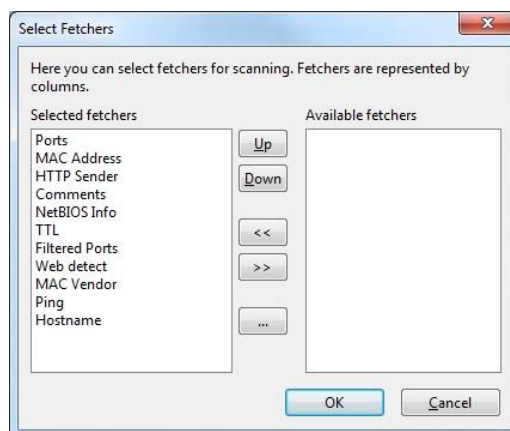
10. ábra Főképernyő

Fetchers

A Tools->Fetchers menüpont alatt a következő ablak jelenik meg. Itt lehetséges beállítani azt, hogy milyen adatok jelenjenek meg a felderített hálózatról a főképernyőn található táblázatban.

A program által felderíthető lényegesebb adatok:

- Nyitott portok egy adott eszközön
- Az eszköz MAC címe
- Az eszköz NetBIOS információi
- MAC cím alapján az eszköz gyártójának beazonosítása
- A pingelés időtartama



11. ábra Megjelenített adatok kiválasztása

Preferences

A Tools->Preferences menüpont esetén a következő ablak jelenik meg. Itt több dolgot is megadhatunk.

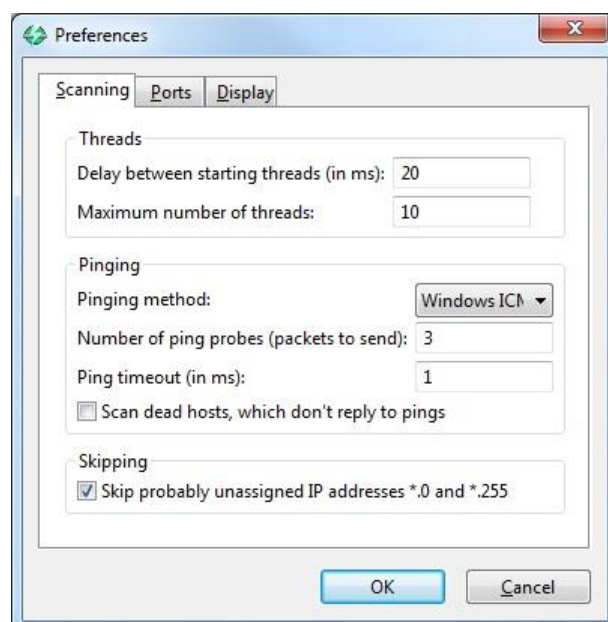
Az első fülön beállítható néhány, a hálózat felderítésével kapcsolatos opció. Megadható, hogy a felderítés hány szálon történjen. Egy ping csomagra viszonylag sok időt kell várni, eközben a saját számítógépünk gyakorlatilag tétlen (a program szempontjából), így érdemes több szálon használni ezt a funkciót. Emellett megadható, hogy az egyes szálak egymáshoz képest milyen időközzel induljanak el.

A pingelési módszer is megadható, továbbá, hogy egy adott cím felé hány darab ping kérést szeretnénk küldeni. Ha gyorsan szeretnénk felderíteni a hálózatot, elég, ha minden cím felé csupán egyetlen kérést próbálunk meg küldeni. Nagyobb, vagy esetlegesen nagyon leterhelt hálózat esetében azonban előfordulhat, hogy 1-2 ping csomag „elveszik”, későn érkezik meg. Ekkor érdemes lehet többet is beállítani.

Megadható továbbá, hogy egy ping kérésre hány ms ideig szeretnénk várakozni. Minél kevesebb ideig várakozik a program, annál gyorsabb a felderítés, de terhelt hálózat, vagy eszköz esetében nagyobb válaszidőt érdemes hagynunk.

Beállítható, hogy „halott” (pingre nem válaszoló) eszközöket esetlegesen részletesebben is vizsgáljon meg a program (nyitott portok szempontjából). Előfordulhat, hogy egy eszköz a tűzfal beállításai miatt pingre nem válaszol, ettől függetlenül még létezhet a hálózaton.

A nagy valószínűséggel nem használható (0-ra, vagy 255-re végződő) IP címekre beállíthatjuk, hogy kihagyja-e őket a program. Ezt akkor érdemes leginkább használni, ha C típusú maszkkal rendelkező hálózaton szeretnénk pingelni, egyéb esetben elveszíthetünk néhány létező eszközt (mert nem pingeli őket a program).



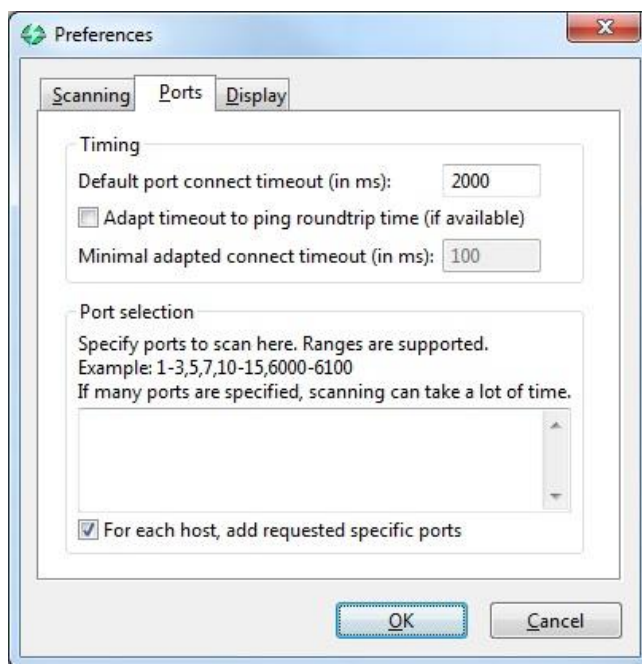
12. ábra Szkeneléshez kapcsolódó beállítások

A Ports fül alatt a hálózat felderítése során az egyes eszközökön nyitott portokat is kereshetünk.

Beállítható, hogy egy port válaszára hány ms-ot várjon a program. Ez is terheltségtől függő lehet, de itt inkább az adott eszköz terheltsége befolyásolja a válaszidőt. Egy nagyobb funkciót kielégítő porton lassabban érkezhetsz válasz, ezért ez alapértelmezetten 2000 ms-ra van állítva. (Ez azonban borzasztóan sok, ha nagy mennyiségű portot szeretnénk szkennelni)

Emellett természetesen megadhatjuk, hogy pontosan milyen portokat (esetlegesen mely eszköz(ök)ön) szeretnénk tesztelni. Minél többet adunk meg, a hálózat felderítése annál tovább fog tartani.

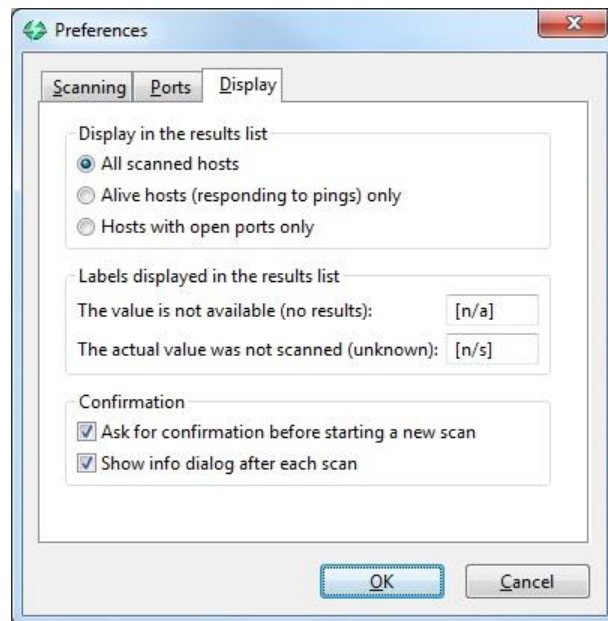
Az egyes portok válaszüzeje függhet az eredetileg már megállapított ping válaszüzejétől (feltéve, hogy a hálózat terheltségétől függ a port válaszüzeje is). Ha így van, beállítható, hogy a portok szkennelése során a várakozási idő alkalmazkodjon a ping válaszüzejéhez, így dinamikusabban kezelhető.



13. ábra Portok szkenneléséhez kapcsolódó beállítások

A Display fülön megjelenítéssel kapcsolatos beállításokat végezhetünk. Kiválaszthatjuk, hogy mely eszközök jelenjenek meg a találati listán (összes, élő, nyitott portal rendelkező).

Beállíthatjuk, hogy milyen jellegű címkék kerüljenek a nem elérhető, vagy nem szkennelt mezőkhöz. Továbbá a program konfirmációs dialógus ablakait is minimálisan személyre szabhatjuk.



14. ábra Eredmények megjelenítéséhez kapcsolódó beállítások

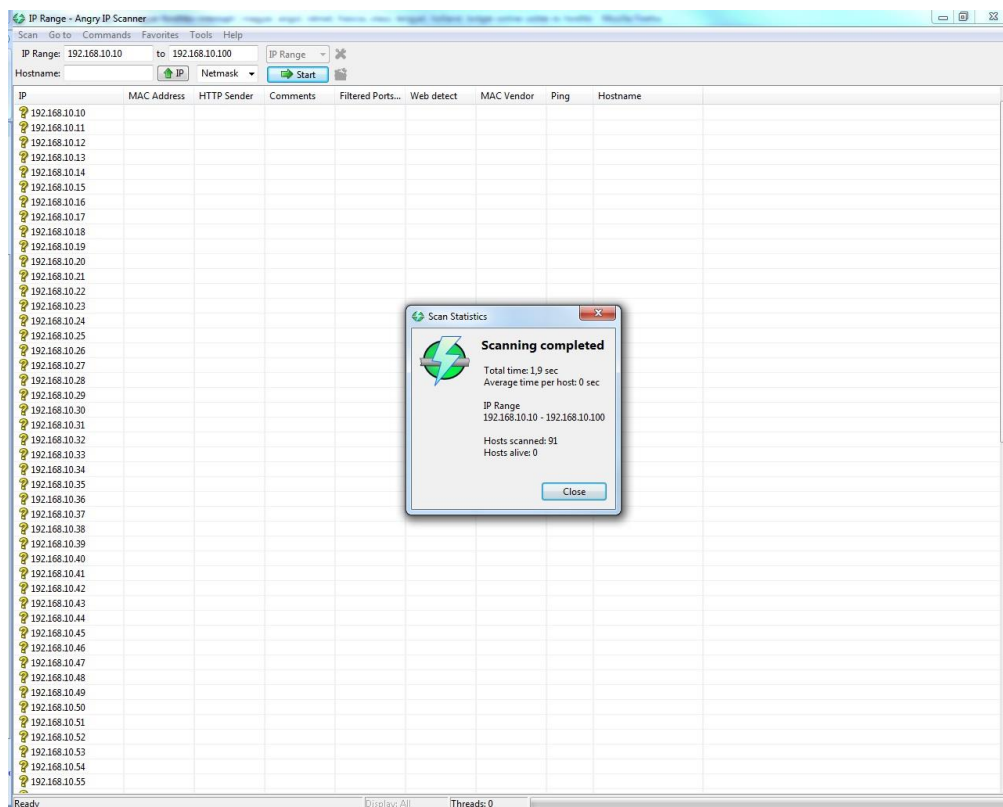
Hálózat felderítése

Miután elvégeztük a kívánt beállításokat, következhet a tényleges eszközfelderítés folyamata. A főképernyőn megadhatjuk, milyen IP cím tartományt szeretnénk végigszkennelni a programmal. Egy kezdő és egy vég IP címet kell megadnunk, a program pedig automatikusan végig fogja iterálni ezeket.

Természetesen az egyes hálózati címek egyedi alhálózati maszkkal is rendelkezhetnek. A program már fel van készítve ezekre is. Megadhatunk úgynevezett osztályos (A, B, C) típusú maszkokat, de akár egyénieket is.

A szkennelés során a főképernyő táblázata folyamatosan frissül. Mindig megjelennek az aktuálisan szkennelt IP címek, valamint ezek állapotai. Ha egy cím nem elérhető, sárga kérdőjel jelenik meg mellette. Ha elérhető, zöld pipát láthatunk. Az elérhető címek mellett megjelennek a korábban Fetchers menüpontban beállított tulajdonságai is az adott eszköznek.

A felderítés végeztével megtekinthetők az egyes eredmények, valamint egy statisztikát mutató ablak is. Az elérhetőnek talált eszközökre duplán kattintva láthatjuk a fontosabb információkat.

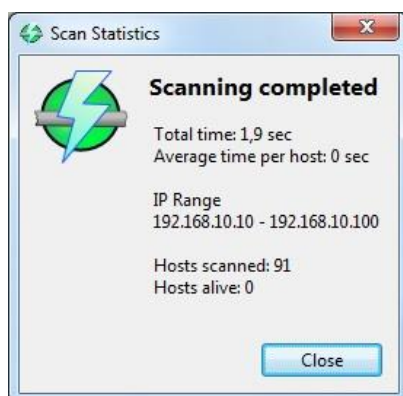


15. ábra Felderített hálózat

Statisztika

Az utoljára elvégzett hálózat felderítésről az adatok elérhetőek a Tools->Scan Statistics menüpont alatt.

Megtekinthető a szkennelés teljes időtartama, mennyi időnként talált meg átlagosan egy eszközt a program, a szkennelt IP tartomány, az összes szkennelt hoszt, és az elérhetőnek talált eszközök száma.



16. ábra Utolsó felderítés statisztikája

Exportálás

A Scan menüpont alatt elérhető egy exportálás funkció is. A legutoljára elvégzett keresés eredményei exportálhatóak különböző típusú fájlokba (pl. CSV).

4. A projekt átvétele

A projekt átvétele során hiányosságokat szerencsére nem tapasztaltunk. A projektet egy fő fejlesztő, Anton Keks készítette. Az egy fős projektek sajnos sok esetben rossz kódminőséggel rendelkeznek, könnyen átláthatatlanná válnak.

Az Angry IP Scanner projektben találhatóak ugyan PMD szabálysértések, a projekt teljes szerkezetét nézve azonban jól módosíthatónak találtuk a forráskódot.

Jelenleg GIT verziókövető segítségével folyik a projekt fejlesztése, viszonylag lassú ütemben (az egy fejlesztő miatt). Ebbe szeretnénk besegíteni, ha minden jól megy, egy saját branch-be is feltölthetjük a változtatásainkat Anton Keks GIT repository-jába.

A program mindemellett jól működik. Java nyelven íródott, és az eltérő hálózatkezelések ellenére (több külső lib segítségével) működik Windows és Linux 32 és 64 bites operációs rendszereken, valamint MAC OS-en is.

Kis nehézséget jelentett a projekt első elindítása (a külső libektől való függőségek miatt), de miután ez megoldódott, úgy gondoljuk, hatékonyan részt vehetünk a fejlesztésében.

A projekt elérhetősége: <https://github.com/angryziber/ipscan>

5. A tervezett változtatások

A projekthez néhány, közösség által igényelt fejlesztést szeretnénk megvalósítani.

5.1 CSV szeparátor karakter

<http://sourceforge.net/p/ipscan/feature-requests/97/>

Felmerült az igény, hogy az exportált szkennelési statisztika CSV fájljában változtatható legyen a szeparátor karakter. Ez egy alapszintű funkció, melyet grafikus és parancssori alkalmazás során is elérhetővé szeretnénk tenni.

5.2 Minimize to tray

<http://sourceforge.net/p/ipscan/feature-requests/137/>

A program jelenleg legfeljebb csak a tálcára rakható le. Hosszabb kereséseknél, folyamatos futtatásnál ez zavaró lehet, ezért szeretnénk elkészíteni egy „minimize to tray” funkciót, mellyel a tálcáról is elrejtve, az óra mellett jelenhet meg.

5.3 Mentés a hálózat felderítés alatt is

<http://sourceforge.net/p/ipscan/feature-requests/98/>

Parancssoros használat esetén felmerült az igény, hogy a felderítési eredményeket ne csak a felderítés végén, hanem a szkennelés során folyamatosan mentse fájlba a program.

5.4 Port szkennelés progress bar

<http://sourceforge.net/p/ipscan/feature-requests/255/>

Ha sok nyitott portot keresünk, a keresés nagyon sokáig tarthat. A jelenlegi állapotcsík csupán azt mutatja a program futása során, hogy az IP címeken belül hol tart a futás a tartományhoz képest. Emellett szeretnénk még egy progress bart megvalósítani, mely az egyes eszközökön belül a port szkennelés állapotát mutatná.

A funkció a program 3.0-s verziójához lett beírva mérföldkőnek, azonban még mindig nem készült el.

5.5 Szkennelés mentése, betöltése

<http://sourceforge.net/p/ipscan/feature-requests/223/>

Jelenleg a programban lehetőség van több féle formátumban is exportálni a keresés eredményét. Nagy tartomány keresése esetén azonban a szkennelés óráig is eltarthat. Emiatt felmerült az igény, hogy egy megállított szkennelést később folytatni lehessen.

Egy egységes (egyformátumos) mentési lehetőséggel, valamint egy ehhez tartozó betöltés funkcióval is szeretnénk bővíteni emiatt a rendszert.

5.6 IP címek fordított sorrendben

<http://sourceforge.net/p/ipscan/feature-requests/242/>

A jelenlegi működés szerint, ha a szkennelni kívánt tartományban a kezdő IP cím nagyobb, mint a vég IP cím, a program egy hibaüzenetet küld erre, és nem kezdi el a szkennelést. Ehelyett szeretnénk az IP címek iterálását (a közösségi igényekhez igazodva) oly módon megváltoztatni, hogy egy nagyobb kezdő IP cím megadása esetén a szkennelés fordított sorrendben történjen.

5.7 Lokalizáció

A felsorolt igények mellett szeretnénk egy saját ihletésű funkciót is megvalósítani. Egyrészt, egy magyar nyelvi fájlt készíteni a programhoz. Másrészt, egy nyelv beállítási lehetőséget a Preferences-be.

A programban található összes String jelenleg is külön properties fájlban található. Ehhez külön nyelvi fájlokat felvéve több nyelvűvé is tehető a program, melyhez GUI-s beállítási lehetőség is tartozna.