ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

**NEUMANN JÁNOS
INFORMATIKAI KAR**

SZAKDOLGOZAT

DIPLOMAMUNKA

OE-NIK Hallgató neve: **Gyebnár Szabolcs**
2020 Hallgató törzskönyvi száma: **T-000123/FI12904**

Óbudai Egyetem
Neumann János Informatikai Kar
Biomatika és Alkalmazott Mesterséges Intelligencia Intézet

SZAKDOLGOZAT
FELADATLAP

Hallgató neve: Gyebnár Szabolcs

Törzskönyvi száma: NIK-O-NI-02-33

Neptun kódja [redacted]

A dolgozat címe:

Okos otthon forgalmi mintázatának elemzése Wifi hálózati jellemzők monitorozásának segítségével

Traffic pattern analysis of smarthome based on monitoring of WiFi network parameters

Intézményi konzulens: Vörösné Dr. Bánáti-Baumann Anna

Külső konzulens: Varga Krisztián

Beadási határidő: 2023. december 15.

A záróvizsga tárgyai: Informatikai rendszerek üzemeltetésének biztonsága
IT hálózatbiztonság

A feladat

Az IoT hálózatok túl gyors technológiai fejlődésével sajnos a kibervédelemi módszerek és megoldások nem tudtak lépést tartani, pedig az otthoni megjelenésüktől (okos otthon) egészen a városi és ipari felhasználásukig (okos város) széles körben elterjedtek világszerte. Az IoT hálózatokat alkotó eszközök a mindennapi életet teszik hatékonyabbá, időtakarékosabbá és kényelmesebbé, miközben a magánszféránk szerves részéhez kapcsolódó információkhoz férnek hozzá, így a különböző eszközök közötti kommunikáció biztonsága elengedhetetlenül fontos.
A szakdolgozat célja több különböző gyártótól származó eszközből álló okosotthon megoldás megtervezése és kialakítása annak érdekében, hogy a forgalom monitorozásával és elemzésével megfelelő biztonsági megoldások kidolgozása váljon lehetővé. További cél egy 'privát felhő' kialakítása, melyet csak hitelesített, azonosított és feljogosított felhasználók érhetnek el egy titkosított csatornán keresztül.
A hallgató feladata az okosotthon hálózatának megtervezése és megvalósítása, a különböző monitorozó és logelemző módszerek és eszközök feltérképezése, a forgalom vizsgálata, valamint a privát felhővel egy biztonságos kommunikációs csatorna kiépítése és tesztelése.

A dolgozatnak tartalmaznia kell:

- IoT eszközök, szolgáltatások bemutatását,
- a rendszer implementációját,
- sérülékenységeinek bemutatását és elemzését,
- a hálózati kommunikáció központi monitorozását és elemzését,
- a megvalósítás során szerzett tapasztalatok összefoglalását és kiértékelését,
- a továbbfejlesztési lehetőségek vizsgálatát.

ÓBUDAI EGYETEM Neumann János Informatikai Kar 1. Budapest

.....
Dr. Eigner György
mb. intézetigazgató

A szakdolgozat elévülésének határideje: **2023. december 15.**
(OE TVSz 55.§ szerint)

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens

HALLGATÓI NYILATKOZAT

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2022. május 15.

hallgató aláírása

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

KONZULTÁCIÓS NAPLÓ

Hallgató neve:	Neptun Kód:	Tagozat:
Gyebnár Szabolcs	[redacted]	levelező

Telefon: Levelezési cím (pl.: lakcím):

[redacted]

Projektmunka címe magyarul:

Okos otthon forgalmi mintázatának elemzése Wifi hálózati jellemzők monitorozásának segítségével

Projektmunka címe angolul:

Traffic pattern analysis of smarthome based on monitoring of Wifi network parameters

Intézményi konzulens: Külső konzulens:

Vörösné Dr. Bánáti-Baumann Anna.....Varga Krisztián

Kérjük, hogy az adatokat nyomtatott nagy betűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.02.16.	Féléves munka átbeszélése	
2.	2022.03.25.	Mérési pontok definiálása	
3.	2022.04.20.	Adatbázis implementálás egyeztetés	
4.	2022.05.05.	Prezentáció bemutatás, véleményezés	

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Projektlabor 1. / 2. / 3. / Záródolgozati projekt[1] tantárgy követelményét teljesítette, beszámolóra/védésre[2] bocsátható. A javasolt érdemjegy:

.....

Intézményi konzulens

Budapest, 2022.05.06.

[1] A megfelelő bekarikázandó/aláhúzandó!

[2] A megfelelő aláhúzandó!

1. Tartalomjegyzék

2. Rövid tartalmi összefoglaló a téma területéről, a feladatról

IoT. Internet of Things. azaz a dolgok internetje, ahogy az éppen olvasható a magyar Wikipédián. Ez a fordítás némiképp nyers és kissé magyartalannak hangzik elsőre, de benne van minden, amit ezen három szócskának jelentenie kell.

Manapság szinte mindenbe építenek valamely kommunikációs technológián alapuló chip-et, legyen az mosógép, villanykapcsoló, vagy akár az autónk. Ezek a chip-ek lehetővé teszik, hogy ’beszélgetni’ tudjunk az eszközeinkkel: a sütő értesítést küld, ha elkészült a Wellington bélszín; a mosógép, ha végzett a programmal; a videós kapucsengő, ha megjött a postás, avagy ne adj’ Isten, a füstérzékelő, ha ott felejtettük a bundáskenyeret.

Ez a kommunikáció természetesen kétirányú, azaz beállíthatjuk a füstérzékelőt, hogy most 23 percig ne jelezzen, mert épp olyan étel készül, avagy a következő 30 percben senki ne zavarjon, így a kapucsengő sem értesít bennünket semmilyen eseményről. Már ezek a funkciók is érezhetően jobbá, kényelmesebbé tehetik az életünket. Az igazi varázslat ezzel együtt akkor jön el, amikor az eszközeink egy közös hálózatba kapcsolva – akár egymást utasítva is – képesek egymással kommunikálni.

Például, ha a telefonunk felcsatlakozott a saját Wifi Access Pointunkra és a garázsajtó vagy a bejárati kapu zárt állapotból 3 perc alatt felvette a nyitott, majd újra a zárt állapotot, ugyanakkor a bejárati ajtó zárja zárt állapotban van, akkor a bejárati ajtó zárja váltson nyitott státuszra. Ezáltal zavartalanul jutunk be az otthonunkba, anélkül, hogy elő kellene keresnünk a kulcsot, hiszen a rendszer azonosította személyünket, illetve egyúttal felismerte, hogy éppen hazaértünk. Az ilyen jellegű automatizmusoknak a képzelet – és persze a pénztárcánk – szabhat csak határt.

Az IoT hálózatok túlzottan gyors technológiai fejlődésével sajnos a kibervédelemi módszerek és megoldások nem, vagy csak nagyon nehezen tudnak lépést tartani, pedig az otthoni megjelenésüktől (okos otthon) egészen a városi és ipari felhasználásukig (okos város) széles körben elterjedtek világszerte. Az IoT hálózatokat alkotó eszközök a mindennapi életet teszik hatékonyabbá, időtakarékosabbá és kényelmesebbé, miközben a magánszféránk szerves részéhez kapcsolódó információkhoz férnek hozzá, így a különböző eszközök közötti kommunikáció biztonsága elengedhetetlenül fontos.

A szakdolgozatom célja egy 15-20, különböző gyártótól származó eszközből álló okosotthon megoldás megtervezése és kialakítása annak érdekében, hogy a forgalom monitorozásával és elemzésével megfelelő biztonsági megoldások kidolgozása váljon lehetővé. További cél egy ’privát felhő’ kialakítása, melyet csak hitelesített, azonosított és feljogosított felhasználók érhetnek el egy titkosított csatornán keresztül.

Dolgozatomban az okosotthon megoldásokban használt különböző technológiák, azok előnyei és hátrányai kerülnek bemutatásra konkrét eszközökön keresztül. Megkísérlek létrehozni egy központi otthon automatizálási rendszert, amelybe minden okos, IoT eszköz és szolgáltatás beintegrálásra kerül. Ez lesz tehát a Központ, itt valósul meg minden automatizmus, vezérlés és logika.

Miután a rendszer felépült, a következő lépés a fenyegetettség és sérülékenység vizsgálat elvégzése, amelyet főként hálózati jellemzők monitorozásának és elemzésének segítségével végzek majd el.

Az Internet világában nincs áthatolhatatlan fal, vagy megmászhatatlan kerítés, de törekednünk kell a biztonságra és a biztonságosságra. Ne kínáljuk tehát tálcán a privát szféránkat, a személyes adatainkat, azaz ne adjunk ingyen használható erőforrást a rosszfiúk kezébe!

„IoT without security = Internet of Threats”

-Stephane Nappo-

3. A megoldandó probléma megfogalmazása, fontossága, felvezetése

Mindig is érdekeltek az IT világ új trendjei, eszközei és szolgáltatásai. Hiszem, hogy a technológia értünk van, azért, hogy az életünket jobbá, kényelmesebbé, egyszerűbbé és biztonságosabbá tegye. Ezért is döntöttem úgy 2020-ban, hogy futólépést tartva a korral felokosítom az otthonom. Ahogy elkezdtem beleásni magam a témába, két dolog rögtön világossá vált: a központosítás és a biztonság.

Sokunkat már az is őrületbe kerget, hogy nem lehet úgy bemenni egy élelmiszerboltba, hogy annak ne legyen valamilyen pontgyűjtő vagy törzsvásárlói kártyája, amivel xy termék olcsóbb, netalán a vásárlásaink után pontokat gyűjthetünk, amit később boldogan használhatunk fel újabb akciók keretében. Szummárum: mindig kell nálunk legyen 19 db plasztik kártya.

Ez a gondolatmenet vezetett egy központosított rendszer megtervezéséig a 19 féle termék 19 féle alkalmazásának implementálását és tizenkilencesével történő irányítását elkerülendő.

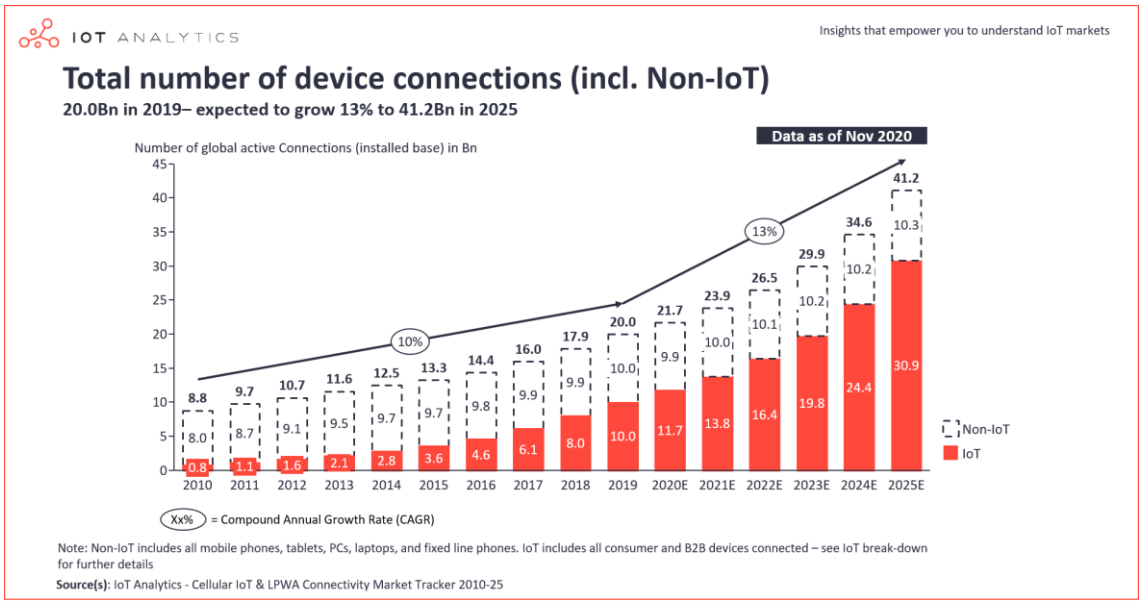
A másik a biztonság kérdése és problematikája, hiszen ezek az eszközök, kapcsolók, szenzorok, kamerák stb. érzékeny adatokat állítanak elő vagy azokkal állnak kapcsolatban. Ezért kiemelt figyelmet kell fordítanom arra, hogy a kialakított rendszer minden elemét, a kommunikációs csatornákat és az azon közlekedő adatokat a lehető legbiztonságosabb környezet fogja közre. Ugyanakkor mindez nem jelenthet funkcióvesztést, azaz emiatt nem cél lemondani a távoli elérés, vezérlés, monitoring, avagy egyéb hasznos tulajdonság használatának a lehetőségéről.

Manapság már szinte minden háztartásban elérhető valamely piaci szolgáltató által biztosított internet elérés. A szolgáltató pár ezer forintért letesz egy routert, kábel modemet vagy valamilyen hálózati eszközt, melyen keresztül az otthonunkból elérhetjük a világhálót. Ezt megtehetjük fizikailag csatlakozva ehhez a hálózati eszközhöz vagy ún. vezeték nélküli technológia segítségével (Wifi).

Az utóbbi évtizedre elmondható, hogy minden család rendelkezik több mobil telefonnal, illetve egy vagy több személyi és/vagy hordozható számítógéppel. Digitális világban élünk, gyakorlatilag ezen eszközök használata elengedhetetlen egy mai modern háztartásban.

Egy 2019. évi amerikai kutatás szerint egy átlagos amerikai otthonban 11db internet elérésű eszköz található. Ez a szám már évek óta stagnál, amely arra enged következtetni, hogy ezen eszközök terén elértük a befogadható mennyiséget. Ezen 11-be azonban nem számítanak bele az IoT eszközök!

Az IoT eszközök számossága rohamosan növekszik a háztartásokban. Ezek az eszközök egyre olcsóbbá, megfizethetőbbé és megfizethetővé válnak számunkra. 2020 volt az első olyan év, amikor az IoT csatlakoztatott eszközök száma világszinten felülmúlta a nem IoT jellegűeket, úgy, mint az okostelefonok, laptopok vagy PC-k. Ez a tendencia az előrejelzések szerint tovább emelkedik, mivel az IoT-k egyre népszerűbbé, illetve elérhetővé válnak az átlagember számára.



3.1. ábra: IoT eszközök számának várható növekedése

4. Lehetséges megközelítési módok, megoldások áttekintése és elemzése

Először is nézzük át a lehetséges technológiákat, eszközöket, protokollokat és szabványokat! A legkézenfekvőbb kommunikációs lehetőség a vezeték nélküli protokollok használata. Kezdjük tehát a wifi-vel!

4.1 Wifi

A wifi egy olyan vezeték nélküli technológia, mely rádióhullámok segítségével építi ki a kapcsolatot és folytatja le a kommunikációt a router és a wifi chippel ellátott eszköz között. E kommunikációnak a minőségét nagyon sok állandó és változó paraméter befolyásolja. A rádióhullámú jeleket eltéríthetik környezeti akadályok, zavarhatják más hasonló jelek, ill. nagyban meghatározza a wifi generációja, szabványa is.

A wifi története nagyon régre nyúlik vissza, bőven a számítógépek megjelenése előtti időkig kell visszatekintsünk. 1942-ben szabadalmaztatta George Antheil zeneszerző és Hedy Lamarr színésznő azt a szórt spektrum alapú technikát, mely megalapozta a ma ismert wifi működését. Mint oly sok minden az IT-ban, így ez is először kizárólag a honvédelemben terjedt el, ez a hálózati technológia csak sokkal később, a 80-as években vált elérhetővé a lakosság számára.

Viszont 1999 az az év, ami ezt a kényelmes és egyszerű kommunikációt mindenki számára elhozta. Ekkor alakult meg a Wifi Alliance, több, akkoriban nagynevű kommunikációs cég hálózataként. Kezdetben még meg sem közelítette a vezetékes kommunikáció sebességét, de mindez az elmúlt több, mint 20 évben megváltozott: jelenleg a 6. generációnál tartunk. A szabványok egymással lefelé kompatibilisek, azaz egy wifi 6-os router képes kommunikálni az összes előző generációt támogató chippel rendelkező eszközzel, de ebben az esetben természetesen csak azon szabványspecifikus jellemzőket használhatja a kommunikáció során, melyet az eszköz tud. Gondolok itt pl. egy titkosítási metódusra.

Szabvány neve	**Újkori neve**	**Megjelenés éve**	**Átviteli sebesség (elméleti)**	**Használt frekvencia**
802.11b		1999	11Mbps	5GHz
802.11a		2000	54Mbps	2.4GHz
802.11g		2003	54Mbps	2.4GHz
802.11n	Wifi 4	2009	72-600Mbps	2.4GHz/5GHz
802.11ac	Wifi 5	2013	akár 6.93Gbps	5GHz
802.11ax	Wifi 6	2020	akár 14Gbps	2.4GHz/5GHz/6GHz

4.2. táblázat: Wifi szabványok

Az első 3 szabvánnyal nem foglalkozom mélyrehatóbban, az ezeket támogató eszközök gyakorlatilag már elavultak, kivesztek a háztartásokból.

4.1.1 Wifi 4 (802.11n)

Ez a szabvány hozta el az igazi áttörést a vezeték nélküli kommunikációban. Túlszárnyalta a hosszú ideig egyeduralkodó LAN sebességet (100Mbps) és ezzel vált igazán térhódítóvá a háztartások körében, amely főként a laptopok esetében volt igazán megfigyelhető. Mivel a laptop csak g-s wifi-t tudott, ezért sok helyen a hordozható eszközöket továbbra is vezetékes hálózaton használták az otthonokban. Az n-es szabvány segítségével azonban már túllépték a 100Mbps-t, hovatovább ha a router is képes volt ilyen sebességen kommunikálni, már okafogyottá vált a vezetékhez kötöttség. Ez a sebesség ilyen mértékű növekedése már elegendő volt multimédiás tartalmak online megtekintésére, online játékokra és minden egyéb másra, ahol fontos volt az adatátviteli sebesség.

Fontosabb jellemzői:

- 2.4 és 5Ghz-es frekvencián is működik
- 20MHz mellett szélesebb frekvencia tartományban 40MHz-et is használ
- képes akár 4 adatfolyam egyszerre történő kezelésére -> elvi sebessége 600Mbps
- MIMO bemutatása
- Nagyobb hatótáv, beltéren akár 70m, kültéren akár 250m
- 64QAM
- Kompatibilis az előző szabványokkal

Fentiek közül csak a MIMO-ra térek ki részletesebben. Ez egy ügyes újítás, Multiple Input Multiple Output azaz több bemenet, több kimenet. Itt a router a továbbítandó adatot több darabra szabdalja, majd azokat 2-4 antennája révén párhuzamosan képes továbbítani a vevőnek, ahol az adatdarabok újra összefűzésre kerülnek. Ezáltal is növelve a sávszélességet. A több antennás működés egyik hátránya a megnövekedett fogyasztás.

Az n-es routerek képesek vagy csak g-s vagy csak n-es módban, avagy ezek keverékeként működni.

4.1.2 Wifi 5 (802.11ac)

Egyértelműen elmondható, hogy ez az előző szabvány továbbfejlesztése. Nagyobb sávszélesség, teljesítmény és megbízhatóbb működés jellemzi. A szabvány több érdekességgel is rendelkezik.

A sűrűn lakott helyeken jelentős probléma a túlzsúfoltság. Egy 10 emeletes házban, emeletenként 4 lakással számolva 40 lakással számolhatunk, min. 1, de akár több wifi access point-tal. Ebben az esetben biztos, hogy a szórt jelek zavarják majd egymást. Ezért is jó ötlet, ha elhagyva a túlzsúfolt 2.4GHz-es tartományt, 5GHz-en sugárzunk - a 802.11ac szabvány csak 5GHz-en működik.

A szabvány másik érdekessége, hogy két fázisban jelent meg. Először 2013-ban hagyta jóvá a Wifi Alliance a wave 1-et, majd csak 3 évvel később (2016) a wave-2-t.

Főbb jellemzői:

- 5GHz-es működés
- 20 és 40 mellett már megjelennek a 80 és a 160MHz-es frekvencia tartományok
- 8 adatfolyam egyidejű kezelése
- továbbfejlesztett MIMO -> MU-MIMO
- az eddigieknél magasabb moduláció, 256QAM
- Beam Forming

A magasabb frekvencia miatt azonban szükség volt egy kis trükk bevezetésére, ez az ún. Beam Forming, mely révén képes megállapítani, hogy a jel honnan érkezett és a jel körkörösen minden irányban történő szórása helyett célirányosan szórni azt.

A MU-MIMO, azaz a Multi User MIMO csak a wave 2-ben jelent meg (a wave 1-ben ugyanaz van, mint az előző n-es szabványban). Ez azt jelenti, hogy a router képes az egyik antennáját, azaz az egyik adatfolyamát egy dedikált eszközhöz rendelni, ezáltal stabilabb és gyorsabb kommunikációt elérhető. Fontos azonban megjegyezni, hogy mindez csak a letöltés irányában igaz.

4.1.3 Wifi 6 (802.11ax)

A legújabb szabvány, amelynél tovább gyorsult a sávszélesség és növekedett a hatótávolság.

Az IoT eszközök megjelenésével még több eszköz kapcsolódik egyetlen hálózati eszközhöz. Ennek hatására a wifi szabványban is lépni kellett abba az irányba, hogy elkerülhető legyen a hálózat instabillá vagy lassúvá válása abban az esetben, ha túl sok eszköz csatlakozna a hálózati eszközhöz vezeték nélkül.

Emiatt számos fejlesztés jelent meg az új generációban.

Főbb jellemzői:

- 2.4GHz és 5GHz-es működés
- 20, 40, 80, 160MHz-es frekvencia tartomány
- OFDMA

- 1024QAM
- Tovább fejlesztett MU-MIMO
- Target Wake Time (TWT) technológia bemutatása
- Resource unit megoldás bevezetése
- BSS coloring

Az OFDMA (Orthogonal Frequency Division Multiple Access – ortogonális frekvenciaosztású többszörös hozzáférés)-nek köszönhetően az eszközök kéréseit a router egyidejűleg tudja kiszolgálni. Ez a korábbi verziókban nem volt megoldott, azoknál ez csak egymás után valósulhatott meg.

A továbbfejlesztett MU-MIMO technológia itt már kétirányú, azaz fel- és letöltésnél is használható és egyszerre akár 8 eszközt is képes kezelni.

A TWT egy olyan új technológia, melynél a router csak bizonyos időnként szólít meg egy eszközt, addig hagyja azt aludni. A kliens és az AP gyakorlatilag előre leegyeztetik, hogy milyen időközönként kommunikálnak majd egymással, ezáltal az érzékeny fogyasztású eszközök – melyeknél kiemelten fontos a fogyasztás – standby üzemmódba kerülnek. Rengeteg akkumulátorról vagy elemről üzemelő kis IoT eszköz létezik melynél ez a technológia megsokszorozhatja az egyszeri feltöltéssel üzemelő órák számát.

A BSS coloring által a router képes ’megszínezni’ azokat az adatcsomagokat, amelyek idegen hálózathoz tartoznak. Ez segít a különböző hálózati interferenciák leküzdésében és segít a kliensnek azonosítani, hogy mi tartozik rá és mi nem.

Természetesen a megelőző összes generációval történő teljes kompatibilitás jellemzi.

4.1.4 Wifi biztonság

A vezeték nélküli hálózat rendkívül kényelmessé teheti számunkra a kommunikációt. Mint az előzőekben láthattuk, sebesség terén napjainkban nemcsak felveszik a versenyt a vezetékes kommunikációval, hanem otthoni használatban ezt akár felül is múlják. Ennek a kényelemnek azonban meglehet az ára, ha nem figyelünk oda. Mivel ez a kommunikáció sajnos akár bárki számára elérhető, így biztonsági szempontból sokkal sérülékenyebb és megbízhatatlanabb, mint a vezetékes.

Mindezek miatt rendkívül fontos, hogy a vezeték nélküli hálózatunkat védjük meg az illetéktelen támadóktól.

A legalapvetőbb védekezési forma, hogy a WiFi-nkre állítsunk be megfelelő biztonsági protokollt és ne hagyjuk nyíltan, jelszó nélkül.

	WEP	WPA	WPA2
Adatvédelem, titkosítás	Rivest Cipher 4 (RC4)	Időbeli kulcs-integritási protokoll (TKIP)	CCMP és AES
Hitelesítés	WEP-Open és WEP-Shared	WPA-PSK és WPA-Enterprise	WPA2-Personal és WPA2-Enterprise
Az adatok integritása	CRC-32	Message Integrity Code (MIC)	Cipher blokk chaining message hitelesítési kód (CBC-MAC)
Kulcskezelés	Nem biztosított	Négyirányú kézfogás	Négyirányú kézfogás
Hardver kompatibilitás	Minden hardver	Minden hardver	A régebbi hálózati interfészkártyák nem támogatottak (2006-ig)
Sérülékenység	Rendkívül sebezhető: Chopchop, Bittau féle fragmentáció és a DoS támadásokra	Jobb, de még mindig sebezhető: Chopchop, Ohigashi-Morii, WPA-PSK és DoS támadások	A legkevésbé sebezhető, bár még mindig hajlamosak a DoS támadásokra, MAC spoofing
Konfiguráció	Könnyen konfigurálható	Nehezebb konfigurálni	A WPA-Personal könnyen konfigurálható, a WPA-Enterprise kevésbé
Visszajátszásos védelem	Nincs védelem	Szekvencia számláló a visszajátszás védelme érdekében	A 48 bites datagram / csomagszámmal védekezik a visszajátszás elleni támadásoktól

4.3. táblázat: Wifi Protokollok összehasonlító táblázat

Egy új protokoll a WPA3. Ennek az alapjait 2018-ban publikálta a Wifi Alliance. Az eszköz gyártók csak mostanában kezdik ezt támogatni, az én háztartásomban egyedül a saját routerem képes ezt kezelni, de még nincs olyan eszközöm ami képes lenne ilyen módon csatlakozni, így ezt nem fogom tudni tesztelni semmilyen módon.

Újításai:

- A jelszóerősséget 192 bitre növeli (WPA3-Ent esetén)
- SAE (Simultaneous Authetntication of Equals) bevezetése az előre megosztott kulccsere helyett.
    - offline szótáras támadás ellen
    - kulcs úlratelepítáési támadások ellen (KRACK)
- Wifi Easy Connect és Enhanced Open bevezetése a rendkívül sérülékeny WPS helyett és a nyílt hotspot titkosításra

Könnyedén felírhatunk egy biztonsági rangsort:

1. WPA3-AES
2. WPA2-AES
3. WPA-AES
4. WPA-TKIP/AES
5. WPA-TKIP
6. WEP
7. Nyílt

Természetesen van még sok egyéb lehetőség ezek mellett, ez csak a legalapvetőbb védelmi vonal. Mindezeket majd egy későbbi fejezetben fogom részletezni.

4.2 Zigbee

A 2000-es évek elejére szükségessé vált egy olyan kommunikációs forma, amely nem a sebességre és a teljesítményre vagy a hatótávolságra helyezte a hangsúlyt, hanem a rövidtávú és kis sebességű átvitelre. Jellemzően ezek a szenzor és vezérlő egységekkel folytatott kommunikációs tulajdonságok. Ezeknél az eszközöknél viszont annál fontosabb a biztonság, az alacsony energia szükséglet és a rövid válaszidő.

2002-ben ezek mentén jött létre a Zigbee Alliance nevű csoportosulás, melyhez 225 cég csatlakozott, mint például a Philips, Samsung, Huawei vagy a Texas Instruments. Jelenleg több, mint 3500 féle tanúsított, ill. általuk jóváhagyott eszköz érhető el a piacon és példányszámban több, mint 300 millió darabról beszélhetünk.

3 verziója jelent meg:

- Zigbee 1.0 (2004)
- Zigbee 2007 (2007)
- Zigbee 3.0 (2016)

Főbb jellemzői:

- WPAN (Wireless Personal Area Network)
- Használt frekvenciák:
  - 2.4GHz (globális, általános működési frekvencia)
  - 915MHz (US)
  - 868MHz (EU)
- Nem hatol vízbe
- Maximális sávszélesség: 250kbps
- Támogatja a mesh hálózatokat
- Biztonságos (AES-128)

- Nagyon rövid válaszidő (néhány msec)
- Elméleti hálózat méret 65000
- Alacsony fogyasztás
- Visszafelé teljes kompatibilitás

4.2.1 Vezeték nélküli személyes hálózatok (WPAN)

A WLAN testvére, de a hangsúly itt nem a sebességre, hanem az alacsony fogyasztásra esik. Ezek 3 osztályba sorolhatók:

- nagy sebességű (HR-WPAN):
  11-55Mbps, (IEEE 802.15.3)
- közepes sebességű (MR-WPAN):
  1-3Mbps, pl. Bluetooth
- alacsony sebességű (LR-WPAN):
  max. 250kbps, pl. ZigBee, (IEEE 802.15.4)

4.2.2 Eszköz típusok

- Teljes funkciójú eszközök (Fully Function Device, FDD), melyek bármilyen szerepet betölthetnek a hálózatban, bármely eszközzel képesek kommunikációt létesíteni.
- Korlátozott funkciójú eszközök (Reduced Function Device, RFD), amelyek csak FFD-vel kommunikálnak.

4.2.3 Eszköz szerepek

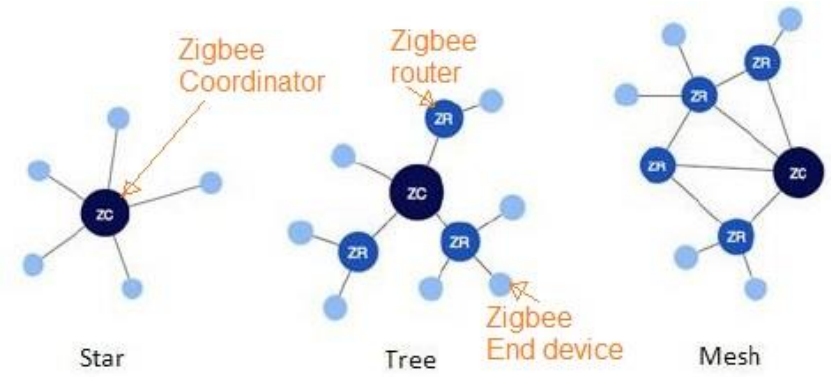
a) IEEE 802.15.4 szerepkörök
   - Koordinátor: üzeneteket közvetít
   - PAN koordinátor: ha hálózatot felügyel pluszban
   - Eszköz: végberendezés, végrehajtó

b) ZigBee szerepkörök
   - ZigBee koordinátor = PAN koordinátor
   - ZigBee router = koordinátor
   - Eszköz = eszköz

4.2.4 Mesh hálózatok

Amint legalább két eszköz aktívvá válik, a ZigBee hálózat azonnal létrejön, gyakorlatilag önmagát hozza létre. A mesh hálózat esetén a legelső FFD eszköz, amely a kommunikációt indítja válik ZigBee koordinátorrá. Majd ezután már felügyeli a hálózatot és irányítani képes a többi eszközt. Ha valamelyik résztvevő a hálózatban gyengélkedik vagy a hálózati paraméterek megváltoznak, akkor a hálózat új útvonalat számít és azon keresztül küldi el a csomagot. A mesh hálózat egy elosztott hálózat, folyamatosan változik ezáltal alkalmazkodva a kialakult helyzethez.

4.2.5 ZigBee topológiák:



4.1. ábra: ZigBee topológiák

4.2.6 Eszköz kategóriák a ZigBee-n belül

a) Központi egység
- folyamatos tápellátás;
- nagy számítási kapacitás;
- hálózat felállítása és üzeneteinek küldése;
- egyszerű csomópontok szervezése, paramétereinek tárolása;
- a párosított csomópontok üzeneteinek továbbítása.

b) Egyszerű csomópont
- csak a központi egységgel tudnak közvetlenül kommunikálni;
- véges tápellátás;
- folyamatosan keresik az elérhető hálózatokat;
- adatküldés cask akkor, ha szükséges;
- ha nem küldenek vagy nincs számukra tárolt adat, akkor igyekeznek minél több időt alvó módban tölteni

Fontos megemlíteni, hogy ahhoz, hogy a ZigBee hálózatot integrálni lehessen a meglévő wifi hálózatunkba, szükség lesz egy ZigBee gateway-re, egy átjáróra. Ezáltal a ZigBee protokoll Internet protokollra kerül átfordításra.

4.3 Z-wave

Az okos vezérlés világában ez a protokoll kéz a kézben jár a ZigBee-vel. Sok köztük a hasonlóság, ugyanazt a célterületet célozták meg, azaz az alacsony fogyasztású, rövid válaszidejű szenzorok, kapcsolók vezérlését.

A protokollt egy dán startup cég, a Zensys fejlesztette ki 1999-ben és ezt a világítástechnikában alkalmazta először. Majd 2003-ban és 2005-ben jelentek meg új chipekkel, melyek alacsonyabb fogyasztást és nagyobb hatótávot jelentettek (100-as és 200-as sorozat). A ZigBee Allience-hez hasonlóan, a Z-wave is megalakította a saját szövetségét Z-wave Allience néven, melyhez több nagy cég csatlakozott (pl. Silicon Labs). 2008-ban a protokoll jogait felvásárolta a Sigma Designs, majd végül 2018-ban minden a már említett Silicon Labs-hoz került. 2020 egy igen fontos dátum a protokoll történetében, ugyanis a Z-wave Allience ekkor jelentette be, hogy a protokollt külső fejlesztők részére is megnyitják.

Főbb jellemzői:

- 40 kbps adatátviteli sebesség
- mesh hálózat alapú működés
- maximum 232 eszköz csatlakoztatható egy hálózatba
- használt frekvenciák országonként eltérnek és alfrekvenciákra vannak bontva
- EU: 868.4, 868.42, 869.85
- részletes lista: https://www.silabs.com/wireless/z-wave/global-regions
- hatótáv kb. 100 méter
- maximum 4 ugrás az adatkommunikációban
- a chipeket a Silicon Labs szállítja
- a különböző sorozatok között teljes kompatibilitás
- rendkívül alacsony fogyasztás

Napjainkban több, mint 3300 féle eszköz közül válogathatunk és több, mint 100 millió az eladott eszköz.

A működési frekvenciája a 800-900MHz között mozog, ami azért nagyszerű, mert messziről elkerüli a már meglehetősen túlzsúfolttá vált 2.4GHz-et, így sokkal kevésbé hajlamos interferrálni, mint az előzőek.

A ZigBee-hez hasonlóan a Z-wave is képes mesh rendszerben működni, azaz minden eszköz képes minden másik eszközzel kommunikálni, nincs egy központi egység. A hálózatba kerülésnél az első lépés az eszköz felismertetése és hozzá kell adni a mesh hálózathoz. A hálózati biztonságot az AES-128 biztosítja.

Ahhoz, hogy a Z-wave hálózatunkat illeszteni tudjuk a meglévő wifi hálózatunkhoz, szükséges egy darab Z-wave HUB, mely átjáróként szolgál a 2 protokoll között.

4.4 ZigBee vs. Z-wave

A két protokoll rendkívül hasonló, a választásban mégis segítségünkre lehet néhány fontos eltérés.

A Z-wave egy szabadalmaztatott protokoll, szigorú tanúsítási előírásokkal és szabályokkal. A jogok a Silicon Labs tulajdonában vannak, minden termékre általuk kerül rá a 'Certified', azaz a bevizsgált és tanúsított pecsét, mely szerint az adott termék megfelel minden hardveres és szoftveres előírásnak, amit a Z-wave szabvány jelent. Biztos, hogy az eszköz beilleszthető a Z-wave mesh hálózatba, kommunikálni képes minden hálózati eszközzel, biztosra vehetjük, hogy nem lesznek kompatibilitási problémák.

Ezzel szemben a ZigBee egy nyílt vezeték nélküli kommunikációs protokoll, melyet a ZigBee Allience felügyel és sokkal lazább hitelesítési szabályok jellemzik. A probléma ott kezdődik, hogy az általuk végzett tanúsításnak két összetevője van egy szoftveres és egy hardveres. Lehetséges olyan eset, hogy az egyik összetevőt tanúsítják, míg a másikat nem, de a terméken megjelenik a ZigBee tanúsítvány. Ebben az esetben a hálózatunkhoz illesztve sajnos előléphetnek inkompatibilitási problémák. A szigorú hitelesítési előírások miatt a Z-wave biztonságosabbnak és megbízhatóbbnak tekinthető.

Egy másik szignifikáns különbség a működési frekvencia tartomány. A Z-wave 900MHz körüli, míg a ZigBee a már eddig is telített 2.4GHz-es frekvenciát használja. Tudni kell, hogy az alacsonyabb frekvencián továbbított jelek messzebbre terjednek, igaz ez a mesh rendszerben nem feltétlen fontos tényező. Míg a Z-wave minden régióban más frekvencián működik, addig a ZigBee globálisan egységes.

A következő paraméter az ár, ami egy kifejezetten fontos tényező és ebben egyértelműen győztes a ZigBee: amennyiben Z-wave probokollban gondolkodunk, úgy minimum 2x-es szorzóval számolhatunk.

A két hálózat termékei természetesen nem kompatibilisek egymással. Meg kell említeni ugyanakkor, hogy néhány olyan vezérlőközpont már elérhető a piacon, amely képes mind a két hálózat eszközeit egyszerre kezelni (Pl. Vera Plus).

5. A megoldási módszer kiválasztása, a választás indoklása

5.1 Otthon automatizálási rendszerek rövid bemutatása

Az okos otthon vezérlésre napjainkban igen sok megoldás létezik. Ha veszünk egy egyszerű okos izzót az ikeában, akkor ezt gyakorlatilag bármilyen egyéb tudás nélkül tudjuk használni az otthonunkban, képesek vagyunk vezérelni mobil eszközünkről, akár még egyszerűbb automatizálásokat is képesek lehetünk beállítani, hogy este 8 után kapcsoljon le, reggel 7-kor kapcsoljon be 30 percre, mint ébresztő fény. Ehhez le kell töltenünk a használati utasításban levő instrukciók alapján a mobil eszközünkre a megfelelő (IOS, Android) alkalmazást és ezt telepítve kisebb beállítás után már vezérelhető is az izzó.

Ha az említett dolgok internetének jobban a mélyére nézünk, akkor láthatjuk, hogy az említett okos izzó a Zigbee szabványt használja, ezért, hogy a meglévő okos otthonunkhoz tudjuk majd csatlakoztatni és mobil telefonunkról vezérelni, ehhez szükséges egy központi egység, ami megérti ezt a Zigbee nyelvet és át tudja fordítani, tudja továbbítani a jelet. Ez egy gateway, vagy jeltovábbító. Ezt egyszer kell megvennünk és a további Zigbee ikeás eszközök már ezen fognak kommunikálni.

A letöltött alkalmazásba ezek könnyedén behelyezhetőek, integrálhatóak, ott ezek vezérelhetőek, közöttük létre lehet hozni jeleneteket, automatizmusokat is.

Viszont, ha megtetszik egy Philips hue okosizzó, vagy egy Yeelight termék, akkor ehhez szintén kell venni egy jeltovábbítót és letölteni, telepíteni a saját alkalmazását és oda beilleszteni a vett terméket. Ezeket sajnos nem tudjuk egy alkalmazásból látni, kezelni a gyári szoftveren keresztül. Természetesen igyekszik minden gyártó fejleszteni és egyre szélesebb körben forgalmazni a termékeit, így egy gyártótól viszonylag komplex okos otthon vezérlést valósíthatunk meg.

De sok mindenben kell kompromisszumot kötnünk:

- Nem igazán van átjárhatóság a különböző termékek között
    - egy Yeelight izzó nem jelenhet meg az Ikeás app-ban
- Flexibilitás hiánya
- Cloud alapú vezérlés
    - nincs internet elérés -> nem működik a termék
- Biztonsági aggályok
- Egy gyártótól függés
- Sok különböző alkalmazás

Ezért is nyernek egyre nagyobb teret a gyártófüggetlen okos otthon automatizálási központok. Ezekből én a Home Assistant-ot és az Openhab központot nézem meg, hogy ki tudjam választani a megfelelőt.

Szempontok:

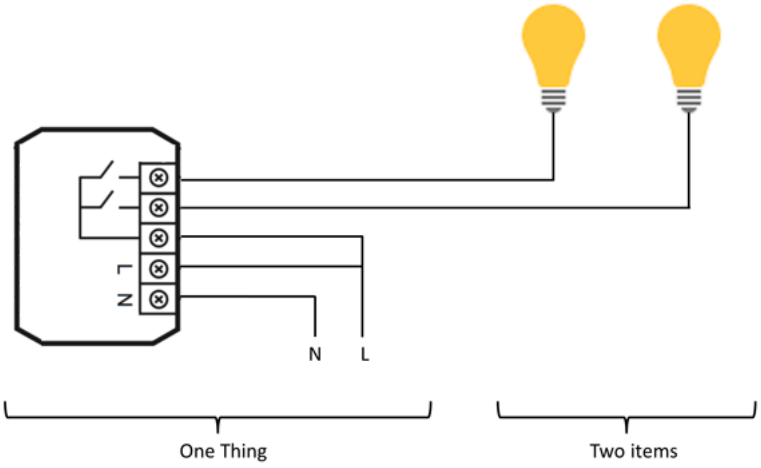
- Ingyenes legyen
- Biztonsági kérdések
- NAS-ra telepíthetőség (saját library, Docker, VM)
- Kezelhetőség
    - programozási ismeretek hiánya
- Kinézet
    - Mobil, webes interface-ek
- Különböző gyártók integrálhatósága
- Különböző protokollak kezelése
- Automatizálási lehetőségek
- Dokumentáció, közösségi támogatottság elérése

5.1.1 Openhab

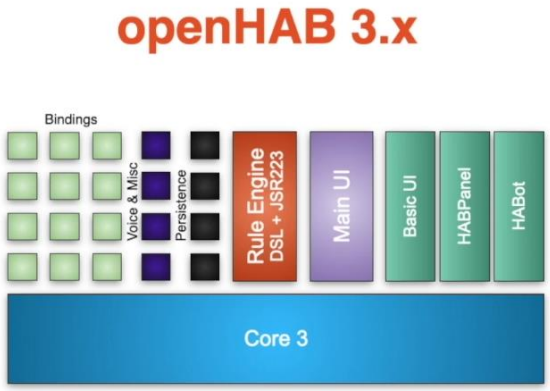
Az OpenHab platformot 2010-ben alkotta meg Kai Kreuzer. Java-ban lett megírva és az Eclipse Smarthome keretrendszeren nyugszik. Az Apache Karaf-ot konténernek és az Eclipse Equinox-ot az OSGi futtató környezetnek és a Jetty-t web szervernek. Jelen dolgozat nem fog kitérni ezek mélyebb jelentésére, bemutatására. Jelenleg a 3.x verziónál tart.

A platform moduláris, hozzá különböző kiegészítéseket lehet csatlakoztatni, „Add-on”-ok formájában. Ezeken keresztül lesz képes a felhasználó kommunikálni az eszközeivel.

Röviden az OpenHab működéséről, koncepciójáról. Dolgok, csatornák, kötelékek, elemek, csatlakozások. Ezekkel operál a szoftver. Ahhoz, hogy integrálni, tudjunk egy eszközt, kell egy konkrét gyártóhoz, dologhoz (Things) megírt kötelék (Bindings). Az OpenHab ezen keresztül tudja, hogy miként kell kommunikáljon a konkrét eszközzel. Az elemek (Items) mutatják be azokat a képességeket, amelyekre a konkrét eszközök logikailag képesek. A dolgok és az elemek között alakulnak ki a csatlakozások (Links). A csatlakozás egy hozzárendelés pontosan egy csatorna és egy elem között. Több csatorna tartozhat több elemhez és több elem is tartozhat több csatornához.



5.1 ábra: OpenHab működési modell



5.2 ábra: OpenHab architektúra

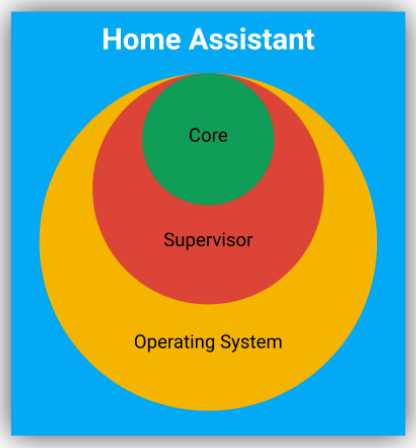
- Core 3: A szoftver alapja, Java-ban megírva
- HABPanel: Az alap felhasználói interface
- HABot: Az OpenHab-en belül futó chat robot.
- Rule Engine DSL+JSR223: logikák szabályok írása javascript, jython vagy groovy használatával
- Main UI, Basic UI, különböző felhasználói felületek
- Persistance: adatok tárolása
- Voice & Misc: TTS (Text to Speech) és STT (Speech to Text) szolgáltatások és egyéb lehetőségek
- Bindings: Kötelékek, kötések a különböző gyártóktól származó eszközökhöz

Sokféle gyári lemezképe van, Linux, Windows, Mac OS, Docker, Raspberry.

5.1.2 Home Assistant

Egy másik széles körben használt okos otthon automatizálási központ a Home Assistant. Ezt Python-ban fejlesztették és fejlesztik a mai mapig. Nyílt forráskódú termék, az Apache 2.0 licensze alatt.

A szoftvert 2013-ban Paulus Schoutsen a Github-on publikálta. Ez egy közösségi kódlerakat, bárki regisztrálhat és publikálhat rá. 2017-ben jelent meg a Hass.io, ami egy komplett operációs rendszerbe ágyazott Home Assistant-ot jelentett, hogy biztosítsa a könyebb használhatóságot. Ebben a verzióban került bevezetésre az OpenHab-nál is már említett „Add-on" megoldás. Majd 2020-ban ismét egy nagyobb változást élt meg a szoftver, ekkor részegységekre darabolták.



5.3. ábra: Home Assistant modell

4 különböző telepítési lehetőség közül választhatunk jelenleg:

- Home Assistant OS: Leggyakoribb telepítése. Ez egy minimalista OS, ami ki van élezve a HA futtatására. Ebben benne vannak előre telepített Add-on-ok és a Supervisor is.
- Home Assitant Container: Önállló konténer alapú telepítése a Core-nak. Nincs és nem is lehet Supervisor
- Home Assistant Supervised: A Supervisor kézi telepítése
- Home Assistant Core: A Core kézi telepítése.

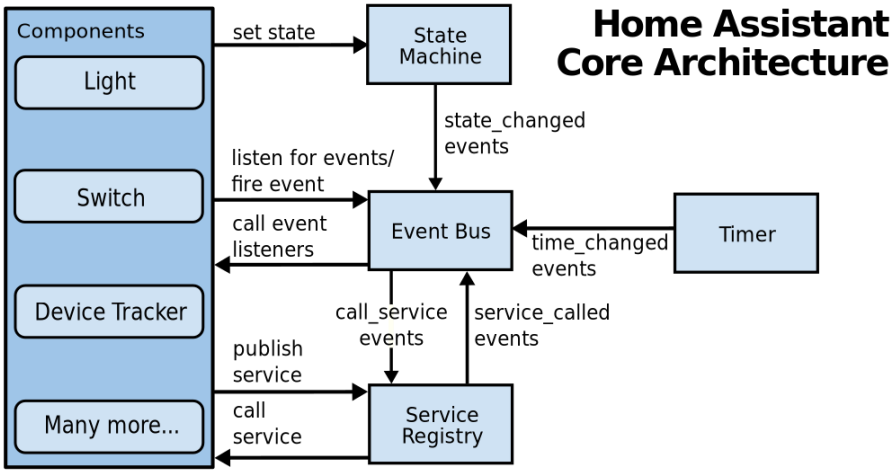
Hogy mire telepíthető? Gyakorlatilag bármire. Windows, Linux, Intel NUC, Mac OS, Raspberry, Docker, VM. Sőt pár hete kiadtak egy Home Assistant Blue nevű mini számítógépet, melyre a HA OS előre telepítve van. Ez egy ODROID-N2+-hardverre készített dobozos megoldás, all-in-one termék.

A különböző telepítési megoldásokkal, különböző képességekre tehetünk szert, vagy éppen kell róluk lemondanunk.

	OS	Konténer	Core	Supervised
Automatizációk	✓	✓	✓	✓
Lovelace UI	✓	✓	✓	✓
Integrációk	✓	✓	✓	✓
Blueprints	✓	✓	✓	✓
Konténer használat	✓	✓	✗	✓
Supervisor	✓	✗	✗	✓
Add-ons	✓	✗	✗	✓
Snapshots	✓	✗	✗	✓
Managed OS	✓	✗	✗	✗

5.1. táblázat: Home Assistant verziók

Ami még fontos lehet, az a Home Assistant Core architektúrája



5.4. ábra: Home Assistant Core architektúra

Mint látható a Core-nak 4 fő eleme van:

- Event Bus: A HA szíve. Ez kezeli a ki és bejövő esemény triggereket
- State Machine: Figyeli a dolgok állapotát és ha változás van, jelzi
- Service Registry: hallgatózik az Event bus irányába, hogy van-e hívás és regisztrálja a szolgáltatásokat
- Timer: Időzítő. Minden másodpercben küld egy *time_changed* nevű eseményt

A HA Core-hoz lehetőségünk van rengeteg integrációt kapcsolni, amik egyénenként felelnek egy-egy előre meghatározott területért (világítás, klíma…) és az ezeken belül meghatározott formátumért és adatkörökért. Ezek az integrációk vagy valamilyen szolgáltatást nyújtanak, vagy állapot információt szolgáltatnak, vagy befogadnak, vagy közvetítenek információt, utasítást. Ezeket bárki fejlesztheti, írhatja, módosíthatja, github-on közzé teheti, python-ban.

5.1.3 Összehasonlítás a felvázolt szempontok szerint

- **Ingyenes legyen**

Mindkettő ingyenesen terjeszthető licenszelés alá esik, ingyenesen letölthető, használható. Mindkettő rendelkezik távoli, felhő alapú eléréssel, de amíg az OpenHab myopenhab.org ingyenes, addig a HA távoli, felhő alapú elérése fizetős, ez a NabuCasa.com. Mindkettő biztosít titkosított távoli elérést.

Mivel én ezt a gyártói felhő alapú elérést biztos, hogy nem szeretném használni, így számomra nincs jelentősége.

- **Biztonsági kérdések**

A fejlesztői távoli elérések hiányában saját megoldás lesz, így ez nem kell, hogy befolyásolja a döntést. Mindkettőnek nagyon széleskörű a támogatása, az OH 1 éve váltott fő verziót, azaz nagyon sok mindenben megújult, de ilyenkor mindig kell egy kis idő, amíg kicsit ki tud forrni, még biztonsági kérdésekben is. A HA-nak fix havi kiadásai vannak és mikor megjelentek CVE sérülékenységek valamelyik főbb komponensre, napokon belül jöttek a friss patch-ek, kiadások, így talán azt mondhatjuk, hogy a HA gyorsabban képes reagálni, de ebben még nincs még nagy tapasztalatom és ez nem is számottevő különbség.

- **NAS-ra telepíthetőség** (saját library, Docker, VM)

Két NAS-al rendelkezem, egy Synology DS414j-vel, ez egy régebbi típus (7+ év), korlátozott erőforrással és kevésbé flexibilis OS-el és egy új QNAP TS543D-vel, ami, bővíthető, többmagos processzorral rendelkezik és képes saját VM-eket és Docker környezeteket futtatni. Így elméletileg mindkettőt problémamentesen tudom telepíteni és futtatni.

- **Kezelhetőség**
    - saját programozási ismereteim hiánya

A HA rendelkezik egy automata felderítés funkcióval, ami hasznos lehet, főleg az elején, kezdő felhasználóknak. Egyébként minkettőnél a webes felületen és konfigurációs fájlok írásával lehet kezelni az eszközök, integrációk, automatizmusok hozzáadását,

módosítását, törlését. Ezen konfigurációs fájlokat a két platformon eltérő nyelven kell megtennünk. Az OH az Xtend-et, a HA a YAML-t használja. Két egyszerű példa.

```yaml
input_number:
  illumination_bedroom_low_cutoff:
    name: 'Illumination bedroom low cutoff value'
    min: 1
    max: 100
    step: 1
  bedroom_no_motion_turn_off_delay:
    name: 'Time in minutes automation waits to turn off bedroom lights after no motio
    min: 1
    max: 120
input_boolean:
  sleepmode:
    name: 'Sleepmode'
input_datetime:
  bedroom_lights_turn_on_before_limit:
    name: 'Bedroom lights no longer turn of before this time'
    has_date: false
    has_time: true
  bedroom_lights_turn_on_after_limit:
    name: 'Bedroom lights no longer turn of after this time'
    has_date: false
    has_time: true
```

5.5. ábra: YAML példa

```
rule “Exercise every morning”
when
     Time cron “0 0 8 1/1 * ? *”
then
     harmonyStartActivity(“Exercise”)
end

Complic
     import org.openhab.core.library.types.*
     importorg.openhab.model.script.actions.*
     import java.lang.String

rule “Humidity Monitor”
when Time cron “0 * * * * ?”
then
     var prevHigh = 0
     var highHum = “”

     Humidity?.members.forEach[hum|
          logDebug(“humidity.rules”, hum.name);
          if(hum.state as DecimalType > prevHigh){
               prevHigh = hum.state
               highHum = hum.name + “: ” + hum.state + “%”
          }
     ]
     logDebug(“humidity.rules”, highHum);
     postUpdate(Dehumidifier_Needed,highHum);
end
```

5.6. ábra: Xtend példa

A Yaml egy programozásban nem jártas ember számára könnyebben értelmezhető, olvasható, tanulható.

Az automatizmusok beállításában a HA ajánl webes kattintós, varázslószerű megoldást és lehet bele integrálni a Node-RED folyamat alapú fejlesztői rendszert. Az OH a Blockly-t adja alapból és ide is lehet integrálni a Node-RED-et.

- **Frissítés**

HA-ban ez nagyon egyszerű, a webes felületen elvégezhető, gyakorlatilag egy kattintással. Az OH ettől kicsit bonyolultabb, nála parancssorban kell ezt elvégezni.

- **Kinézet**
    - Mobil, webes interface-ek

Ez tényleg ízlés kérdése, legalábbis a legelején. Mindkettő testre szabható, így szinte bármit meg tudunk valósítani. Az OH-nak több kezelői felülete van, ami az elején zavaró lehet, míg a HA-nak ez egy komplex, de mégis átlátható felületben jelenik meg és már az első telepítésnél „kinéz valahogy”, ami a kezdetekben segítség lehet. Mindkettő rendelkezik természetesen mobil app-al android és IOS-re is.

- **Különböző gyártók integrálhatósága**

Mindkettő nagyon széles körű támogatással rendelkezik integrációk terén, így a népszerűbb gyártókkal aligha esz gondunk.

- **Különböző protokollak kezelése**

Itt sem lehet különbséget tenni, mindkettő széles körben kezeli a szükséges protokollokat.

- **Dokumentáció, közösségi támogatottság elérése**

Mindkét projektnek van állandóan frissített, jó dokumentációja, melyben minden szükséges megtalálható. A HA nemrég nyitott egy saját fejlesztői oldalt erre a célra, developers.home-assistant.io. Vannak „hivatalos” közösségi fórumok, ahol bármilyen kérdés, kérés esetén a közösség igyekszik segíteni.

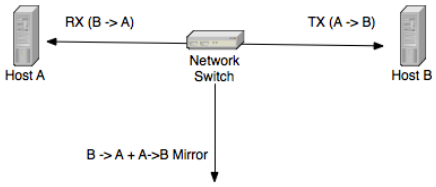
A központ kiválasztásánál nem igazán lehet rosszul választani, mindkettő jól dokumentált, kiforrott platform, melyek széleskörű támogatással rendelkeznek. Mivel nincsenek gyakorlati ismereteim, ezért a kutatásaim alapján úgy látom, hogy a Home Assistant talán jobban kezdőbarát, így e mellé teszem le a voksom.

5.2 Kommunikáció elemzéshez tartozó eszközök

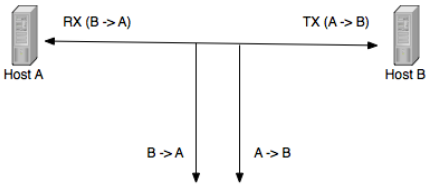
Fontos szempont volt, hogy minden szoftver ingyenes legyen. Arra helyeztem fókuszt, hogy a kiválasztott eszközök segítségével képet kapjak az otthoni hálózatról, a telepített eszközökről.

5.2.1 Ntopng

Röviden, ez egy hálózati forgalom monitorozó és elemző eszköz. Segítségével be tudunk ékelődni a hálózati adatfolyamra, vagy egyéb hálózati megoldásokkal az eszköz számára elérhetővé tudjuk tenni a forgalmi adatokat. Erre kétféle lehetőségünk van. Port tükrözéssel (SPAN, Switched Port Analyzer) vagy hálózati TAP-el (Test Access Points). Míg a tükrözés egy aktív csomag irányítás, a csomagokat duplikálja és átküldi egy arra kijelölt switch portra, addig a hálózati TAP egy passzív technika. Ebben az esetben a csomagokat lemásolják és nem állnak bele a forgalomba egy aktív eszközzel. A TAP-ok jóval drágább eszközök.



5.7. ábra: Port mirror



5.8. ábra: Hálózati TAP

Az *ntopng* elődje az *ntop* volt, ami nagyon hasonló a jól ismert linux-os *top* parancshoz, mellyel a folyamatokra tudunk ránézni. Az *ntopng* egy web alapú, titkosítással ellátott eszköz, amely a *libpcap* linux beépülő modulra és a *pf_ring* kernel modulra épül. Képes valós idejű és historikus elemzésre, melyet a *redis* nyílt forráskódú és ingyenes adatbázissal valósít meg. Néhány fontos tulajdonsága:

- van belőle ingyenes, közösségi kiadás
- aktív hostok figyelése
- riasztások kezelése
- L7 alkalmazás rétegbeli protokoll alapú információ gyűjtés (nDPI)
- legnagyobb forgalmazók
- geolokációs megjelenítés
- syslog integráció
- snmp v1/v2c/v3 támogatás
- históriku adatok

Az nDPI-t az ntopng fejlesztette, hogy azonosítani tudja a különböző L7 protokollokat. Ez egy gyűjtemény, amit folyamatosan fejlesztenek, bővítenek. Az aktuális állapot a hivatalos honlapon megtalálható: https://www.ntop.org/products/deep-packet-inspection/ndpi/

5.2.2 InfluxDB (TICK Stack)

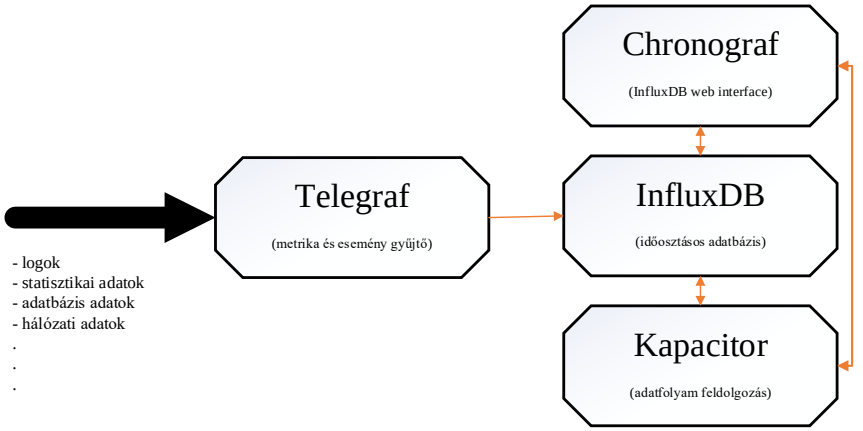
Az InfluxDB egy időbélyegzett adatbázis. Úgy lett megalkotva, hogy segítségével könnyen lehessen tárolni nagy mennyiségű adatokat, metrikákat. Az időnek különleges értéke van és nemcsak egy adat, mint a hagyományos relációs adatbázisokban.

Maga az InfluxDB felel az adatok tárolásáért, de mindez még kiegészül néhány főbb komponenssel. Ezért egyben ezt úgy is szokták hívni, hogy 'TICK stack'.

A Telegrafon keresztül jönnek be az adatok és kerülnek továbbításra az adatbázis felé. Rajta keresztül lehet konfigurálni az adatköröket.

Az InfluxDB-nek nincs grafikus felülete, parancssorból adminisztrálható. A Chronograf telepítésével az adatbázis fölé helyezhetünk egy webes interfészt, mellyel sokkal könnyebben, átláthatóbban tudjuk konfigurálni.

Az utolsó elem a Kapacitor. Vele tudunk definiálni különböző funkciókat, melyeket a felhasználás során szeretnénk figyelni. Különböző értesítéseket, küszöbértékeket.



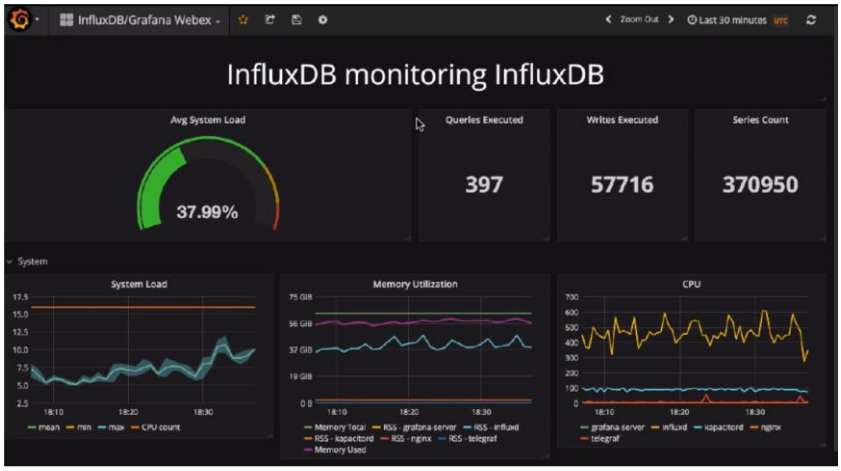
5.9. ábra: TICK stack

Mindez egy ingyenes termék, mely GO-ban íródott és mindenféle függőségek nélkül telepíthető. Egy SQL szerű lekérő nyelve van, InfluxQL.

Gyakran jár kéz a kézben egy szintén ingyenes szoftverrel, a Grafana-val.

5.2.3 Grafana

Legfőképp egy vizualizáló eszköz, de képes értesítéseket is kezelni. Meg lehet táplálni sokféle adatforrással, köztük természetesen az InfluxDB-vel is és számtalan előre megírt dashboard-on keresztül gyönyörű grafikonokat rajzol. Természetesen megcsinálhatjuk a sajátunkat is és az előre megírtakba is könnyedén bele tudunk módosítani.



5.10. ábra: grafana+influxdb példa

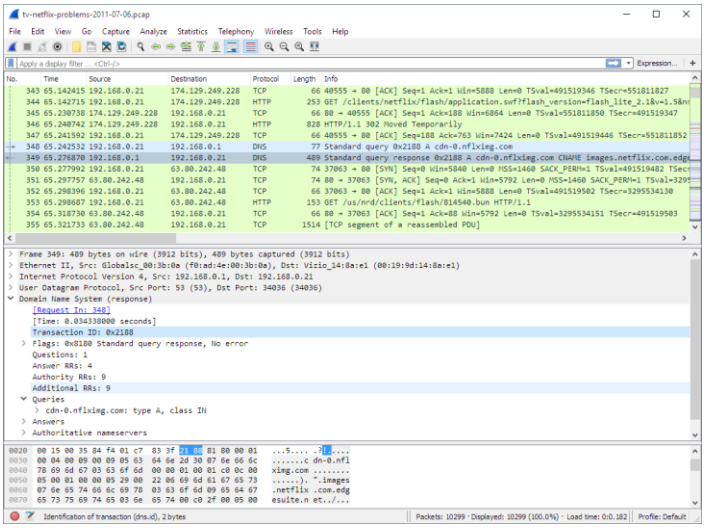
5.2.4 Wireshark

A Wireshark széles körben ismert és használt csomag elemző eszköz. Gyakorlatilag megkerülhetetlenné vált a hálózati kommunikáció analízisénél egyszerű telepíthetősége, könnyű kezelhetősége és protokoll támogatottsága okán.

A története 1998-ra nyúlik vissza, ekkor fejlesztette Gerald Combs, akkor még Ethereal néven. Majd 2006-ban különböző okok miatt nem használhatta tovább a nevet, így született meg a ma ismert Wireshark elnevezés.

- Több, mint 1000 protokollt ismer
- Rendkívül felhasználóbarát grafikus felülettel rendelkezik, mindemellett képes parancssorban is működni
- Teljesen ingyenes, nyílt forráskódú, GPL licenszelésű
- Könnyen telepíthető Windows, Linux és MAC OS-re is.
- Testre szabható (profilok kezelése)

Az alapvető működése igen egyszerű. Ki kell választani egy, vagy több, a telepített gépen található interface-t, melyet akár promiszkusz és/vagy monitor módba is kapcsolhatunk. Ezzel minden csomagot érzékelni tud majd és nemcsak azt, ami neki van címezve. Majd elindíthatjuk a rögzítést és az eszköz elkezdi rögzíteni csomagok százait, ezreit, azok teljes tartalmával. Képes felismerni a protokollokat, azokat működésük szerint megjeleníteni. Természetesen képes elő és utószűrni, statisztikázni és még rengeteg mindenre alkalmas az analízis terén.



5.11. ábra: Wireshark példa

5.2.5 Kismet

A Kismet egy vezeték nélküli hálózat és eszköz felderítő, lehallgató és behatolás érzékelő keretrendszer. Képes működni Linux és Windows környezetben is.

- Képes azonosítani az Acces Pointokat és a hozzájuk kapcsolódó klienseket
- Nyílt forráskódú, ingyenes..
- Képes a működés közbeni csatorna váltásra.
- Az elfogott csomagokat Wireshark számára olvasható módon tudja elmenteni a későbbi elemzés végett.
- Észlelni tudja, ha éppen van élő másik felderítő eszköz.

5.2.6 Aircrack-ng

A nyers vezeték nélküli 802.11 keretek felderítésére és elkapására használt programok. Ezek is képesek fellistázni a látható AP-ket, a hozzájuk kapcsolódó klienseket, állomásokat. Ingyenesek, GPLv2 licensz alatt futnak.

6. Otthoni infrastruktúra és a használt rendszerek rövid bemutatása

A ház uplinkje egy optikán bejövő, elméleti 1Gbps adatátvitel sebességű DSL kapcsolat. A szolgáltató egy Huawei típusú routeren keresztül végződteti a végpontot. Az eszköz képes kezelni ezt a sávszélességet, mellette képes egy 300Mbps-es vezeték nélküli hálózat szórására és kezelésére is. Továbbá a szolgáltató bekötéskor NAT-olt IP-t adott, melyet az ügyfélszolgálatukon keresztül azonnal megújíttattam egy publikus IP címre. Erre azért volt szükség, hogy az otthoni LAN hálózatot a jövőben képes legyek elérni. Ehhez az internet felől csatlakozni kell tudni egy, az otthoni hálózatban telepített és felkonfigurált VPN szerverhez.

A szolgáltató routerét átkonfiguráltam bridge módba az alap PPPoE-ről. Ez gyakorlatilag azt jelenti, hogy nem a Huawei jelentkezik fel a DSL hálózatra az előfizetői szerződésben megadott autentikációs adatokkal, hanem egy saját magam által választott router. A két eszközt szimplán egy Cat5E jelölésű UTP kábellel összekötöttem. Mindemellett kikapcsoltam a vezeték nélküli és a DHCP szolgáltatását. Így az eszköz gyakorlatilag egy média konverternek tekinthető, ami a bejövő optikán érkező jelet átalakítja rézre.

6.1 Router-ek

6.1.1 Asus RT-AX92U

A vételkor és mióta használatban van, úgy gondolom, hogy ez egy jövőt álló eszköz. Néhány főbb tulajdonsága:

- 802.11n, ac és ax sávokon is működik
- Képes WPA2/3 alapú titkosításra
- Beépített IPS/IDS
- VPN szerver képesség
- (D)DOS védelem, beépített tűzfal
- Port továbbítás lehetősége
- Manuális IP cím kiosztás
- Adaptív QoS
- Forgalomelemzés

6.1.2 TP-Link TL-WR1043ND

Ez egy régi Wireless N-es átviteli sebességre képes router. A ház nagysága miatt a fő router nem volt képes stabilan beszórni a teljes házat. Plusz került kívülre is egy-két eszköz (kamera, okos hálózati aljzat). Ezért szükségem volt rá, hogy a meglévő kialakított LAN hálózat stabilabban és nagyobb területen legyen szórva.

Ezt többféleképpen meg lehet oldani, lehet venni WiFi jelerősítőt, vagy a hálózati aljzatba dugható, a ház elektromos kábelezését használó eszközt, melynek mindkét végébe becsatlakoztatható a LAN kábel. Én találtam otthon egy régi routert, melyre feltettem egy dd-wrt firmware-t. Ezután már csak client bridge módra kellett állítanom, adni neki egy helyi DHCP tartományi fix IP címet, a fő routert megadni, mint átjáró és mint dns szerver. Csatlakoztatni a meglévő WiFi hálózatra, majd létrehozni egy virtuális interface-t, melyet AP módba kellett rakni és beállítani ugyanazokat az adatokat, ahogyan a fő router autentikál és ugyanazt az SSID-t.

Ezek után már csak jól el kellett helyezni és stabilizálta a vezeték nélküli hálózatot.

6.2 NAS-ok

Már több, mint 10 éve vettem az első NAS-omat és már a kezdetekkor nagy hasznát vettem. Most két 4 fiókos NAS egységet használok, két különböző gyártótól.

6.2.1 Qnap TS-453D NAS

Linux alapú operációs rendszerrel működő, erős hardverrel rendelkező Hálózati adattároló

Főbb tulajdonságai:

- 2db 2,5Gbps Ethernet csatlakozóval rendelkezik
- 4 HDD fiókos, jelenleg 2x1TB Seagate Ironwolf Pro Winchester üzemel benne RAID 1-be kötve a nagyobb adatbiztonság miatt
- Négymagos 2GHz-es processzor
- 8GB DDR4-es memória
- PCIe bővíthetőség
- HDMI kimenet
- Saját virtualizációs platform
- Saját Docker konténer állomása van
- 8db ingyenes IP kamera csatlakozási lehetőség

Jelenleg fel van bővítve egy Intel WiFi 6 AX200 típusú PCIe kártyával, mellyel az eszköz képessé vált vezeték nélkül csatlakozni a routerhez. Erre azért volt szükség, hogy az eszközt el tudjam zárni egy szellőzéssel rendelkező szekrénybe, mert elég hangos már így, a 2db SATA merevlemezzel is és ez csak hangosabb lesz, ha fel lesz bővítve 4-re. Így csak hálózati feszültséget kell számára biztosítanom.

Ez a rendszer kiépítésekor problémát is okozott, mert a beépített hálózati kártyát a QNAP operációs rendszere úgynevezett WiFi kliensnek látja. Ez azért baj, mert a virtualizációs platformján létrehozott, üzemeltetett virtuális gépet, egy virtuális switchen keresztül kell kiadni egy fizikai adapternek. Ahhoz, hogy ezt LAN-on keresztül el lehessen érni, bármit bele lehessen majd 'kötni' és ne csak önmagában fusson, a platformján össze kell drótozni. De ezt nem lehetett megtenni a WiFi adapterével. Ezt csak Ethernet portra lehet kiadni. Így ezt egy kis trükkel kellett megoldani, hogy ne kelljen az eszközt beköltöztetni a nappaliba és mégis működhessen megfelelően a virtuális gép. A megoldást a későbbiekben részletezni fogom.

6.2.2 Synology DS414j NAS

Ez egy nagyon régi NAS (8+ éves), viszont a gyártónak hála a mai napig frissítik, megkapja a biztonsági patch-eket. Rendkívül funkciógazdag és könnyű kezelés jellemzi. A szakdolgozatomban részletezett okos otthon megoldásban nem tölt be központi szerepet. Egyetlen, de annál fontosabb szerepe van, a mentések tárolása, szinkronizálása, duplikálása.

Egyébként a család média központja, zene és filmtár. Ide kerülnek feltöltésre a fotók, a telefonok, laptopok mentései, vagy bármilyen fontos anyag, fájl.

- 4*2TB Sata HDD-vel rendelkezik Raid 5-be kötve.
- fix 512MB RAM és 2*1,2GHz CPU
- 1db 1Gbps hálózati csatlakozó

6.3 Egyéb hálózati és szerver komponensek

A hálózati forgalom méréséhez, monitorozásához szükség volt néhány eszköz beszerzésére. Egy kis, hangtalan, alacsony fogyasztású, de úgy gondoltam, hogy nem a legalacsonyabb teljesítményű mini PC-re. Ez lehetett volna egy Raspberry PI, vagy valamilyen hasonló olcsó eszköz, de én inkább egy kis gyári mini PC-t választottam.

Továbbá kellett egy megoldás arra, hogy valahogy be tudjak ékelődni a hálózati forgalomba, hogy lássak mindent. Ezt meg lehet oldani úgynevezett hálózati TAP-ekkel és port tükrözéssel. A TAP-ek elég drágák, így könnyű volt a választás.

6.3.1 HP T430 miniPC

Ez egy hangtalan pici doboz (135mm-135mm*32mm), mely kiválóan alkalmas egy Linux futtatására és ezáltal hosztolni bármilyen szolgáltatást. Jelen projektben hálózati forgalomfigyelőnek használom. Ehhez telepítettem rá egy Ntopng nevű alkalmazást.

- OS: Ubuntu 18.04 LTS
- Ntopng+SSH szerver+TICK Stack
- 32GB SSD
- 4GB RAM
- dual core N4000 CPU
- 1db 1Gbps Ethernet csatlakozás
- 3db USB port

6.3.2 Netgear GS105eV2

Ez egy egyszerű 5 portos nem menedzselhető switch. De van egy számomra igen fontos tulajdonsága, hogy képes port tükrözésre. Így egy kis konfiguráció útján be lehet kötni két eszköz közé és be lehet állítani, hogy a két bekötött portjának a forgalmát áttükrözze egy 3. portra. Melyre rácsatlakozva a megfelelő szoftverekkel lehet hálózati analízist végezni.

6.4 Egyéb eszközök

A teljesség igénye nélkül. Jelenleg kb. 30 eszköz található a háztartásban, melyek szinte kivétel nélkül vezeték nélküli kapcsolaton csatlakoznak az otthoni hálózathoz. Ezt folyamatosan bővítem különböző IoT eszközökkel, szenzorokkal, kapcsolókkal, stb.

A szakdolgozat 4-es fejezetében szenteltem egy kis helyet a jelen IoT technológiák bemutatásának és véleményt is formáltam, hogy a direkt IoT-re kitalált technológiák közül én melyiket választom, de végül csak a jól bevált, régi WiFi-nél maradtam és nem ruháztam be egy Zigbee gateway-be, vagy bridge-be, mert egyszerűen nem éreztem szükségét a váltásnak. Ez annak is köszönhető, hogy megtaláltam a Shelly családot, ami tökéletesen bevált, megbízhatóság és szolgáltatások terén is.

Egy nagyon népszerű termékcsaládról van szó és talán mindezért a különböző integrációk, kompatibilitásaik is nagyon jók. Pl. a választott okos otthon rendszerhez (HA) is nagyon könnyen illeszthetőek. Csak felsorolás jelleggel, az eszközpark:

- Windows, linux munkaállomások
- IOS, android mobiltelefonok
- Kültéri, beltéri kamerák, robotporszívó
- Amazon Echo Dot hangszórók, személyi asszisztensek
- Android TV, Android media player, erősítő, Xbox
- Shelly-k
    - relék, okosdugaljak, motorvezérlők, dimmerek. mozgásérzékelők, stb.

7. Részletes specifikáció, a tervezés során végzett munkafázisok és tapasztalataik leírása

Egy átfogó képet szeretnék arról, hogy mi történik az otthoni hálózaton. Nagyságrendileg 30 db eszköz ül rajta, de ez folyamatosan bővül. Azt a döntést hoztam, hogy egyelőre nem lépek a zigbee vagy a z-wave technológiák irányába, mert a WiFi is tökéletesen kiszolgál.

Mindezt szeretném megvalósítani a legköltséghatékonyabban.

7.1 Tervek

7.1.1 Okosotthon központ (Home Assistant, HA)

Telepíteni kell egy okosotthon központot. A cél az, hogy minden működjön együtt vele és programozási ismeretek nélkül is képes legyek üzemeltetni az infrastruktúrát. 4-féle verzióban elérhető, így a használat során előfordulhat, hogy verziót kell majd váltani, ha valami nem vagy nem megfelelően működik.

A hálózat 'szíve' egy Qnap NAS, mely rendelkezik Konténer és Virtualizációs megoldással, ezeket ki szeretném használni. Ha valami elfuttatható konténerben, akkor az a legkevésbé erőforrásigényes megoldás, így mindenképp szeretném ezzel kezdeni. Ha valamiért nem megfelelő, akkor jöhet a virtualizálás. Nincs tapasztalatom a QNAP ezen megoldásaival, így elképzelhető, hogy fog meglepetés érni.

Szeretném, ha a központ tudna logot gyűjteni és tovább küldeni az adatbázisnak. Így egy központosított adatforrásom lenne.

7.1.2 Hálózati átalakítás, bővítés

Amíg a vezeték nélküli méréseket bárhonnan végezhetjük, addig a vezetékes hálózatban ez nem olyan egyszerű. Én szeretnék egy mérőeszközzel beleállni a hálózati forgalomba és csomagszinten figyelni. Ki, kivel, hova beszél. Kik látszódnak a hálózaton. Itt jó lenne, ha találnék olyan pontot, ahol minden forgalmat látok. Ehhez meg kell találni ezt a pontot és kell hozzá egy megfelelő eszköz, hogy tudjak port tükrözést csinálni.

Majd, ha megvan a tükrözött forgalom, akkor ezt az adatfolyamot valamivel meg kell etetni. Szeretnék egy 7/24-es mérőeszközt illeszteni rá és Wireshark-al is szeretnék ránézni.

7.1.3 Adatbázis, megjelenítés

Kell egy adatbázis, ami tárolja az adatokat és szükséges egy megjelenítési réteg, amin keresztül ezt látványosan meg lehet jeleníteni. A kapcsolók, szenzorok és pl. a klíma nem képes magáról logokat küldeni. Valahogy mégis szeretnék ezekhez az adatokhoz hozzáférni. Egy ügyes kis megoldás lehet, ha API oldalról támadok és úgy kérem le a szükséges adatokat, ha már önmagától nem képesek ezeket a rendelkezésemre bocsátani.

Mindez számomra teljesen új terület, így biztosan sok kihívás és sok újrakezdés vár rám.

7.1.4 Biztonság, mentés

Hálózati, fizikai, adatmentéssel és tárolással specifikus biztonság. Sokféle van és mindegyik fontos és szorosan kéz a kézben kell járjanak. Így amennyire lehet, ezt szem előtt tartva szeretném az egész projektet megvalósítani.

Kezdjük kívülről. A LAN hálózat csak VPN alól legyen elérhető. Ehhez szükséges egy megfelelő konfigurációval ellátott VPN szerver beállítása. Az elmúlt 1-2 év igen hangos a különböző ransomware támadásoktól és sérülékenységek kihasználásától. Valamiért a Qnap nasokat pedig kiemelt törődéssel kezelik a támadók. Nem szeretnék nyitott ajtókat hagyni, így a gyártók felhő alapú megoldásait is szeretném kerülni. Többször olvastam a szaksajtóban, hogy ezen talált 0-day sérülékenységet kihasználva jutottak be és titkosítottak le mindent.

Biztonságos kommunikáció használata. Mindem webes tartalmat https-re szeretnék terelni. FTP helyett SFTP használata és az alapértelmezett portok kerülése.

Az első védelmi vonalam a PPPoE router, így ott minden képességet be kell kapcsolni, hogy megfelelően védve legyen a helyi hálózat. Csak a szükséges port továbbítás használata. IDS, IPS aktiválása.

A vezeték nélküli hálózatokat (3 sávos WiFi) megfelelően erős titkosítással és jelszóvédelemmel ellátni. WDS, WPS kikapcsolása.

Eszközök, szoftverek konfigurációjának mentése, duplikált tárolása. 2 NAS van, az egyikben 4 HDD van Raid 5-ben, a másikban 2 db Raid 1-ben, így az alap adatbiztonság úgy gondolom megvan.

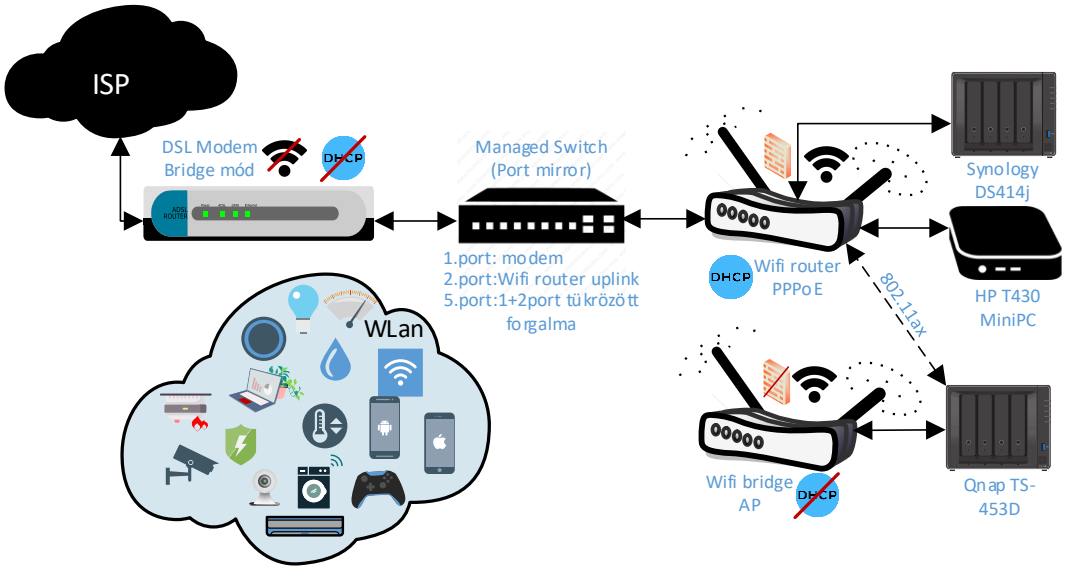
Ide sorolnám még a fizikai biztonságot is. Mivel ez nem egy adatközpont, hanem egy családi ház, egy otthon, így annyiban ez ki tud merülni, hogy fel van szerelve egy ’buta’ riasztó rendszer és két kamera. A kamerák képesek mozgás detektálásra és az aktivált időben push üzenetben való értesítésre.

8. A megvalósítás leírása

Szükség volt a hálózat átalakítására, bővítésére, hogy a célokat el tudjam érni. Be kellett szereznem egy menedzselhető, port tükrözésre képes switchet, mellyel be tudtam ékelődni a hálózatba. A vezeték nélküli hálózat lefedettségének bővítése és a Qnap NAS virtualizációs platformjának megoldása miatt szükség volt egy WiFi bridge AP-ra, mellyel a vezeték nélkül kommunikáló NAS egy fizikai hálózati portját fel tudtam kötni egy UTP kábel segítségével a hálózatra. Ehhez a router gyári firmware-t le kellett cserélnem egy dd-wrt-re és fel kellett konfigurálni, hogy ugyanazt a hálózatot adja, mint a fő router.

Nem volt része a hálózatnak, de szükség volt egy miniPC beállítására. Erre kellett egy operációs rendszer és erre került telepítésre az ntopng forgalom elemző eszköz.

8.1 Otthoni hálózati topológia

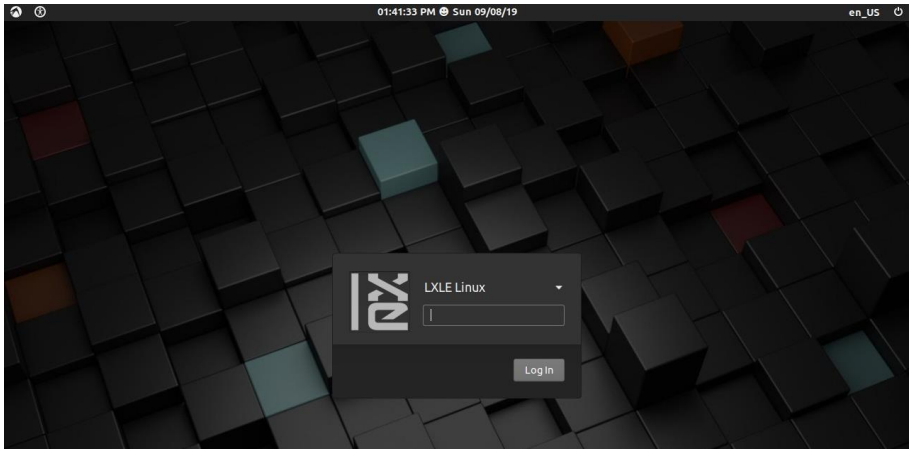


8.1. ábra: Hálózati Topológia

8.2 Ntopng+LXLE Linux+SSH szerver+TICK Stack

8.2.1 LXLE OS

Kellett egy nem erőforrásigényes OS. Nekem a LXLE linux disztribúcióra esett a választásom (https://lxle.net/). Ez egy Ubuntu-ra épülő operációs rendszer, mely az LTS (Long Time Support) kiadásokkal jelenik meg. Ez azt jelenti, hogy a frissítéseket patch-eket 5 évig megkapja. A telepítés egyszerű, le kell tölteni a 64-bites verziót (LXLE letöltés), ezt kiírni egy pendrive-ra (pl. https://rufus.ie/hu/ -al). Majd beállítani az USB boot-ot a HP T430-on és indul a telepítés. Erről nem írnék részletesen, gyakorlatilag módosítás nélkül, alap beállításokkal fel kell telepíteni a linuxot. Természetesen a felhasználókat kellően bonyolult jelszavakkal ellátni és egy újraindulás után egy hasonló felület fogad. Belépés után frissítettem az operációs rendszert.



8.2. ábra: LXLE kezdőképernyő

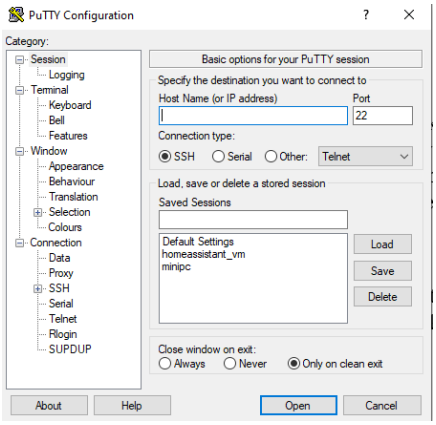
8.2.2 SSH szerver

Szeretném elérni az eszközt ssh-n, így kezdésként telepítettem rá egy ssh szervert. Ehhez egy terminalt nyitottam és kiadtam a következő parancsot: *sudo apt install openssh-sever*. Majd a /etc/ssh/sshd_conf állományban lehet mindent beállítani. Nekem most ezt nem nagyon kell módosítanom egyelőre, így csak annyit teszek, hogy ne lehessen root-ként belépni. Ezt a következő sor beillesztésével lehet elérni: *PermitRootLogin no*. A linux beépített tűzfalához kivételként hozzá kell adni az ssh szerver portját, hogy el lehessen érni.

- *ufw allow ssh*

8.2.3 Ntopng

Majd jöhet az ntopng. Ehhez az előzőekben feltelepített ssh szerveren keresztül már egy másik gépről el tudtam érni az eszközt. Ehhez kellett egy putty (https://www.putty.org/).



8.3. ábra: putty felület

A Host Name-hez be kell írni az IP-t, majd mivel a portját nem változtattam meg az ssh szervernek, így a 22-es maradhat. Belépve, egy *sudo -s* és a megfelelő jelszó beírása után az ntopng hivatalos telepítési leírását követve feltelepítettem.

- *wget http://apt.ntop.org/18.04/all/apt-ntop.deb*
- *dpkg -i apt-ntop.deb*
- *apt update*
- *apt install pfring-dkms nprobe ntopng n2disk cento*

Webes interface-e van, így a linux beépített tűzfalához hozzá kell adni az ntopng portját.

- *ufw allow 3005*

Ezek után a konfigurációs állományát be kell állítani *(/etc/ntopng/ntopng.conf*). Itt kell definiálni néhány alapvető paramétert:

- interfészek nevei, amin hallgatózik
- ip tartomány, amit figyel
- port
- az ip címeket megpróbálja-e feloldani

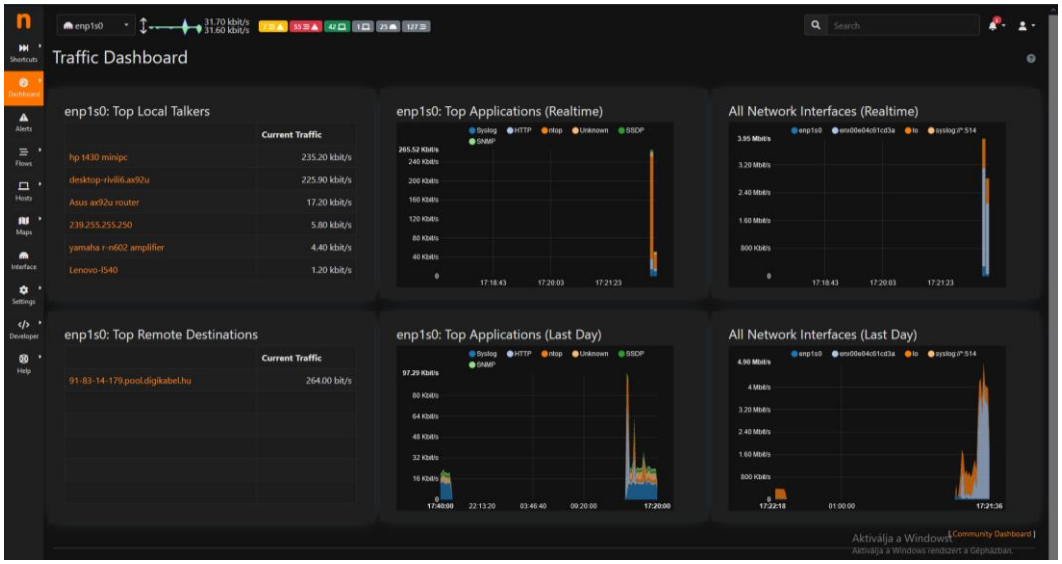
```
-i=enp1s0
-i=enx00e04c61cd3a
-i=syslog://*:514
# -i=eth2
-i=lo
#       -w|--http-port
#        Sets the HTTP port of the embedded web server.
#
-w=3005
#
#       -m|--local-networks
#        ntopng determines the ip addresses and netmasks for each active interfa$
#        those  networks  is considered local. This parameter allows the user to$
#        networks and subnetworks whose traffic is also considered local in  nto$
#        other hosts are considered remote. If not specified the default is set $
#
#        Commas  separate  multiple  network  values.  Both netmask and CIDR not$
#        even mixed together, for instance "131.114.21.0/24,10.0.0.0/255.0.0.0".
#
-m="10.[redacted]/24"
# -m=10.10.124.0/24
#
#       -n|--dns-mode
#        Sets the DNS address resolution mode: 0 - Decode DNS responses  and  re$
#        (-m)  numeric  IPs  1  -  Decode DNS responses and resolve all numeric $
#        responses and don't resolve numeric IPs 3 - Don't decode DNS responses $
#
-n=1
```

8.4. ábra: ntopng.conf

A konfigurációs állomány változtatása után szükséges az ntopng szervíz újraindításra.

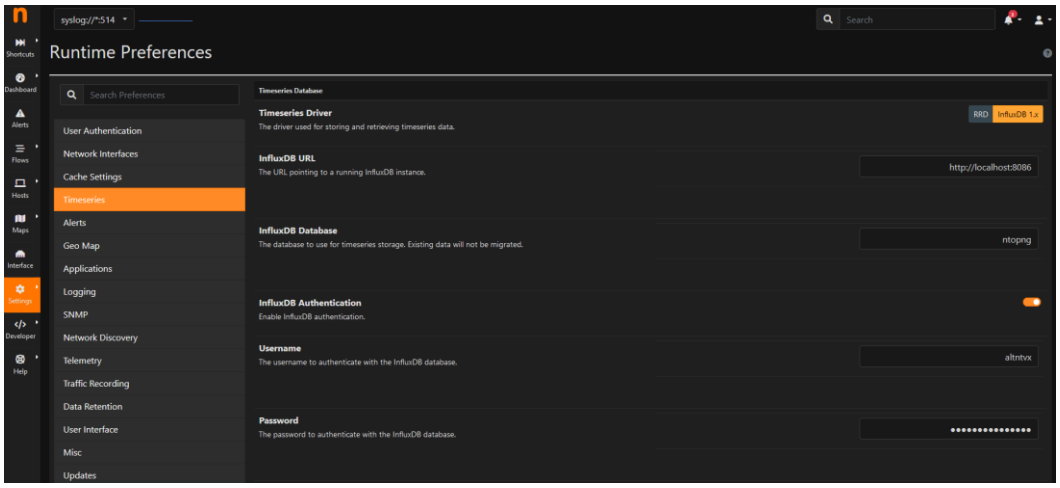
systemctl restart ntopng

Ezek után a 3005-ös porton elérhető a webes felület. Itt rögtön meg kell változtatnunk az alap admin/admin belépési jogosultságot. Majd egy hasonló felület fogad.



8.5. ábra: ntopng kezdő felület

A tárolt adatokat el tudja küldeni 1-es fő verziójú InfluxDB-nek, ezt be kell állítani.



8.6. ábra: ntopng influxdb integráció

8.2.4 TICK Stack

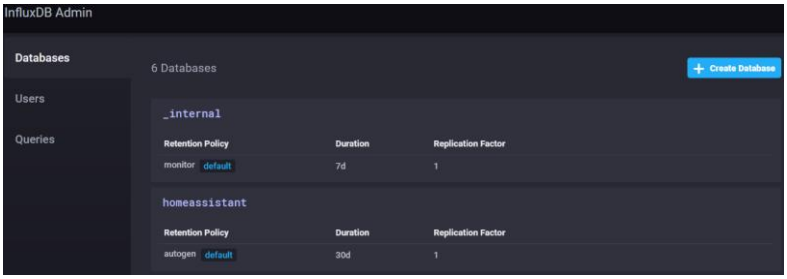
A 5.2.2 pontban bemutattam, hogy hogyan is épül fel ez a stack. Ezért ezeket az elemeket külön-külön kell telepíteni. Az InfluxDB nem olyan régen fő verziót váltott és az új verzióban egész más megközelítéssel tárolja és kezeli az adatokat, ott már van beépített webes felület és új lekérési nyelve is van. Először ezzel próbálkoztam, de több helyen is problémába ütköztem és mivel új, ezért a dokumentáltsága nem olyan részletes, mint az előző verzióé. Így az 1.8-as verziót választottam. A hivatalos dokumentáció szerinti telepítési útmutató szerint, először hozzá kell adni az Influxdata csomagtárat, majd jöhet az influxdb szerviz.

- *wget -qO- https://repos.influxdata.com/influxdb.key | gpg --dearmor > /etc/apt/trusted.gpg.d/influxdb.gpg*
- *export DISTRIB_ID=$(lsb_release -si); export DISTRIB_CODENAME=$(lsb_release -sc)*
- *echo "deb [signed-by=/etc/apt/trusted.gpg.d/influxdb.gpg] https://repos.influxdata.com/${DISTRIB_ID,,} ${DISTRIB_CODENAME} stable" > /etc/apt/sources.list.d/influxdb.list*
- *sudo apt-get update && sudo apt-get install influxdb*
- *sudo service influxdb start*

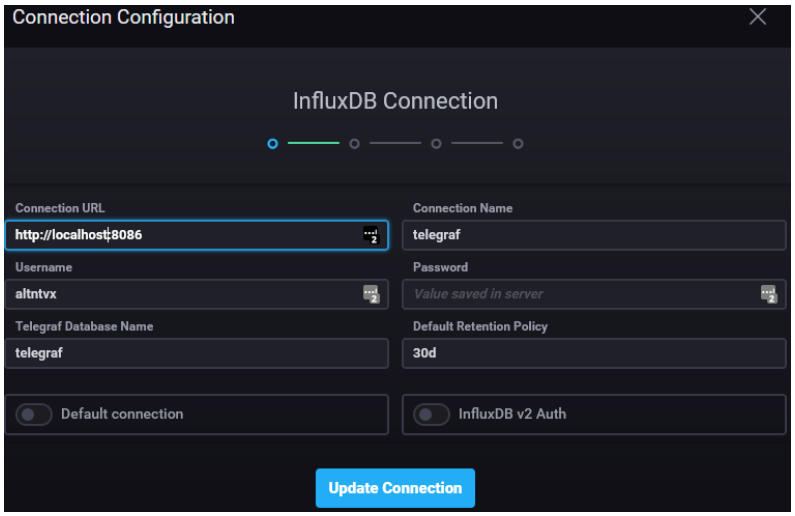
Az *influx* vagy a *systemctl status influxdb* paranccsal tudjuk ellenőrizni, hogy rendben elindult-e a szerviz. Majd jöhet a többi összetevő, a telegraf, a kapacitor és a chronograf:

- *wgethttps://dl.influxdata.com/chronograf/releases/chronograf_1.9.4_amd64.deb*
- *sudo dpkg -i chronograf_1.9.4_amd64.deb*
- *ufw allow 8888*
- *wget https://dl.influxdata.com/telegraf/releases/telegraf_1.22.0-1_amd64.deb*
- *sudo dpkg -i telegraf_1.22.0-1_amd64.deb*
- *wget https://dl.influxdata.com/kapacitor/releases/kapacitor_1.6.4-1_amd64.deb*
- *sudo dpkg -i kapacitor_1.6.4-1_amd64.deb*

A webes gui-t a chronograf adja, ehhez kellett megnyitni a 8888-as portot, hogy elérhető lehessen hálózaton belül böngészőből. Így már a felületen tudunk állítani sok mindent, felhasználók felvétele, adatbázisok létrehozása és itt tudjuk megmutatni a kapacitornak és a telegrafnak, hogy melyik adatbázisba dolgozzon.



8.7. ábra: chronograf adminisztráció



8.8. ábra: kapacitor kapcsolat

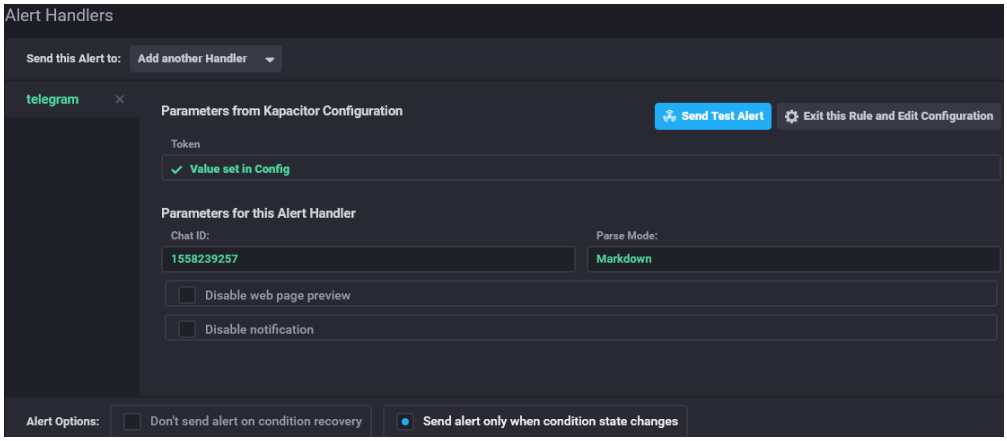
A telegraf a lelke az egész stack-nek, rajta keresztül lehet konfigurálni, hogy milyen adatokat várhat, miket küldjön, milyen url-re, melyik adatbázisba. A konfigurációs állomány 9677 sorból áll, így ezt nem másolom ide.

Valamilyen riasztási csatornát kellett keresnem, ahol egy beállított adatnál tud a TICK stack, azon belül is a kapacitor riasztani. A chronograf felületen hozzá lehet adni úgynevezett riasztás kezelőket a rendszerhez. Az e-mail-es értesítés mellett van sok más lehetőség, én most a telegram üzenetküldési lehetőségét választottam. A platformot amúgy is használom mobilon és számítógépen is, így kézenfekvő volt a választás.

Ahhoz, hogy a kapacitor küldeni tudjon riasztást, létre kell hozni egy telegram bot-ot.

a) A telegram-ot megnyitva beszélgetést kell kezdeményezni @BotFather-rel és be kell írni a */newbot* parancsot.
b) Ő megkérdi a leendő bot nevét, felhasználónevét, majd létrehoz egy egyedi tokent.
c) Viszont nekünk nem erre van szükségünk, hanem a létrehozott bot chat id-ra.
d) Ehhez egy másik beépített telegram bot-hoz kell fordulnunk. A @RawDataBot-al kell beszélgetést kezdeményeznünk, aki a létrehozott bot-hoz tartozó összes azonosítót elküldi, köztük a chat id-t is.

Majd ezt a chat id-t a chronograf Riasztás menüjében hozzá kell adni.



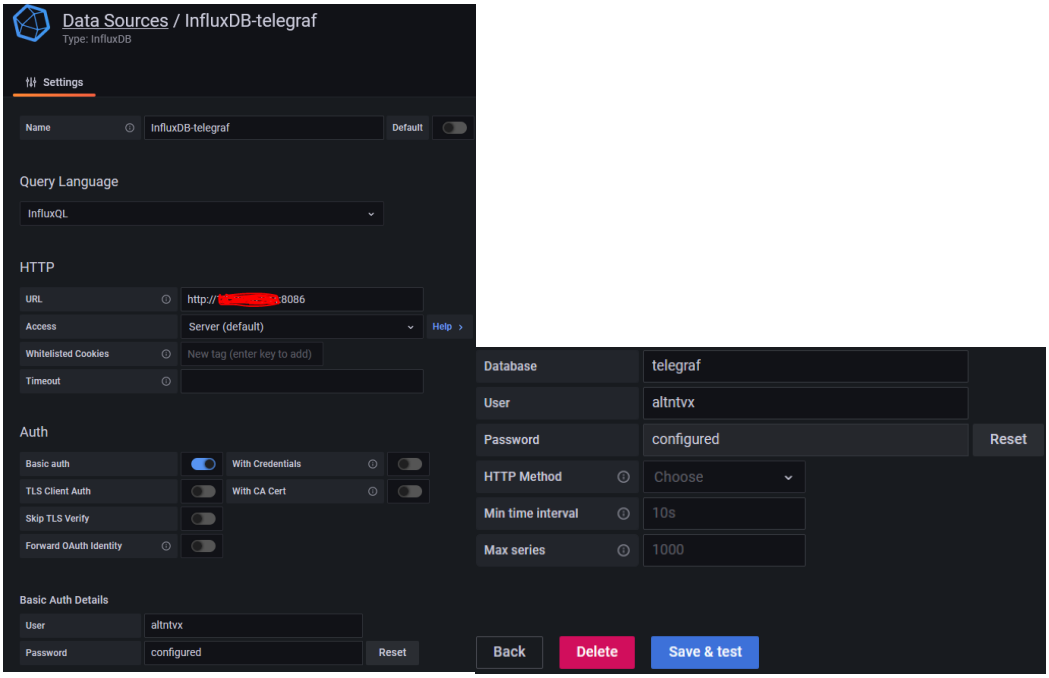
8.9. ábra: chronograf értesítés kezelő

Ezután csak ki kell választani a beállított riasztásnál a telegram-ot és riasztási eseményél jön az előre beállított üzenet.

8.3 Grafana

A grafana virtualizációs eszközt szerencsére natívan támogatja a Qnap, azaz felületről telepíthető. Ehhez csak hozzá kell adni az alkalmazás központjához a https://qnapclub.eu/en/repo.xml csomagtárat és megjelenik, mint elérhető áruház, ahonnan fel tudjuk telepíteni. A 3000-es porton hallgat, azaz a Nas IP címéről, ezen porton, böngészőből elérhetővé válik.

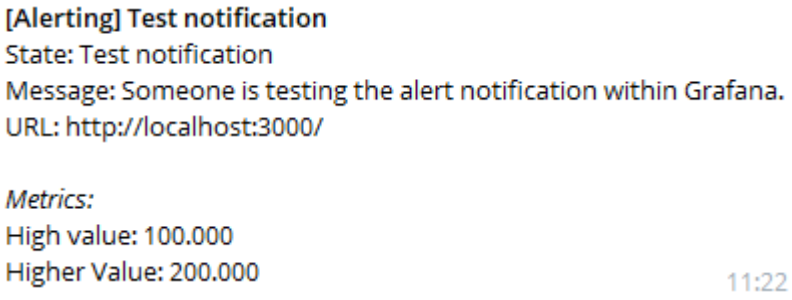
Az alap jelszó lecserélése után hozzá kell adni az adatforrásokat. Ha egy adatforrás több adatbázist kezel, akkor ezeket külön-külön.



8.10. ábra: Grafana adatforrás hozzáadás

Rengeteg előre gyártott dashboard-ja van, ami a grafana oldaláról letölthető és kézzel is létre tudunk hozni.

Az előző pontban létrehozott telegram bot-ot a Grafana-ban is tudjuk használni, mert támogatja ezt az eseménykezelőt. A riasztások menüben, az értesítési csatornáknál tudjuk hozzáadni. Itt meg kell adni az Api token-t és a chat id-t is és a végén érdemes ezt letesztelni, hogy működik-e.



8.11. ábra: grafana teszt riasztás

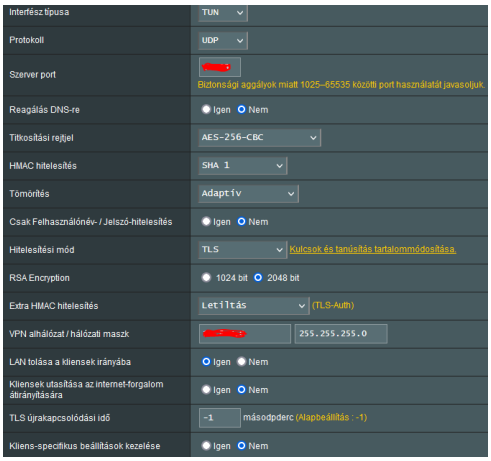
8.4 VPN (Virtual Private Network)

A VPN igen nagy népszerűségnek örvend manapság a home office, a kialakult vírushelyzet miatt. A Covid kikényszerítette szinte minden munkáltatóból, hogy a munkavállalói távolról tudjanak dolgozni. Ennek egy biztonságos módja, ha a céges hálózatot VPN csatornán keresztül érik el. Ebben az esetben a VPN felépülése után gyakorlatilag ugyanazt az IT infrastruktúrát tudja a munkavállaló használni, mintha benn ülne az irodában.

Ahhoz, hogy az otthoni hálózat ne legyen elérhető közvetlenül az internet felől, a támadások csökkentése végett, én is ehhez a technológiához fordultam. Beállítottam egy VPN szervert. A belépési pont a router, így a legkézenfekvőbb ennek az eszköznek a használata volt. Régebben is kísérleteztem saját VPN megoldással, mert gyakorlatilag a VPN szerver bárhol lehet. Akkor a régi Synology NAS-on állítottam be egyet, de ezt szerettem volna már régen lecserélni, mert az nem egy erőmű és hálózati szempontból sem a legmegfelelőbb helyen van.

Az Asus RT-AX92U router ismeri az PPTP, az IPSec és az OpenVPN megoldásokat. A PPTP kissé elavult, azt nem ajánlatos használni. Az IPSec-et pedig site to site VPN kapcsolatnál szokták használni. Így az OpenVPN-t választottam. Ezzel már volt tapasztalatom és szeretem a nyílt forrású megoldásokat és ez is az. Igaz ezt natívan nem támogatják az operációs rendszerek, szükség van alkalmazás telepítésére.

Egy csomó beállítási lehetősége van, pl. tud UDP-n, vagy TCP-n is működni, rengeteg titkosítási algoritmus állítható és meghatározható, hogy alhálózatot hozzon létre.



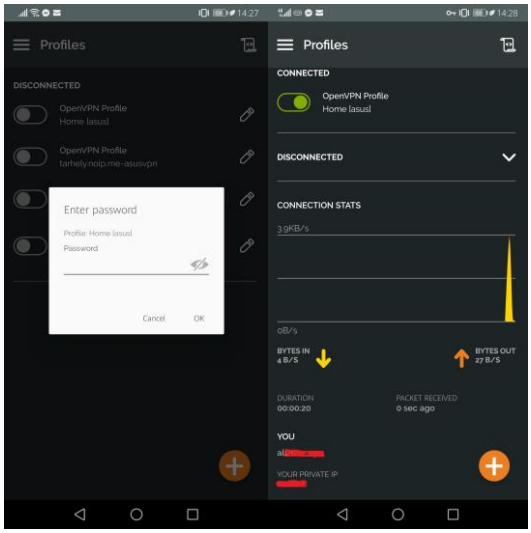
8.13. ábra: Openvpn konfiguráció

Továbbá a beállítások mentése után meg kell adni egy felhasználónév/jelszó párost, melyet a vpn kliens kérni fog a csatlakozáskor a kapcsolat kiépüléséhez.

A router felületén van lehetőség az openvpn konfigurációs fájl exportálására. Ez tartalmaz minden fenti beállítást, a létrehozott tanúsítványokat és a privát kulcsot is. Ezt a fájlt a lehető legbiztonságosabb módon el kell juttatni arra gépre, amellyel majd szeretném létrehozni a VPN kapcsolatot és meg kell mutatni, be kell másolni a megfelelő helyre a kliensnek, ezáltal ő fel tudja olvasni a beállításokat.

Mivel nincs statikus IP címem, így ez időnként változik. Viszont regisztráltam egy ingyenes DDNS (Dinamikus DNS) szolgáltatásra, így az otthoni hálózatnak tudtam adni egy nevet. Mivel a konfigurációs állomány első sorába az IP cím kerül, ezt át kellett kézzel írnom erre a névre, hogy ne kelljen minden egyes csatlakozáskor ezt módosítanom, mikor a szolgáltató lecseréli az IP címet.

Android operációs rendszerű telefonon fel kell telepíteni az ingyenes openvpn alkalmazást az áruházból és a jobb alsó ’+’ jellel hozzá kell adni az előzőekben létrehozott *.opvpn* fájlt, majd el kell indítani a csatlakozást. Kérni fogja a jelszót, majd a jobb oldali képernyőn látható, hogy a biztonságos kapcsolat kiépült.



8.14. ábra: openvpn csatlakozás

8.5 Hálózat biztonság

Ezeket a biztonsági beállításokat mindenkinek csak javasolni tudom. Ezt talán úgy lehetne a legérthetőbben összefoglalni, hogy ezekkel a beállításokkal nem hívogatjuk a támadókat és ha szét is néznének, akkor ha egy bicikli tárolót látnának számos kerékpárral, akkor a mi kerékpárunk jó minőségű lánccal lenne lekötve, egy vasgerendához. Nem lenne rajta a csengő, vagy a sebességmérő, amik egy mozdulattal levehetőek. Talán az első kerék is külön le lenne láncolva egy másik gerendához és nem úgy, mint pár másik mellette levő, ami lehet, hogy le van láncolva, de csak egy 2m-es oszlophoz, amin játszi könnyedséggel átemelhető a bringa, vagy ki lenne kötve egy masszív vasoszlophoz, de spagetti vastagságú lánccal.

- Alap SSID átírása olyanra, ami nem tartalmazza a gyártót vagy egyéb hasznos információt
- A vezeték nélküli hitelesítés minimum WPA2
- A WPA titkosítás AES
- A WPA kulcs megfelelően hosszú és bonyolult (min. 15 karakter, kis és nagybetű, szám és speciális karaktert is tartalmaz és nem értelmes szó, ami köthető lehetne hozzám)
- WPS (WiFi Protected Setup) kikapcsolása (Ez igen sérülékeny. Egy PIN kód megadásával lehet csatlakozni a hálózatra és ez brute force támadással igen könnyen kivitelezhető)
- WDS (Wireless Distribution System) kikapcsolása (Csak nyílt, vagy WEP hitelesítéssel működik, ez könnyen törhető)
- MAC szűrés bekapcsolása (Csak megadott MAC címmel rendelkező eszköz csatlakozhat a hálózatra)

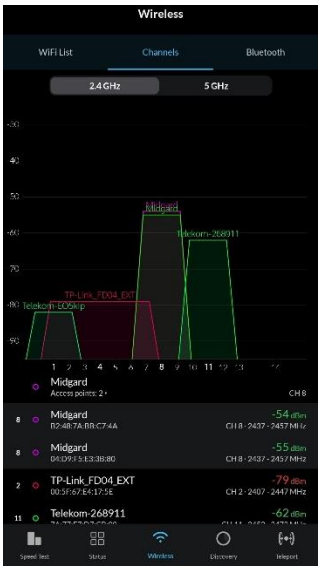
- Egzotikus IP tartomány használata (Ne 192.168.0.1/32 legyen a router címe, a kezdő, befejező IP címmel is lehet játszani)
- UPnP tiltása (Ez alapvetően biztonsági kockázat, nyílt portokon működik)
- Port továbbítás kerülése (Hacsak nem hosztolunk valamit, ami miatt ez fontos és elkerülhetetlen, akkor ne használjuk)
- IPv6 tiltása
- Tűzfal, DOS védelem bekapcsolása
- WAN IP ping tiltása
- Router firmware frissen tartása
- Telnet tiltása (Nem titkosított csatorna)
- SSH tiltása WAN-ról
- Helyi WebUI hozzáférés https-en
- Távoli hozzáférés tiltása

Plusz ezzel az Asus RT-AX92U routerrel járt egy örök licenszes két irányú IPS, rosszindulatú oldalak figyelése és fertőzött készülék ellenőrző funkció is, AiProtection néven. Ezt a szolgáltatást a Trend Micro-val karöltve nyújtja az Asus. A kezdetek óta be van kapcsolva és 2022. áprilisig több, mint 400 alkalommal észlelt és hárított el fenyegetést. Riasztás esetén e-mail értesítést küld.

8.15. ábra: Asus AiProtection

Végül, nem biztonsági beállítás, de érdemes erre is figyelni. A vezérlőcsatornát olyan helyre tenni, ami nem annyira telített. Így sokkal jobb vételi kapacitás lesz a csatlakozó klienseknek. Erre a WiFiman, Ubiquiti cég által fejlesztett ingyenes mobil alkalmazását javaslom. Egyszerűen használható, a WiFi csatornák és jelerősségük megmutatása mellett tartalmaz még jópár hasznos eszközt.

- port szkenner
- sebesség teszt
- automatikus eszköz felismerés (IP, MAC, gyártó)



8.16. ábra: WiFiman vezeték nélküli hálózat térkép

8.6 Home Assistant

Mivel ebből többféle verzió létezik, lehet futtatni konténerben, van saját OS-e és további két alternatív telepítési módja, így kísérleteztem.

Először feltelepítettem a Qnap NAS Container Station-ön keresztül. Ebben az esetben ez nem vett igénybe többet, mint 2 percet. Arra kellett figyelni, hogy a telepítés közben a HA konfigurációs mappáját ki kellett mountolni a konténerből, hogy módosításokat, kézzel felveendő szenzorokat hozzá lehessen adni a megfelelő *.yaml állományokhoz. Plusz fel kellett venni egy admin felhasználót a NAS-ra minden egyéb jogosultság nélkül. Ez azért kellett, hogy a HA hozzáférjen a QNAP-hez.

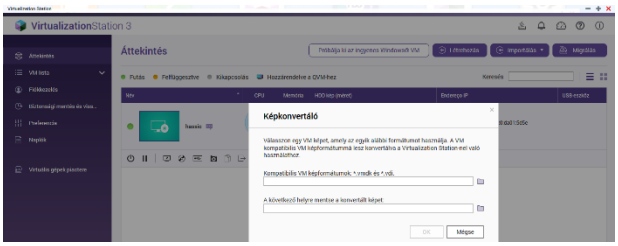
Ebben az esetben pár dologról le kell mondani, de cserébe igencsak erőforráshatékonyan üzemeltethető a HA.

- Nincs Supervisor mód/menü
- Nincsenek Add-on-ok
- Nehézkes a HA Backup
- A frissítés a konténer újratelepítésével oldható meg

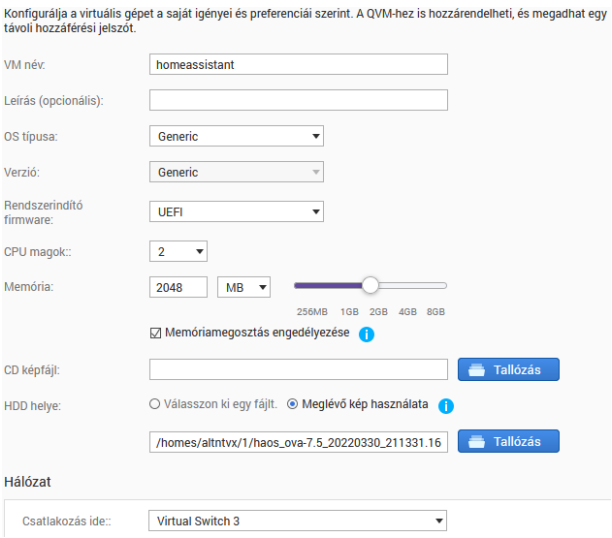
A mentés nem érdekes, mert magát a konténert viszont nagyon egyszerű menteni, kis helyet foglal, azaz könnyű a mentés menedzsment. A másik kettő fontossága csak használat közben derült ki és emiatt sajnos nem is tudtam ezt a megoldást hosszútávon választani.

Kisebb kutatás után kiderült, hogy az InfluxDB és a Grafana is Add-on-ként telepíthető elméletben. Hogy bizonyos integrációkhoz szükséges a *supervisor* mód és a frissítés is sokkal könnyebbé válhat. Cserébe viszont több erőforrást igényel. Így a QNAP Virtualizációs platformjához (Virtualization Station 3, VS3) fordultam.

A HA rendelkezik igen sok telepítési image-el, a teljesség igénye nélkül van Windows, Linux, Mac, Raspberry Pi, KVM, VmWare, Virtualbox, stb. De nincs Qnap NAS. Ebben az esetben ki kellett találni, hogy mit használjak. A HA letöltési oldaláról letöltöttem a VmWare és a Virtualbox fájlokat. Ezeket a VS3 felületén a beépített képkonvertálóval át tudtam alakítani *.img* kiterjesztésre. A *.vmdk* (vmware)-ből kreált valamiért nem működött, az image nem boot-olt fel, viszont a *.vdi* (virtualbox) igen.



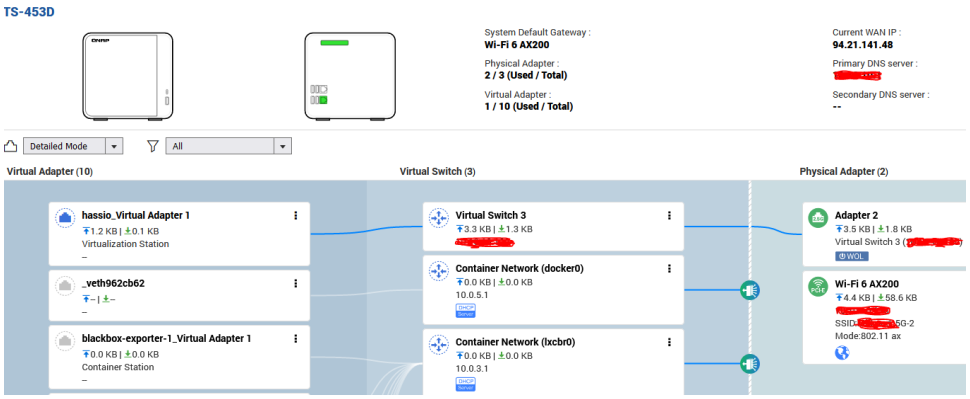
8.17. ábra: VS3 képkonvertálás



8.18. ábra: HA virtuális gép létrehozás

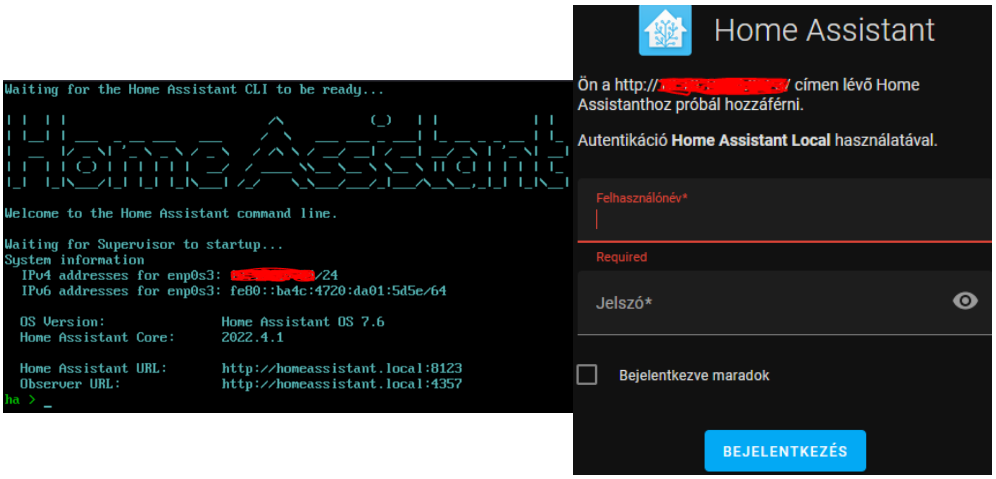
Csak UEFI módban volt hajlandó bebootolni. A default opció az ’örökölt BIOS’ volt. Ezzel nem indult el. Plusz fontos beállítás, hogy az interface-e hova csatlakozzon. A virtuális gép telepítése előtt létrehoztam egy virtuális switch-et, ami kapott egy fix IP-t az otthoni tartományból. A switch-et NAT-olás és DHCP nélkül bekötöttem virtuálisan

arra a fizikai adapterre, ami be van kötve a TP-Link WiFi bridge AP-be. Ez a Virtual Switch 3. Így ezen az interface-en keresztül elérhető a HA.



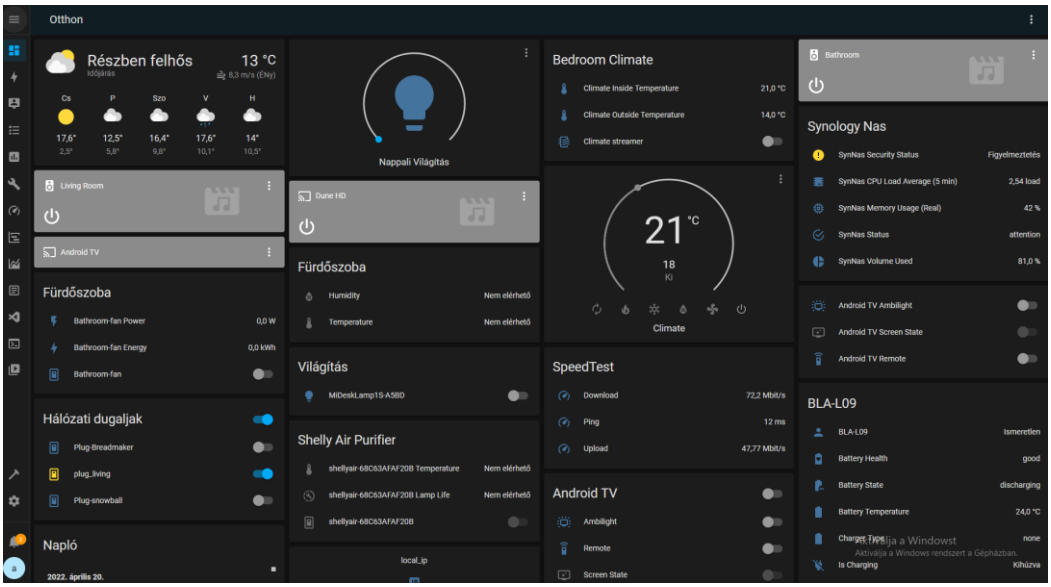
8.19. ábra: Qnap NAS virtuális hálózat

Majd pár perc múlva elindult a Home Assistant Operációs Rendszer (HASSIO) és elérhetővé vált a webes felülete.



8.20. ábra: HA kezdőképernyők

Van beépített felfedező funkciója, így egy csomó integrációt alapból felismert (Android TV, Shelly eszközök, Yamaha erősítők, stb.). De jópárat nem (Xbox, Irobot robotporszívó, néhány Shelly szenzor, NAS-ok, stb). Ezeket a felületről lehet konfigurálni, amire létezik gyári integráció. Amire nem, azt kézzel a konfigurációs állományokon keresztül (pl. Qnap NAS).



8.21. ábra: HA kezelőfelület

A felületen keresztül lehet hozzáférni a bővítményekhez is természetesen. Pl. ahhoz, hogy a virtuális gépet el tudjam érni ssh-n keresztül és ne kelljen emiatt mindig belépni a NAS VS3 konzoljára, fel kellett installálni egy [SSH&Web terminal] nevű ssh szervert. Így néz ki egy bővítmény konfigurációja:

8.22. ábra: InfluxDB HA bővítmény

Be lehet állítani, hogy a rendszer indulásakor elinduljon, hogy az oldalsávra, egy klikkre kikerüljön az automatikus frissítést, le lehet állítani, törölni, újraindítani. Lehet szerkeszteni a konfigurációs állományát. Teljes körűen vezérelhető a webes felületről.

9. Az eredmények bemutatása, tesztelés, elemzés

9.1 TICK Stack loggyűjtés

A telegraf konfigurációs állományában (*/etc/telegraf/telegraf.conf*) kell beállítani a logforrást, amivel igencsak meggyűlt a bajom a syslog RFC szabványok és a protokollok miatt. RFC5424 és RFC3164 definiálható és képes tcp-n és udp-n fogadni. Viszont több órányi próbálkozás során sem akart megjelenni a log. Bellállítottam több logküldő forrást is, melyek képesek syslog szabvány szerint logot küldeni. A két NAS, a router, egy tasmota fw-el rendelkező redőnymotor, de nem érkezett be a log.

A hibakeresés számomra igen tanulságos volt, így ezt részletesen is bemutatom. A konfigja szerint így adható meg:

- *server = "tcp://localhost:6514"*
- *server = "udp://:6514"*

A konfigurációs állomány nem tér ki rá, hogy itt milyen IP-t, host-ot kell definiálni. Én sokáig a forrásokat adtam itt meg, ami persze hiba volt és ez a hiba a tapasztalatlanságomból fakad.

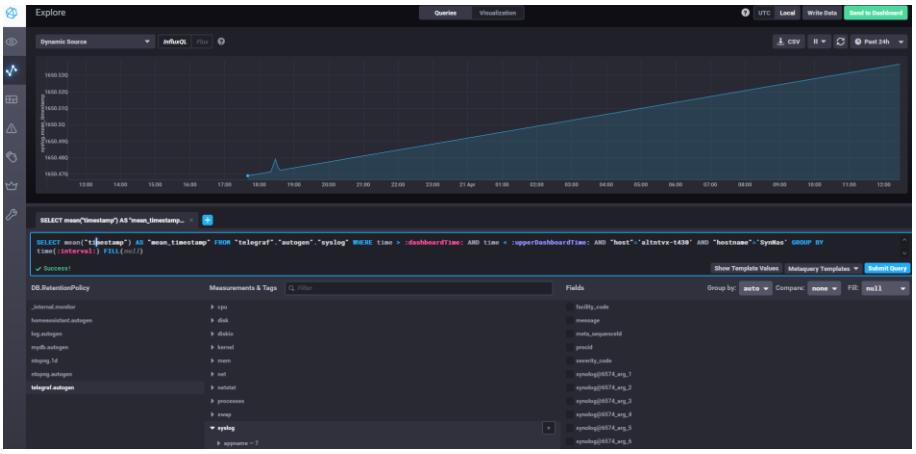
Nem lehet több sort megadni, miért is lehetne, hisz itt nem a távoli host-ot állítom be. A konfigurációs állomány módosítása után mindig újra kell indítani a telegraf szervízt (*systemctl restart telegraf*). Majd, ha ez megvan, érdemes ránézni, hogy valóban elindult-e, aktív, vagy inaktív, ugyanis, ha elírást, vagy szintaktikai hibát vétek az állomány módosításakor, akkor nem fog elindulni. Ezért is és mindenkori jó tanács, hogy mielőtt elkezdjük módosítani az alapértelmezett, működő konfigurációs állományt, csináljunk róla egy másolatot, amihez vissza tudunk térni.

Volt olyan logforrás, aminél nem lehetett beállítani a küldést csak abban az esetben, ha nem localhost volt megadva, hanem a valós IP és az sem volt egzaktul kifejtve a leírásban, hogy a telegraf nem tud eltérni az általa preferált 6514-es porttól, különben hibára fut induláskor. Pedig a syslog alapértelmezett portja nem a 6514, hanem az 514.

Miután rájöttem, hogy milyen formában kell megadni ezt az egy sort és mégsem történt semmi, tovább gondolkodtam, hogy mi lehet még a baj. Nézzük meg, hogy nyitva van-e a port és melyik program figyel rajta (*netstat -tulpn*). Ha nincs nyitva a port, a program sem tud rajta kihallgatni. Azaz hozzá kell adni a tűzfalhoz (*ufw allow 6514*). Miután látszott, hogy a telegraf látszik a 6514-es porton és még mindig nem jó, jöhetett a *tcpdump*, amivel lehet nézni, hogy adott hostról érkezik-e kérés, csomag. Ezt a *grep*-el szépen lehet szűrni, pl. a 6514-es portra.

A webes felületen továbbra sem jelent meg semmi. Van külön Log Viewer menüje, ami folyamatosan keresi a megadott táblát és az Explore menüben is tudunk rá kézi

keresést indítani. Úgy működik a felület, hogy ezeket a lekéréseket össze lehessen kattintani, vagy haladók kézzel meg tudják írni.



9.1. ábra: chronograf lekérési felület

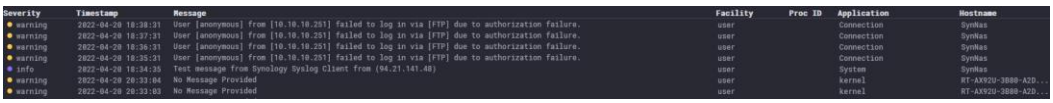
Itt látszik is, hogy már benne van a syslog plugin, de ekkor ez még nem jelent meg, hiába frissítettem rá többször. Mindezt természetesen képesek vagyunk az OS konzolon is lekérdezni.

```
root@altntvx-t430:~# influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> use telegraf
Using database telegraf
> show measurements
name: measurements
name
----
cpu
disk
diskio
kernel
mem
net
netstat
processes
swap
syslog
system
> select * from syslog
```

9.2. ábra: influxdb terminál

Itt is látszik a syslog, de ekkor természetesen még nem látszódott. Vissza a konfigurációs állományhoz. Kiderült, hogy a syslog plugin alapértelmezetten nincs bekapcsolva a *telegraf* konfigban, miért is lenne. Miután módosítottam, nagy reményekkel fordultam a terminálhoz, végre kész, látszik, hogy nyitva a port, hogy hallgatózik rajta a telegraf, a *tcpdump* szerint is jönnek a csomagok, jó a konfigurációs állomány. De nem jártam sikerrel. Nem értettem, hogy a többi plugin mellett miért nem jelenik meg a syslog is.

Nézzük meg a logküldőt. Egészen eddig az Asus routerrel próbálkoztam, ennek a felületén annyit lehet megadni, hogy IP/port. Ebben hiba nem lehet, így ráküldtem a nehéztüzérséget és minden syslogot küldeni képes eszközömet ide irányítottam. Ez nyerő lépésnek bizonyult, ugyanis kiderült, hogy a router udp-n küldi, én pedig tcp-n várom. Az is kiderült, ami szintén a tapasztalatlanságom rovására írható, hogy addig nem jön létre az adatbázisban egy új mező, amíg nem jön be rá érték. Ezek után a konzolon is szépen megjelent és a *select * from syslog* kérés is behozta a logokat és a felületen is gyönyörűen láthatóvá váltak.



9.3. ábra: chronograf lognéző

9.2 HA loggyűjtés

Az összes beintegrált eszközt, szenzort, kapcsolót a HA kezel, így nagy bizalommal fordultam a HA OS felé logok tekintetében. A webes felületen a beállítások között van egy napló menü, ahol hasznos infókat mutat, de ennyi. Nem lehet exportálni és elküldeni se egy távoli loggyűjtőnek a felületről.

Kiterjedt közösség csoportosult a HA fejlesztésbe az évek alatt, egyértelműen ez a legnépszerűbb okosotthon központ manapság, így körbenéztem, hogy van-e erre megoldás. A kérés nem csak bennem fogalmazódott már meg. Találtam is rá egy kerülő megoldást.

OS szinten nem lehet hozzáférni a logokhoz, ezek el vannak zárva a felhasználótól. Viszont találtam két bővítményt, amivel ez mégis megoldható.

9.2.1 Grafana&Grafana Loki kliens, Promtail

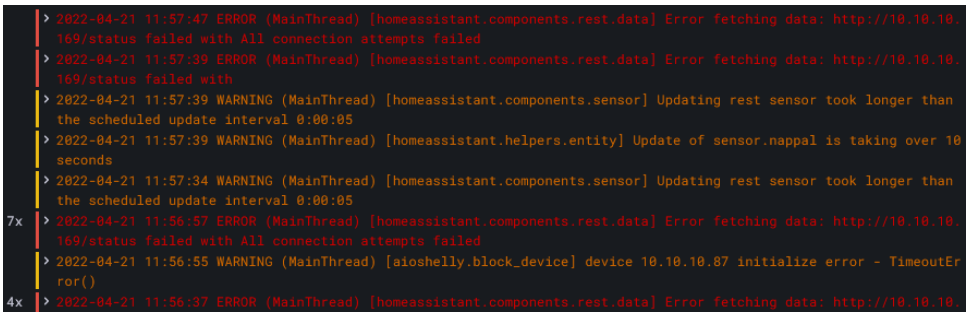
A Grafana-t mindenki a vizualizációval köti össze, pedig van még pár apróbb termékük, amik természetesen össze vannak integrálva a fő termékükkel. A Loki is ilyen. Ez egy a logolásra kihegyezett stack, adatbázis. Csak a logok cimkéit indexeli, a log tartalmát tömöríti, így kiemelten hatékonyan tud működni.

A Loki-val tehát le tudom tárolni a logokat és ezt összekötve a Grafana-val meg tudom jeleníteni, de még mindig nincsenek logok. Szerencsére a Grafana mérnökei erre is gondoltak és megírták a Promtail klienst. Ezt telepítve képes felolvasni minden logot a *systemd journal*-ból. A HA minden bővítményét, a HA saját logját, az aktív entitások logját és magáról a host-ról is.

Mindehhez a HA bővítményboltjához hozzá kellett adni a https://github.com/mdegat01/hassio-addons, csomagtárat és felinstallálni a két eszközt. Installálás után a Promtail konfigurálható a HA felületén, hogy hova küldje a logokat és milyen szintű logokat küldjön, végül mindkét csomagot el kell indítani.

- *client:*
- *url: http://39bd2704-loki:3100/loki/api/v1/push*
- *log_level: info*

Majd a Grafana felületén a tanult módon hozzá kell adni a Loki adatforrást. Ezek után már létre tudunk hozni egy dashboard-ot és be tudjuk állítani a kívánt log elemeket.



9.4. ábra: HA logok egy Grafana dashboard-on

Innentől kezdve mindenről tudunk, amiről a HA tud.

9.3 Ntopng

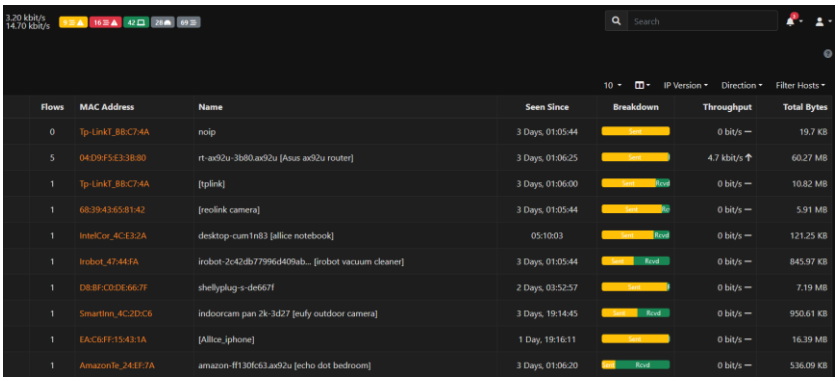
Ahogy a megvalósításban bekonfiguráltam az eszközt, onnantól kezdve teszi a dolgát, monitoroz, analizál. A megadott IP tartományban, a beállított interface-eken (max. 8db interface engedélyezett az ingyenes licensszel), az nDPI segítségével feltérképezi a host-okat, az aktív folyamokat, protokollokat.

Tárol hisztorikus adatokat az egész hálózatról és a hosztokról is. Hozzáadható kivétellista, hogy mit ne figyeljen, milyen IP-ket, tartományt, protokollt. Az SNMP képes eszközöket bekonfigurálhatjuk a felületen és azokról is átfogó képet tud adni hálózati szempontból. Figyeli a gyanús protokollokat, ha egy forgalom, nem az alapértelmezett porton halad. Ha titkosítatlan hozzáférést észlel, vagy egy hoszton megugrik a forgalom.

A korábban létrehozott telegram bot-ot itt is fel tudjuk venni, mint esemény értesítési címzettet. Nagyon részletesen beállítható a riasztás. Mely hosztokra, létrehozott hoszt csoportokra, esemény szintekre, milyen moduljaira kérünk értesítést.

Rendkívül barátságos és igen jól paraméterezhető a kezelőfelületen keresztül.

Felderített hosztok:

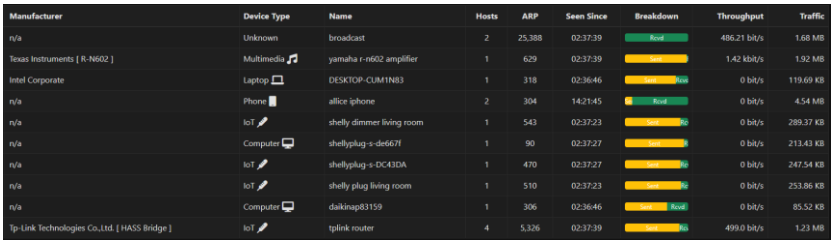


9.5. ábra: ntopng felderített hosztok

Legtöbbet beszélők:

9.6. ábra: ntopng dashboard

Eszközlista:



9.7. ábra: ntopng MAC lista szint (MAC levágva)

Egy apró érdekesség, amit csak véletlen vettem észre, de igencsak hasznos lehet. A termék bármelyik licenszét, a fejlesztő, ingyenesen használatra bocsájtja bárkinek, aki oktatási célokra használja. Ennek utána jártam a gyártónál és pár óra alatt küldtek számomra egy egy éves Enterprise M licenszt, miután megírtam nekik, hogy tanulmányaimat hol folytatom és milyen célra szeretném használni. Csupán annyit kellett tennem, hogy ezt az egyetemi e-mail címről kérjem. Még a szoftver licenszelési leírását is elküldték, ami amúgy sem egy bonyolult dolog. 1000€/év költségű szoftverről van szó.

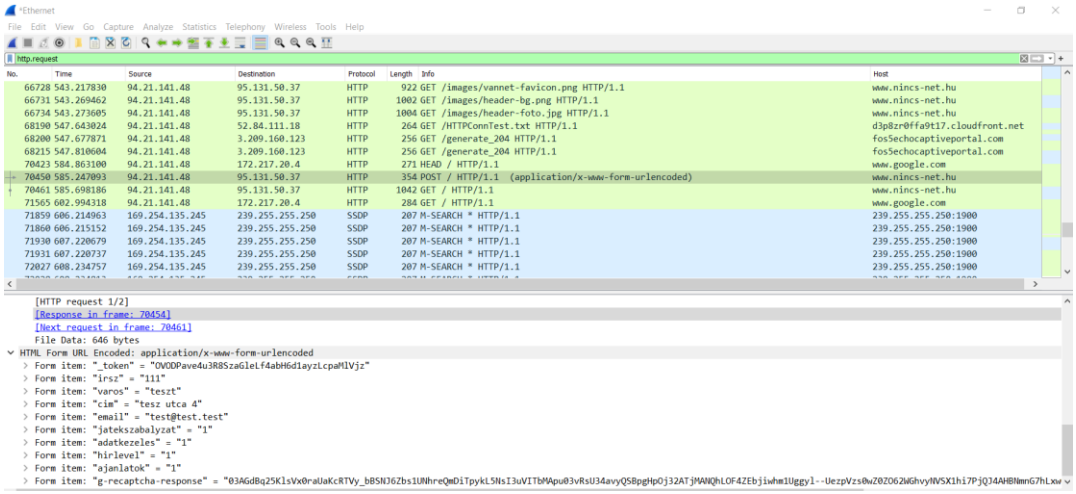
9.4 Forgalom, csomag elemzés

A vezetékes és a vezeték nélküli hálózaton is végeztem elemzéseket.

9.4.1 Vezetékes forgalom - port tükrözött forgalom

Amire először kíváncsi voltam, hogy mi látszik a tükrözött porton, megy-e kifelé titkosítatlan forgalom, amibe bele tudok nézni. Ehhez a Wireshark programot használtam. A telepítése igen egyszerű, windows-okon a hivatalos honlapon a letöltést követően tovább, tovább, befejez. Teljesen multi platform, elfut MAC OS-en, Linux-okon is.

Csináltam egy tesztet. Nehéz, de még lehet találni http oldalt. A nincs-net.hu egy valódi oldal, legalábbis annak látszik, ahol olyan helyeket lehet megadni az országban, ahol nincs internet elérés. A Wireshark-el szépen látszanak a megadott adatok.



9.8. ábra: Wireshark, http://nincs-net.hu teszt

Majd azt figyeltem meg, hogy milyen forgalom látszik sima szövegben, ami a hálózatomat elhagyja. Szépen jöttek a találatok. Látszódott, hogy a facebook ránézett a mobilom státuszára és volt egy pár amazonos host is. Mivel több amazon személyi kliens (Alexa) található a háztartásban, így ez is rendben van, nem volt benne semmi érdekes. Ami érdekes volt és nem is igazán tudom hova tenni, hogy a Dune média lejátszóm miért mond el magáról egy csomó mindent a 172.217.20.4-nek.

9.9. ábra: Wireshark, Dune

Az *nslookup* szerint ez a ham02s13-in-f4.1e100.net, aminek kicsit jobban utánanézve nem másba botlottam, mint a google-be, ki másba. 2009-ben indították ezt a 'szolgáltatásukat', hogy ezzel is jobbá és biztonságosabbá tegyék a hálózatot. Egészen biztosan nincs más céljuk ezzel.

9.4.2 Vezeték nélküli forgalom

A Kali Linux-ot választottam, hogy különböző támadásoknak, elemzéseknek vessem alá a saját vezeték nélküli hálózatomat, hogy megbizonyosodjak róla, hogy nem hagytam könnyen kihasználható sérülékenységet.

Ehhez egy nem használt laptopra feltelepítettem egy friss Kali linux-ot (https://cdimage.kali.org/kali-2022.1/kali-linux-2022.1-installer-amd64.iso), majd rádugtam egy külső usb vezeték nélküli adaptert. Az eszköz egy Alfa AWUS036NEH stick. Teljesen kompatibilis a Kali-val, képes monitor módban működni és csomag befecskendezésre is alkalmas.

A monitor módba kapcsoláshoz az *aircrack-ng* program családot használtam.

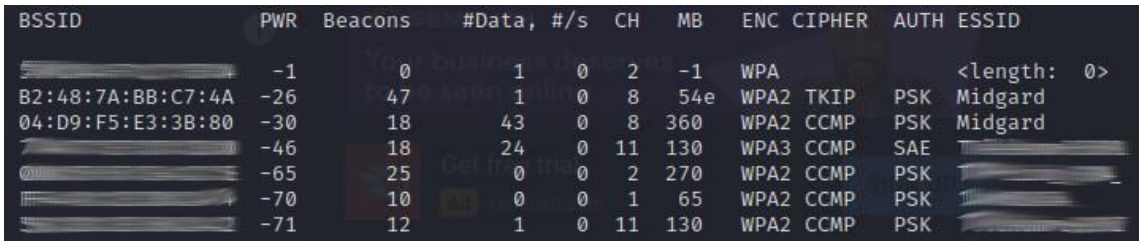
- *iwconfig*-al megnéztem a vezeték nélküli adaptereket és az aktuális állapotukat.
  - *managed – alap állapot*
  - *monitor – monitor mód*

- *sudo airmon-ng check kill*: ellenőrzi is leállítja azokat a folyamatokat, amik beavatkozhatnak a működésbe
- *sudo airmon-ng start wlan1: monitor módba állítás*
    - a parancs kiadása után létrejön egy wlan1mon interfész, ami már monitor módban fut

Jelszólista alapú törés

A monitor módba kapcsolás után végeztem egy nyers 802.11 keret, csomaggyűjtést:

- *sudo airodump-ng wlan1mon*



9.10. ábra: Felderített hálózatok

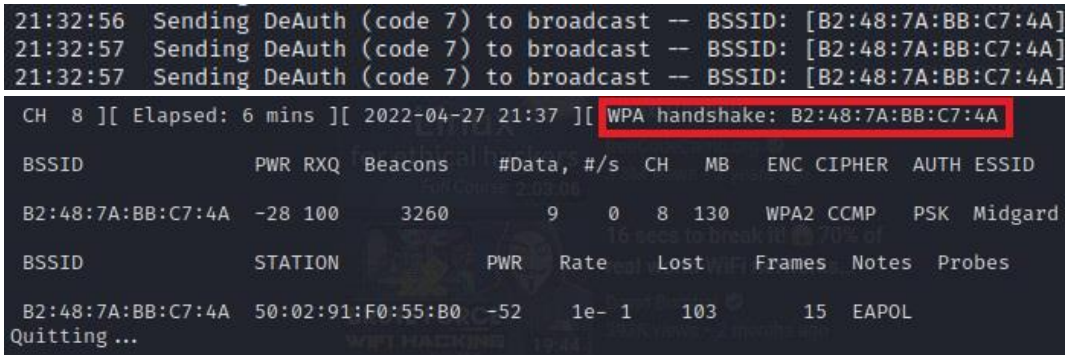
A Midgard a saját hálózatom, azért van kettő, mert két router is ugyanazt szórja. Látszik, hogy a 8-as vezérlőcsatornán van. A koncepció az, hogy az AP-n található klienseket leválasztjuk a hálózatról, majd az újracsatlakozásnál elfogjuk a 4 lépcsős kézfogás csomagjait, melyben elküldi a kliens a WPA2 jelszót titkosítva. Erre a titkosított adatcsomagra már rá tudunk nézni wireshark-al, de ami még fontosabb, hogy egy szótárlistával offline tudunk rá támadást indítani és ha a szótárban benne van a szó, akkor megvan a jelszó.

Tehát megvan a kinézett hálózathoz a MAC cím, rá lehet erre külön engedni az *airodump-ng*-t, úgy paraméterezve, hogy az egész figyelést írja ki egy .cap fájlba.

- *sudo airodump-ng –w hack –c 8 –bssid B2:48:7A:BB:C7:4A wlan1mon*

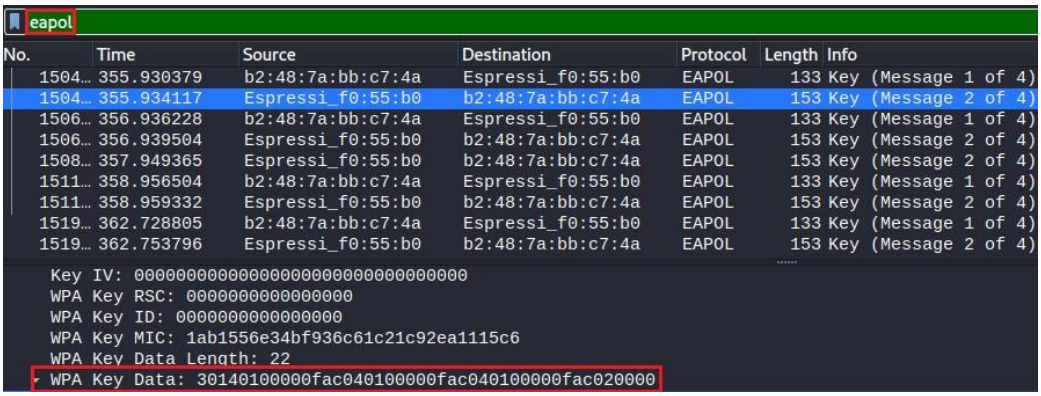
Miközben ez fut, indíthatjuk egy másik konzolon a leíratkozási, deautentikációs broadcast utasítást.

- *sudo aireplay-ng –deauth 0 –a B2:48:7A:BB:C7:4A wlan1mon*



9.11. ábra: AP kliensek leválasztása és a kézfogás elkapása

Ha megvan a pirossal jelölt, le lehet állítani CTRL+C-vel mindkét parancsot. Ekkor létrejön egy hack-01.cap állomány, amit be tudunk importálni wireshark-ba és rá tudjuk engedni a törést is. Az 'eapol'-ra szűrve már csak az autentikációs csomagokat fogjuk látni. Ez a protokol felel a hitelesítésért (EAPoL, Extensible Authentication Protocol over Lan)



9.12. ábra: wireshark WPA2 kulcs

A bekeretezett adatot akarjuk megfejteni. A kali-ban alapból megtalálhatóak jelszó szólisták, szótárak. Az egyik népszerű lista a 'rockyou'. Ez egy szivárgásból keletkezett lista. Érdekességképp megjegyzendő, hogy a legnépszerűbb jelszó manapság az *123456.*

- *aircrack-ng hack-01.cap –w rockyou.txt*

Természetesen nem sikerült megfejtenie.

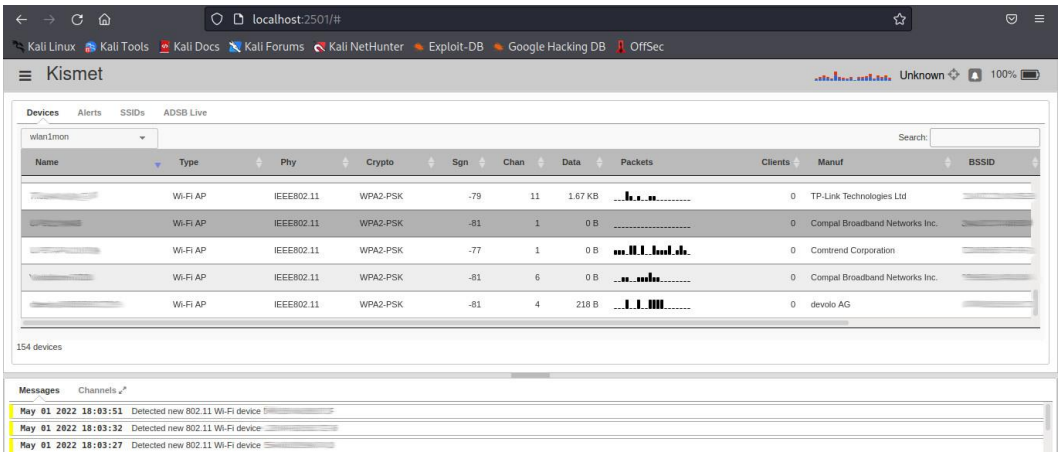
Vannak előre megírt, úgynevezett auditáló eszközök is a kaliban, amik egy parancs elindításával fellistázzák az elérhető hálózatokat és azok tulajdonságait és nekünk gyakorlatilag csak ki kell választani a célpontot és a szoftver végigpróbálja az ismert sérülékenységekkel. Egy ilyen népszerű program a Wifite. Automatikusan monitor módba teszi az interfészt, elkapja a kézfogás csomagokat, megpróbálja saját jelszó listájából felismerni a jelszót. Ha be van kapcsolva a WPS mód, akkor ellene indít brute-force támadást. Ebben az esetben nyers erőből, próbálgatásból kísérli meg kitalálni a kódot.

Kismet

A Kali-nak ez is része, nem kell külön telepíteni. A

- *kismet –c [interfész]*

parancssal kell elindítani. Az interfészt a fejezet elején leírt módon tegyük előtte monitor módba. Majd a legfelső sorban ki is írja, hogy a http://localhost:2501-et, böngészőbe beírva elérhetjük a grafikus felületét.



9.13. ábra: kismet

Látható, hogy 154db eszközt felismert, kb. 5perc futtatás után és ez a szám 200 fölé növekedett. Több, mint 30 AP-t listázott fel. A vezeték nélküli technológiából adódóan sajnos ez ilyen könnyen kivitelezhető. Láthatóak egyéb fontos információk is, titkosítás, vezérlőcsatornák, jelerősségek és gyártók is.

A MAC címek első 3 tagja azonosítja a gyártót, így a neten fellelhető online MAC adatbázisok alapján könnyen beazonosítható egy vezeték nélküli kamera és az előzőekben ismertetett paranccsal,

- *aireplay-ng -0 0 -a [AP MAC] -c [kliens MAC] [interfész]*

a kinézett eszköz igen könnyen leválasztható a hálózatról.

10. További teendők, fejlesztési lehetőségek

A mentések 3-2-1-es szabályának utolsó szereplője még nincs megvalósítva, mert jelenleg nincs rá lehetőségem, hogy egy harmadik helyen is tároljak automatán mentést. Ezt nem szeretném kitenni az egyik népszerű felhő megoldásba (Drive, Dropbox). De itt is vannak már kifejezetten biztonságra hangsúlyt fektető megoldások, pl. a magyar fejlesztésű Tresorit. Ha nem ezt, akkor valószínűleg egy bérelt Linux-os virtuális gépet fogok valamelyik hoszting szolgáltatónál vásárolni és ezt a gépet fogom erre a célra használni.

Hálózati szinten egy újabb védelmi megoldás lehet, ha a különböző eszközöket hálózatilag szeparálom, külön IP címtartományok, VLAN-okkal. Egy ésszerű megközelítésnek gondolom, hogy a publikus, használati eszközök (telefonok, laptopok, számítógépek) egy csoport, a megbízható IoT eszközök (neves gyátók, Sonoff, Shelly) egy másik és az olcsó kínai IoT eszközök a harmadik (Blitzwolf, Xiaomi).

Az adatbázis terén az InfluxDB jelenleg a 2.2-es verziónál tart, én az 1.8-as verziót használom. Ez már egy új lekérdező nyelvet tartalmaz és más az adattárolási modellje is, idővel érdemes lehet áttérni erre.

A Home Assistant-ot folyamatosan fejlesztik, egyre több gyári integrációval, bővítménnyel és funkciókkal látják el. Ezeket havi és hóközi frissítésekben publikálják. Egy ház körül mindig akad tennivaló, felújításra váró terület, így egyre több és több eszköz kerül majd implementálásra, melyekkel újabb és újabb lehetőségek nyílnak meg az okosítás és az automatizálás terén. Ha minden jól megy, akkor idén végre sikerül a kertet füvesíteni, melyhez kiválóan illeszthető egy öntözőrendszer, csapadékméréssel, esetleg talajnedvesség méréssel.

Az okos otthonok világában a fejlesztések sosem érhetnek véget.

11. A szakdolgozat tartalmi összefoglalója magyarul és angol nyelven

Mikor 2020-ban kitaláltam a szakdolgozatom témáját, még nem tudtam mekkora fába vágom a fejszémet. Azt gondoltam, hogy az eszközök telepítése, konfigurálása és legfőképpen arról információt szerezni, hogy mi az állapota az eszköznek, mik a paraméterei sokkal egyszerűbbek. Nagyon sok időt töltöttem kutatással, mire megtaláltam az ntopng-t, melynek van közösségi, ingyenes verziója.

In 2020 my thesis topic was finalized, I have had no idea how long journey did I started. My expectation and experience with the device setup, configuraion and retrieve information were totaly different . The result of multiple hours of investigation was to get familiar with ntopng and its usage (community edition).

A következő meglepetés akkor ért, mikor már feltelepítésre és beállításra került az okosotthon, de nem fértem hozzá az eszközök logjaihoz, vagy az eszköz nem volt képes magáról logot küldeni. Sosem gondoltam volna, hogy egy prémium kategóriájú router nem képes magáról snmp protokollon információt adni. Ehhez közösségi fejlesztésű firmware-re van szükség, vagy speciális eszközre, új beruházásra.

Next step was to retrieve logs and information from the smart home device after proper setup and configuration. Expectiations about the feature set of a premium device did not meet with my expectations (snmp informations). In order to extend device capabilities a firmware (community supported) upgrade is necessary or invest into a new device

A loggyűjtésben, eszköz információ szerzésben és ezek tárolásában is szükség volt egy újrakezdésre, mert az első eszközt, amit az irodalomkutatás alapján megfelelőnek tartottam, le kellett cserélnem. A Prometheus nevű időbélyegzett adatbázisról úgy gondoltam, hogy tökéletes lesz, de több okból is el kellett vetnem. Mivel natívan támogatja a Qnap NAS, ezért is választottam, de a kollektorait külön-külön kellett telepíteni és ezeket konténerben hoztam létre, ahol nehezen voltak menedzselhetőek. Ezért váltottam a TICK stack-re, az InfluxDB-re. Ahol egy kollektorban van minden definiálva és váltottam egy Linux alapú miniPC-re a konténerek helyett, ami igencsak megkönnyítette a konfigurációt.

There was a need to restart because the log and device information collection and storage capabilities of the first device, I found during my research was not enough. I also needed to alter my choice of timestamped database, Prometheus since it had multiple issues. Prometheus supports native Qnap NAS (main reason to go with this database), although its collectors needed to be setup one-by-one, a container setup was also needed and it increased huge management overhead. I had to change to TICK stack and InfluxDB. In this case all definitions are inside the containers. The change of the containers to a Linux based miniPC containers eased the configuration.

A Grafana vizualizáció és integrációi nagyon könnyen beállíthatók és kezelhetők a webes felületen keresztül. Rengeteg előre megírt dashboard-ja van, de kézzel is nagyon könnyen tudunk létrehozni paneleket és komplett dashboardokat.

Grafana has an advanced visualization and integration. Grafana web interface gives simple management and configuration. It provides multiple dashboard presets but manual dashboard configuration is also possible.

Az okosotthon központ, mint entitás egyszerűen telepíthető, de érdemes előre jól átgondolni, hogy melyik verziót válasszuk. Egyszerű normál használatra bőven elegendő a HA konténer. Működnek az integrációk nagyon nagy része, képes vezérelni az entitásokat, automatizmusokat hozhatunk létre. Viszont, ha jobban bele akarunk nézni a virtuális doboz alá, akkor szükségünk van a Supervisor módra és ebben az esetben el kell hagyni a konténert és egy sokkal erőforrásigényesebb megoldás kell. Virtualizációra, vagy egy dedikált számítógépre. Van erre kész megoldása is a HA-nak (Home Assitant Blue néven, előre telepített HA OS-el), ha valaki 0-ról építkezik, érdemes lehet ebbe beruházni.

The smarthome center, as entity, is easy to setup, but a proper planning is needed, which version to go with. HA container is sufficient for a normal use-case. Integration, control and automation capabilities are available but not complete. In order to look under the hood of the virtual box, we need to have a Supervisor mode, it leaves the container and this solution requires more resources (virtualization or a dedicated compute node). There is a solution called HA (Home Assistant Blue). If we are building from zero it would make sense to invest into it.

A biztonságra már a tervezés alatt is kiemelt figyelmet fordítottam. A kilépési pontnál próbáltam minden létező védelmi megoldást érvényesíteni. Talán a legfontosabb védelmi megoldás az lett, hogy az otthoni hálózat egyetlen eszköze sincs kiengedve az internetre, az otthoni hálózat csak SSL VPN-el érhető el. További végpont védelmi megoldásokat ad a router AiProtection funkciója. Ha hasonló megoldásokat tud biztosítani egy belépési pont, ezeket érdemes bekapcsolni. Semmi nem nyújthat 100%-os védelmet, folyamatosan derülnek ki sérülékenységek, de adataink védelme érdekében érdemes mindent megtenni és rászánni egy kis időt megnézni, hogy miket tudunk egyszerűen aktiválni.

Security was in focus during the design phase. I tried to apply every possible mitigation to its endpoint security. The major mitigation was to limit home network devcices access from internet side, only SSL VPN access is allowed. Further endpoint security is provided by AiProtection feature. If similar solutions can be provided to the endpoint, it is recommended to turn on. There is no 100% security since vulnerabilites are discovered in daliy basis. Although data protection is important and spending some time to check out security capabilities is a must.

A védelmet folyamatosan fenn kell tartani, a legalapvetőbb tevékenység ehhez, hogy a telepített eszközök firmware-ének frissen tartása. Ez a webes felületükön könnyen kivitelezhető és a HA dashboard-ra is ki lehet tenni az entitáshoz, hogy van-e új. Plusz a

telepített rendszerek is képesek értesítést küldeni a saját és a többi eszköz frissíthetőségéről, amit érdemes minél előbb megtenni. Persze előtte érdemes megnézni a kiadási jegyzeteket, mert lehetnek bennük olyan változások, amik valamit működésképtelenné tesznek. Ezért nagyobb frissítés előtt érdemes betartani a sorrendet, hogy először készítünk mentést (snapshot, backup), újraindítás, még egy mentés, majd mehet a frissítés. Mindezek után én szintén szoktam csinálni még egy mentést az szoftverrel és lehet tesztelni.

Protection need to be maintained by keeping firmwares up-to-date. Web user interface provides a simple way to do so on HA dashboard with updates of entities. Additionally the setup system can send notifications about its and neighbouring device updates. Before applying a patch it is recommended to read release notes, since changes may turn existing devices unaccessible. Backup, snapshot first and after apply patching is a must before any major upgrades. A final save and testing is recommended as well.

A port tükrözött forgalom figyeléséhez be kellett szerezni egy megfelelő switch-et és be kellett konfigurálni, majd a megfelelő helyre implementálni. Erre behallgatva láthatóvá válik minden ki- és bejövő csomag. Itt van egy-két titkosítatlan csomag, de semmi jelentős.

In order to mirror ports an additional switch install and configuraiton was necessary. After sniffing incoming and outgoing traffic, it was visible that some packages were encrypted but nothing major.

A vezeték nélküli hálózat vizsgálatához a Kali linux disztribúciót vettem segítségül. Ebben megtalálható minden szoftver amire szükség lehet. Megvizsgáltam, hogy egy képzeletbeli támadó mit lát a *kismet*, a *wireshark* és az *aircrack-ng* programcsalád segítségével. Nagyon könnyen beazonosíthatóak és leválaszthatóak a vezeték nélküli kommunikációt folytató eszközök. Végeztem szimulált támadást a saját hálózatom ellen, melynek során elkaptam a 4 lépéses kézfogás során küldött titkosított kulcsot, melyet offline megpróbáltam megfejteni, sikertelenül. Majd a Wifite prpgrammal teszteltem a hálózatom, amely szintén nem járt sikerrel.

Kali linux was used to analyse wireless network. This distribution includes all necessary software packages. I have verified what could a potential attacker might see with the help of with kismet, wireshark and aircrack-ng tooling families. Devices on the wireless network have been identified and segregated easily. During a simulated attack against my own network I managed to reveal secret key during a 4-way-handshake that I could not decrpyt later offline. Simulated attack with Wifite toolset was unsuccessful as well.

12. Irodalomjegyzék

Könyvek:

[1] Chris, Sanders.: *Practical Packet Analysis, 3rd edition. No Starch Press*, 2017

Kaung Nyunt, San.: Implementation of IoT-based home control and monitoring system using raspberry pi and Node MCU. *University of Computer Studies, Yangon*, 2019

He-Jun, Lu., Yang, Yu.: Research on WiFi Penetration Testing with Kali Linux. *Hindawi Complexity Volume 2021*, 2021

Michael, Asante., Kwabena, Akomea-Agyin.: Analysis of Security Vulnerabilities in Wifi-Protected Access Pre-Shared Key. *International Research Journal of Engineering and Technology (IRJET)*, 2019

Arif, Sarl., Mehmet, Karay.: Comparative Analysis of Wireless Security Protocols: WEP vs WPA. International Journal of Communications, *Network and System Sciences*, 2015

CISA.: Home Network Security. *Cybersecurity & Infrastructure Security Agency*, 2020

Folyóiratok:

Internetes források:

(https://www2.deloitte.com/us/en/pages/about-deloitte/articles/press-releases/deloitte-launches-connectivity-mobile-trends-survey.html#), utoljára megtekintve: 2020.12.02.

(https://iot-analytics.com/wp/wp-content/uploads/2020/11/IoT-connections-total-number-of-device-connections-min-1536x806.png), utoljára megtekintve: 2020.12.08.

(https://www.wi-fi.org/download.php?file=/sites/default/files/private/Wi-Fi_CERTIFIED_6_Highlights_201910_0.pdf), utoljára megtekintve: 2021.05.21.

(https://zigbeealliance.org/), utoljára megtekintve: 2020.12.15.

(https://www.rfwireless-world.com/images/network-topology-types.jpg), utoljára megtekintve: 2020.12.15.

(https://z-wavealliance.org/), utoljára megtekintve: 2020.12.15.

(https://www.ikea.com/hu/hu/customer-service/product-support/app-gateway/), utoljára megtekintve: 2021.05.21.

(https://www.openhab.org/), utoljára megtekintve: 2021.05.03.

(https://developers.home-assistant.io/), utoljára megtekintve: 2021.05.05.

(https://getvera.com/pages/veraplus), utoljára megtekintve: 2020.12.15.

(https://grafana.com/docs/grafana/latest/), utoljára megtekintve: 2022.04.30.

(https://docs.influxdata.com/influxdb/v1.8/), utoljára megtekintve: 2022.04.30.

(https://www.ntop.org/guides/ntopng/), utoljára megtekintve: 2022.04.30.

(https://www.kali.org/docs/), utoljára megtekintve: 2022.05.05.

(https://shelly-api-docs.shelly.cloud/gen1/), utoljára megtekintve: 2022.05.05.