ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS
INFORMATIKAI KAR

5. sz. melléklet

SZAKDOLGOZAT

OE-NIK	Hallgató neve:	**Bardi Dániel Márk**
2021	Hallgató törzskönyvi száma:	**T/006124/FI12904/N**

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

SZAKDOLGOZAT FELADATLAP

Hallgató neve: **Bardi Dániel Márk**
Törzskönyvi száma: T/006124/FI12904/N
Neptun kódja: [redacted]

A dolgozat címe:

IPv6-os hálózatok védelmének és sérülékenységeinek elemzése
Analyzing the security and the vulnerabilities of IPv6 networks

Intézményi konzulens: Vörösné Dr. Bánáti-Baumann Anna
Külső konzulens:

Beadási határidő: 2021. december 15.

A záróvizsga tárgyai: Számítógép architektúrák
Felhőszolgáltatási technológiák és IT biztonság specializáció

A feladat

Napjainkban egyre több és több, az internetre csatlakoztatott eszközt használnak az emberek világszerte. Emiatt az IPv4-es címtérből kioszthatóIP címek lassan egy évtizede már elfogytak, annak ellenére, hogy a folyamatot igyekeztek lelassítani különféle technológiák és módszerek bevezetésével, mint például a NAT-tal és privát hálózati címek használatával. Ennek következtében szükség volt egy új címzési módszer bevezetésére és ezt a szerepet jelenleg is az IPv6 protokoll tölti be. Jelenleg az IP protokoll 6-os verziója még nem terjedt el olyan széleskörben, mint az elődje, de emellett vannak használatban olyan technológiák, amelyek lehetővé teszik a két protokoll egymás melletti működését, amíg nem történik meg a teljes átállás. Bizonyos komponensek működése megegyezik az IP mindkét verziójában, viszont emellett bevezetésre kerültek olyan mechanizmusok, amik az IPv4-ben nem voltak jelen és új támadási felületet biztosíthatnak a támadók számára.
A szakadolgozat célja az IPv6 specifikus sérülékenységek feltérképezése és bemutatása, valamint ezen sérülékenységeket kihasználó támadások detektálásának és védelmi megoldásainak elemzése egy SOC (Security Operation Center, Biztonsági Műveleti Központ) laborkörnyezetben.
A hallgató feladata IPv6-os támadások feltérképezése és szimulálása egy erre a célra kialakított tesztkörnyezetben, továbbá a támadások detektálásának vizsgálata és védelmi megoldások kialakítása SOC környezetben.

A dolgozatnak tartalmaznia kell:

- az IPv6 protokoll működésének bemutatását összehasonlítva az IPv4 protokollal,
- az IPv6 sérülékenységeit, illetve az IPv6 specifikus támadások bemutatását,
- a szimulációkhoz használt tesztkörnyezet tervét és leírását,
- adott támadás szimulációját és elemzését,
- a detektáló és megelőző megoldások lehetőségeit SOC környezetben,
- az eredmények értékelését és továbblépési lehetőségeket.

Ph.

ÓBUDAI EGYETEM * Neumann János Informatikai Kar * 1. Budapest

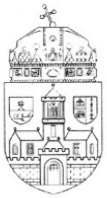
.....
Dr. Eigner György
intézetigazgató

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

A dolgozatot beadásra alkalmasnak tartom:

.....
külső konzulens

.....
intézményi konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
7. sz. melléklet

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, 2022.05.12.

Bagdi Dániel
hallgató aláírása

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:

BARDI DÁNIEL MÁRK [redacted] ..... NAPPALI.....

Telefon: Levelezési cím (pl: lakcím):

[redacted]

Szakdolgozat / Diplomamunka[1] címe magyarul:

IPV6-OS HÁLÓZATOK VÉDELMÉNEK ÉS SÉRÜLÉKENYSÉGEINEK ELEMZÉSE

Szakdolgozat / Diplomamunka[2] címe angolul:

ANALYZING THE SECURITY AND THE VULNERABILITIES OF IPV6 NETWORKS

Intézményi konzulens: Külső konzulens:

VÖRÖSNÉ DR. BÁNÁTI-BAUMANN ANNA

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.02.22	TÉMA MEGHATÁROZÁS, MEGVALÓSÍTANDÓ RÉSZFELADATOK KITŰZÉSE	Bánáti Anna
2.	2021.03.26	FELADATKIÍRÁS PONTOSÍTÁSA, FORRÁSOK ÁTTEKINTÉSE	Bánáti Anna
3.	2021.04.22	MEGVALÓSÍTÁS ÁTTEKINTÉSE	Bánáti Anna
4.	2021.05.11	PREZENTÁCIÓ MEGBESZÉLÉSE	Bánáti Anna

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

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplo-mamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2021.05.14.....

Bánáti Anna
.....
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Neptun kód: Tagozat:

BARDI DÁNIEL MÁRK [redacted] NAPPALI

Telefon: Levelezési cím (pl: lakcím):

[redacted]

Szakdolgozat / Diplomamunka[1] címe magyarul:

IPV6-OS HÁLÓZATOK VÉDELMÉNEK ÉS SÉRÜLÉKENYSÉGEINEK ELEMZÉSE

Szakdolgozat / Diplomamunka[2] címe angolul:

ANALYZING THE SECURITY AND THE VULNERABILITIES OF IPV6 NETWORKS

Intézményi konzulens: Külső konzulens:

VÖRÖSNÉ DR. BÁNÁTI-BAUMANN ANNA

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2022.03.11	IMPLEMENTÁLANDÓ TÁMADÁSOK ÉS VÉDEKEZÉSI FORMÁK ÁTTEKINTÉSE	Bánáti Anna
2.	2022.03.30	BESZÁMOLÓ KÖVETELMÉNYEINEK MEGBESZÉLÉSE	Bánáti Anna
3.	2022.04.21	MEGVALÓSÍTÁS ÁTTEKINTÉSE, TOVÁBBI FELADATOK KIJELÖLÉSE	Bánáti Anna
4.	2022.05.10	PREZENTÁCIÓ TARTALMÁNAK ÉRTÉKELÉSE	Bánáti Anna

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

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplo-mamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2022. május 13.

Bánáti Anna
.....
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

Tartalom

Tartalmi Összefoglaló 10
1. Bevezetés 11
2. Internet Protocol Version 4 (IPv4) 13
2.1. IPv4 címek felépítése 13
2.2. Az IPv4 protokoll működése és felépítése 14
2.2.1. Az IPv4 protokoll karakterisztikái 15
2.2.2. IPv4 csomag fejléce 16
2.3. IPv4 címek kimerülése 18
3. Internet Protocol Version 6 (IPv6) 20
3.1. IPv6 címek felépítése 20
3.1.1. Global Unicast IPv6 címek 21
3.1.2. Link-local IPv6 címek 22
3.2. Az IPv6 fejléc felépítése 22
3.2.1. Kiegészítő fejlécek 24
3.3. Neighbor Discovery Protocol 25
3.3.1. Stateless Address Autoconfiguration (SLAAC) 26
3.3.2. Duplicate Address Detection (DAD) 27
4. IPv6 biztonság 29
4.1. Implementációs problémák 29
4.2. IPv6 sérülékenységek 30
4.2.1. Több kioszthatő IP cím sérülékenység 30
4.2.2. EUI-64 sérülékenység 30
4.2.3. Multicast sérülékenység 31
4.2.4. Kiegészítő fejlécek sérülékenysége 31
4.2.5. Fragmentálási sérülékenység 31
4.2.6. Neighbor Discovery sérülékenység 31
4.2.7. IPv6 bevezetését támogató technikák sérülékenységei 32
4.3. Védelmi megoldások 32
4.3.1. IPv6 tűzfal szabályok 32
4.3.2. Secure Neighbor Discovery 33

4.3.3. RA Guard 34
4.3.4. IPsec 34
5. IPv6 támadási formák 36
5.1. Neighbor Discovery Protokollt alkalmazó támadások 36
5.1.1. Neighbor Solicitation/Advertisement Spoofing 36
5.1.2. Duplicate Address Detection támadás 36
5.2. Kiegészítő fejlécekkel kapcsolatos támadások 37
5.2.1. Routing Header Type 0 támadás 37
5.2.2. Fragment headert alkalmazó támadások 38
6. Rendszerterv 39
6.1. Virtuális hálózat 39
6.1.1. Hálózati topológia elemei 39
6.2. Megvalósított támadások 40
6.2.1. Denial Of Service támadás Duplicate Address Detection üzenetekkel 40
6.2.2. Denial Of Service támadás Router megszemélyesítésével 41
6.3. Támadás detektálása a Security Onion eszközeivel 41
6.3.1. Behatolás érzékelő rendszerek szerepe 41
7. Megvalósítás 43
7.1. Virtualizált környezet kezdeti konfigurációja 43
7.1.1. Cisco routerek kezdeti konfigurációja 44
7.1.2. Használt hálózati címzés 45
7.1.3. Security Onion kezdeti konfigurációja 45
7.2. Alapvető működés megerősítése 46
7.2.1. Hálózat alapvető funkcionalitásának megerősítése 47
7.2.2. Security Onion alapvető funkcionalitásának megerősítése 47
7.3. Megvalósított támadások 50
7.3.1. Denial Of Service támadás Duplicate Address Detection üzenetekkel 50
7.3.2. Denial Of Service támadás Router megszemélyesítésével 53
7.4. Támadások detektálása 56
7.4.1. Duplicate Address Detection üzeneteken alapuló támadás detektálása 56
7.4.2. Router Advertisement csomagokon alapuló támadás detektálása 63

7.5. Egyéb védelmi megoldások 65
7.5.1. RA Guard konfigurálása 66
8. Eredmények értékelése 67
8.1. Végrehajtott támadások hatása és effektivitása 67
8.2. Alkalmazott detektálási módszerek hatékonysága 67
8.2.1. Szignatúra alapú detektálási módszer hatékonysága 68
8.2.2. Anomália alapú detektálási módszer hatékonysága 69
9. Továbbfejlesztési lehetőségek 71
10. Összefoglalás 72
11. Summary 73
12. Irodalomjegyzék 74
13. Ábrajegyzék 76
14. Mellékletek 78

TARTALMI ÖSSZEFOGLALÓ

A Szakdolgozatomban bemutatom az Internet Protocol Version 4 (IPv4) és az Internet Protocol Version 6 (IPv6) protokollok működésében jelentkező alapvető különbségeket és hasonlóságokat. Ezen belül kitérek a protokollok fontos karakterisztikáira, az általuk használt címek sajátosságaira és a működésük szempontjából fontosabb tényezőkre. Továbbá ismertetem az IPv6 hálózat biztonsági szempontból fontos hiányosságait, valamint bemutatom a protokoll új mechanizmusait és az ezeket kiaknázó támadási módszereket. A támadások folyamatát egy GNS3 által szimulált hálózatban mutatom be részleteiben. A támadásokat a THC IPv6 eszközkészlet segítségével valósítom meg. A virtuális hálózatnak szintén része egy Security Onion nevű SOC (Security Operation Center). A Security Onion részét képzi több hoszt és hálózatalapú behatolás érzékelő rendszer, valamint további elemzésre és adatok vizualizálására képes alkalmazások. A támadások során generált adatforgalom ezen a hálózatbiztonsági rendszeren halad keresztül és az elkapott adatcsomagokon keresztül bemutatom a támadások részleteit és folyamatát. Gyakori probléma, hogy a hálózati behatolás érzékelő rendszerek közel sem 100%-os hatékonysággal szűrik ki az IPv6 specifikus támadásokat, így sokszor szükséges a szűrés pontosítása, valamint a megfelelő beállítások alkalmazása. A folyamat során különféle szabályok beállításával próbálom megismertetni a rendszerrel az egyes IPv6-os sérülékenységeket kiaknázó támadások ismérveit, annak érdekében, hogy a jövőben automatikusan képes legyen észlelni azokat. Ezek után megvizsgálom, hogy az említett támadások, milyen más módszerekkel előzhetőek meg.

1. BEVEZETÉS

Számítógépes hálózatról akkor beszélhetünk, ha egy vagy több számítógépet összecsatlakoztatunk, oly módon, hogy azok képesek egymással kommunikálni.

A számítógépes hálózatok használata iránti igény, akkor jelentkezett először, amikor számítógépek csoportjáról szerettek volna elérni egy közös erőforrást. Ilyen erőforrások lehetnek például háttértárolók, nyomtatók vagy egyéb programok és adatbázisok. A későbbiekben világossá vált, hogy nem elegendő a helyi hálózatok használata, mivel a területi kiterjedésük nem biztosít lehetőséget távoli állomások összekapcsolására. [1]

Az Internet elődjeként az Amerikai Védelmi Minisztérium kutatóintézete által 1969-ben létrehozott ARPANET-et (Advanced Research Projects Agency) tartjuk számon. Eleinte ezt a nagy kiterjedésű hálózatot kutatási és katonai célokra használták fel, de későbbiekben elérhetővé vált szélesebb körben is. A kutatás eredményeképpen olyan technológiáknak tették le az alapjait, mint az elosztott csomagkapcsolás és forgalomirányítás. [2]

Egy hálózatban minden résztvevő eszközt azonosítani kell valamilyen egyedi azonosítóval. Azonosítóként szolgál az eszköz hosztneve, logikai és fizikai címe. Fizikai címként gyakran MAC (Media Access Control) címeket használunk. Ez a 48 bites fizikai cím egyértelműen azonosítja egy eszköz hálózati kártyáját és ezáltal magát az eszközt is. A hálózati kártyákhoz a MAC címeket a gyártás során rendelik hozzá. Logikai címzésként gyakran IP címeket használunk. Jelenleg két elérhető IP protokoll létezik, IPv4 és IPv6. Az IP protokoll négyes verzióját először az ARPANET-ben használták 1983-ban. A publikus IP címek kezeléséért és kiosztásáért az IANA (Internet Assigned Numbers Authority) és további öt regionális szervezet felel. Az IPv4-es címek 32 bites logikai címek, amelyek oktettenként (8 bit) ponttal vannak tagolva és az emberek számára könnyen olvasható, tízes számrendszerben vannak ábrázolva. Ez a 32 bites címtér összesen 4,294,967,296 egyedi cím kiosztására ad lehetőséget, leszámítva azokat a speciális címeket, amelyek nem kioszthatóak. [3][4]

Már az 1980-as években egyértelművé vált, hogy az IPv4-es címtérből kioszthatő IP címek egyszer majd elfogynak. A folyamatot felgyorsította az, hogy egyre több és több felhasználó használta az internetet, valamint rendelkezett olyan mobileszközzel, amely képes az internethez való csatlakozásra. Annak érdekében, hogy lelassítsák ezt a folyamatot különféle technológia megoldásokkal álltak elő: CIDR (Classless Inter-Domain Routing), NAT (Network Address Translation). Ezek segítségével sikerült mérsékelni valamilyen szinten az IP címek fogyását, de nem jelentett végleges megoldást. Az IANA és a további öt IP címekért felelős szervezet címteréből 2019. novemberéig elfogytak a kioszthatő IPv4-es címek. Erre a problémára az IPv6-os protokoll kifejlesztése jelentette a végleges gyógyírt, bőven elegendő IP címet biztosítva a távoli jövőben is. [4]

Az IPv6-os protokoll sok szempontból jobb, mint az elődje, például sokkal hatékonyabb adatátvitelt biztosít és biztonságosabb is. A két protokoll működésében felfedezhetünk hasonlóságokat és olyan új mechanizmusokat, amely a régebbi verzióban még nem szerepelt.[5]

Az IPv6 ezek mellett nagyobb címteret is biztosít, az egyes címek 128 bitesek, így lényegesen több kioszthatő IP cím áll rendelkezésre, mint IPv4 esetén. A IPv4 utódja nagyjából 1998 óta áll rendelkezésre, de a kezdeti adoptálása viszonylag lassú folyamatnak bizonyult – 2014 áprilisában a világon generált internetes forgalomnak mindössze 3,5 százalékát tette ki IPv6-os forgalom. [6] A Google statisztikái szerint ez a szám manapság 30% körül alakul. [7]

A protokollok közötti váltás megkönnyítése érdekében létrehoztak számos mechanizmust, amely áthidalja a két protokoll különbözőségeit (6in4, NAT64, 6RD stb.), amíg nem zajlik le ez a folyamat. Tekintve, hogy előbb utóbb elengedhetetlen lesz az IPv6 szélesebb körben való használata, így fontos feladat lett a protokoll új sérülékenységeinek feltérképezése és támogatása különféle biztonsági megoldásokban.

2. INTERNET PROTOCOL VERSION 4 (IPV4)

Az internet működése érdekében elengedhetetlen az IP címek használata. Az egyéni IP címek segítségével – amelyeket a használatba vett eszközök hálózati kártyáihoz rendelünk – egyértelműen azonosíthatóak a kommunikációban résztvevő végpontok és segítségükkel a forgalomirányítók meghatározzák a forrástól a célig vezető útvonalat.

Az Internet Protocol Version 4-et először 1978-ban adták ki és 1981-ban lett meghatározva a hozzátartozó szabvány (RFC 791), majd 1983-ban került bevezetésre az ARPANET hálózatában.[8] Az IPv4 az Internet Protocol negyedik verziója, de az első olyan, ami széleskörben elterjedtté vált. A szabvány szerint az IPv4-es címek 32 bit hosszúak, ez elméletben 4,3 milliárd egyéni IP címet jelent, de ezeknek egy része speciális célokra van fenntartva, így ténylegesen 3,7 milliárd logikai cím osztható ki. Mivel idővel nyilvánvalóvá vált, hogy ez a címtér nem fogja tudni kielégíteni az igényeket, így megkezdődött egy új protokoll kifejlesztése, amely nagyobb címtérrel rendelkezik és átveheti az IPv4 helyét a jövőben. Ezt a szerepet az IPv6 töltötte be, amelyet 1995-ben szabványosítottak (RFC 1883). [9][10]

2.1. IPv4 címek felépítése

Az IPv4-es címek hierarchikus 32 bites logikai címek. Pontozott decimális formában írják őket, hogy könnyebben olvasható legyen az emberek számára. Tehát ez azt jelenti, hogy nyolc bitenként ponttal választják el őket és így négy oktettre bomlik a teljes cím. Az egyes oktettek tízes számrendszerben 0-255-ig vehetnek fel értékeket (pl. 192.168.0.1). Az IP címek két részre bomlanak, hálózat- és gépazonosító bitekre. Azt, hogy ez a határ hol húzódik azt az alhálózati maszkok segítségével határozhatjuk meg. Ugyanazon alhálózatban az összes hoszt IP címének a hálózatazonosító része meg kell egyezzen, viszont minden hoszt esetében a gépazonosító biteknek el kell térniük.

Eleinte az egyes IP címek az egyes osztályokból voltak kioszthatóak, ahogyan az RFC 790 szabványban meghatározták azokat. Összesen öt darab osztályra bontották a teljes címteret, ezek sorban: A, B, C, D és E osztály. Az A, B és C osztályok voltak elérhetőek mindenki számára, a D osztályú IP címeket multicast címzéshez tartották fent, az E osztályt pedig kísérleti célokkal hozták létre. [11]

Osztály	Tartomány	Alhálózati maszk
A	1.0.0.0 – 126.255.255.255	255.0.0.0
B	128.0.0.0 – 192.255.255.255	255.255.0.0
C	192.0.0.0 – 223.255.255.255	255.255.255.0
D	224.0.0.0 – 239.255.255.255	-
E	240.0.0.0 – 254.255.255.255	-

2.1. táblázat: Az IPv4-es címosztályokhoz tartozó IP címtartományok és a hozzájuk tartozó alhálózati maszkok

Az IP címosztályokhoz tartozó alhálózati maszkok határozzák meg a cím hálózatazonosító és gépazonosító részét. Az A osztálynál az első oktett felel meg az IP cím hálózatazonosító részeként, a B osztálynál az első két oktett és C osztálynál pedig az első három. Az A osztályú IP címeket olyan esetben érdemes használni, ha kevés alhálózatra van szükségünk, de azokon belül sok hosztot szeretnénk megcímezni (~16 millió kioszthatő cím). B osztályú IP címek választásával közepes méretű alhálózatokat valósíthatunk meg, nagyjából 65000 cím osztható ki alhálózatonként. A C osztályú IP címek segítségével kisebb hálózatokat tudunk kialakítani, alhálózatonként 254 hoszttal.

Ezek mellett beszélhetünk privát és publikus IPv4 címekről. A publikus IPv4 címek részt vesznek az internetes forgalomirányításban, míg a privát címek nem. A privát címeket belső hosztoknak osztják ki egy szervezeten belül a helyi hálózatokon. A privát IP címek használata az 1990-es évektől kezdődően terjedt el, ez a módszer egyike az IPv4-es címek elfogyását lelassító módszereknek. A privát IP címtartományok (az RFC 1918 szerint) az A, B és C osztályokban a következőek: [11][12]

- A osztály: 10.0.0.0 – 10.255.255.255
- B osztály: 172.16.0.0 – 172.31.255.255
- C osztály: 192.168.0.0 – 192.168.255.255

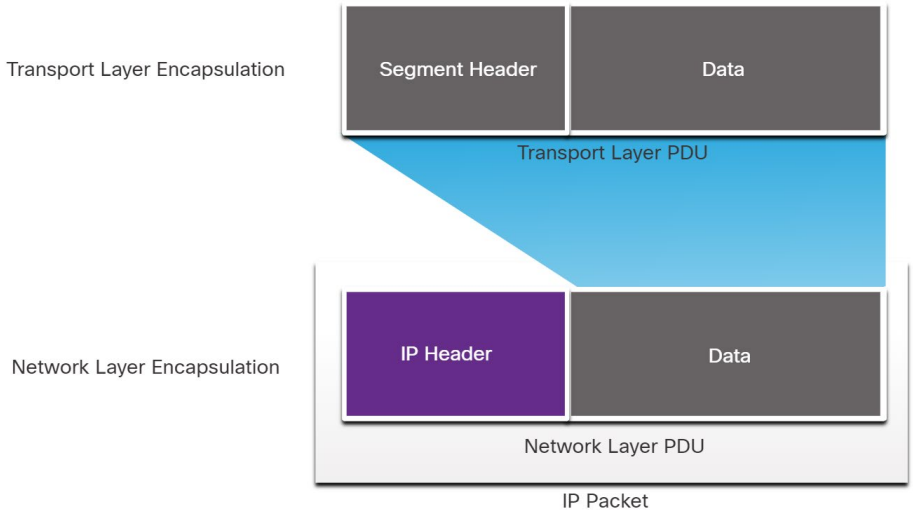
2.2. Az IPv4 protokoll működése és felépítése

Az IPv4 protokoll az OSI modell szerinti 3. rétegben foglal helyet, vagyis a hálózati rétegben. A hálózati réteg lehetővé teszi a távoli végpontok közötti kommunikációt hálózatokon keresztül. A végponttól végpontig tartó kapcsolat eléréséhez a hálózat rétegbeli protokollok be- és kicsomagolják a szomszédos rétegektől érkező adatot. Emellett szükséges, hogy a hálózatokon minden végpont egyedi IP címmel rendelkezzen, valamint, hogy a csomagok forgalomirányítása megtörténjen a hálózati rétegben. A forgalomirányítást routerek végzik, amelyek segítségével a csomagok több hálózaton keresztül haladva érhetnek el a célhoz. A forgalomirányítás keretein belül a csomagok több routeren haladnak át és a routerek megpróbálják kiválasztani a csomagok számára a cél felé vezető legideálisabb útvonalat.

A becsomagolás során a hálózati réteg a felette lévő szállítási rétegtől kapja meg a protocol data unit-ot (PDU), amit becsomagol és ezáltal elkészül egy csomag (packet). A küldő feladata a becsomagolás elvégzése, melynek során a meglévő PDU-hoz hozzáadja az IP fejlécet, amelyben szerepel a forrás és a cél IP címe.

A hálózati réteg esetében akkor történik a kicsomagolás, amikor a keretről már eltávolításra került a fejléc és a hálózati réteghez elérkezik a csomag. Amikor ez megtörtént, akkor a fogadó hoszt kiolvassa az IP fejlécből az adatokat és ha a fejlécben található cél IP címe megegyezik a saját IP címével, akkor a hoszt leválasztja az IP fejlécet a csomagról és az így kapott szegmenst továbbküldi a szállítási rétegbeli szolgáltatásnak.

Azért előnyös, hogy az adatokat rétegenként becsomagolják, mert így a különböző rétegekben lévő szolgáltatások nem függenek a felettük vagy alattuk lévő rétegtől, így mindegyik réteg külön fejleszthető és skálázható. Ez azt jelenti, hogy egy felülről érkező adatot becsomagolhatunk bármilyen új protokoll fejlécével, az nem fogja érinteni a felette lévő réteg PDU-ját.



2.1. ábra: Hálózati rétegben történő becsomagolás

2.2.1. Az IPv4 protokoll karakterisztikái

Az IP protokoll megtervezésénél fontos szempont volt az, hogy ne járjon sok többletterheléssel a csomagok célbajuttatása. A protokoll egyetlen célja, hogy összekapcsolt hálózatokon keresztül egy csomag eljusson a küldőtől a fogadóig. Ezáltal elmondható róla, hogy nem követi nyomon minden egyes csomag útját és azt, hogy azok megérkeztek-e a célponthoz. Ezeket a feladatokat a szállítás rétegbeli TCP protokoll látja el.

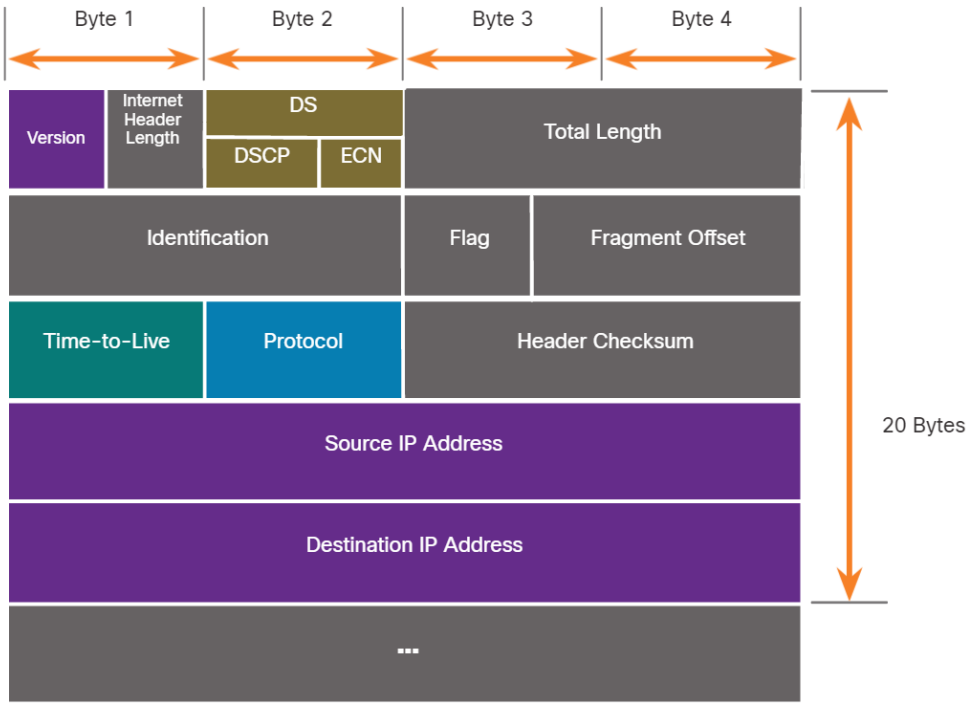
Három csomagtovábbítási módszer lehetséges az IPv4 esetében, ezek az unicast, multicast és a broadcast címzések. Mindegyik címzési módszer esetében a forrás címe csak egyetlen hoszt címe lehet, azaz egy unicast cím. Unicast címzés esetében „egy-az-egyhez” kapcsolatról beszélhetünk, azaz a küldött csomagot egyetlen célpont kapja csak meg. Multicast címzésről akkor beszélhetünk, ha célpontnak hosztok egy csoportja van megadva, tehát ha egy „egy-a-többhöz” típusú kommunikáció valósul meg. Ez a csoport egy multicast csoport, amelyben a kiválaszott hosztok akkor vesznek részt, ha azok feliratkoznak a definiált csoportokra. Minden multicast csoport egyetlen multicast IPv4 címmel címezhető meg, ezek a címek a 224.0.0.0 – 239.255.255.255 címtartományából kerülhetnek ki. Ez a címzési módszer például az OSPF dinamikus forgalomirányító protokoll működése során kerül használatra. A broadcast módszer alkalmazása során a küldő hoszt az alhálózati szegmensen lévő összes hosztot címezi meg, ily módon egy „egy-az-összeshez” típusú kommunikáció valósul meg. Egy broadcast csomag esetében a célpont IP címének minden bitje „1” (255.255.255.255) vagy csak az IP cím gépazonosító részének minden bitje 1-es. Az előbbi a limitált broadcast, míg az utóbbi az irányított. Alapértelmezetten a routerek nem továbbítanak broadcast üzeneteket.

Általában három fő tulajdonsággal jellemzik az IP protokollt, ezek a következőek: összeköttetés-mentes, legjobb szándékú, átviteli közegtől független. Összeköttetés-mentes alatt azt értjük, hogy a protokoll a cél és a forrás között nem alakít ki konkrét dedikált útvonalat mielőtt adatcsomagokat küld. Mivel nincs kiépített dedikált útvonal, így a küldő nem tudhatja, hogy a célpont valóban létezik és hogy egyáltalán megkapja a küldött csomagokat vagy azt, hogy a fogadó képes ezen csomagok olvasására. Ezt a tulajdonságát hívjuk úgy, hogy legjobb szándékú, hiszen nem garantálja, hogy a csomagokat megkapja valaki, de ettől függetlenül elküldi azokat. Ezentúl az IP protokoll közegfüggetlen, tehát nem számít, hogy milyen átviteli közegen haladnak át az adatcsomagok, ez lehet például digitális jel formájában rézkábeleken vagy fényimpulzusok formájában optikai kábeleken, vagy akár vezeték nélkül rádiójelek formájában is.

Ezek a karakterisztikák mellett elmondható, hogy az IPv4 protokoll megbízhatatlan, mert ha egy-egy csomag megsérül szállítás során vagy nem érkezik meg a célhoz, akkor nem tudja újraküldeni az érintett csomagokat, hiszen nincsen integrálva benne egy olyan mechanizmus, amely értesíthetné a küldőt, ha hiba történne. Ennek következtében a csomagok megérkezhetnek rossz sorrendben vagy akár az is lehetséges, hogy egyáltalán nem érkeznek meg. [12]

2.2.2. IPv4 csomag fejléce

Az IPv4-es fejléc fontos információkat tárol a csomagról, amelyeket a harmadik rétegbeli szolgáltatások kezelhetnek. A fejlécben elhelyezkedő mezőkben binárisan vannak tárolva az adatok. Ezek a bináris értékek az IP csomag különféle beállításainak feleltethető meg.



2.2. ábra: Az IPv4 csomag fejlécének mezői

A protokoll fejlécéről készült diagram balról jobbra, majd fentről lefelé olvasandó. Az ábrán látható mezők a következőek:

- **Version:** Egy 4 bites bináris érték, amely az IP protokoll verzióját jelöli, IPv4 esetében 0100 az értéke binárisan.
- **Internet Header Length:** Ez a 4 bites mező adja meg az IP fejléc hosszát. Egy IP fejléc hossza minimum 20 bájt és maximum 60 bájtos lehet.
- **Differentiated Services (DS):** Egy 8 bites mező, amely a korábban definiált ToS (Type of service) mező helyét vette át, aminek a segítségével megállapítható volt egy csomag prioritása. A Differentiated Services mező segítségével biztosíthatjuk egy adott adatforgalom típus által elvárt műszaki paramétereket (csomagvesztés, késleltetés, jitter stb.). A mező első 6 bitje a DSCP (Differentiated Services Code Point) bitek, az utolsó kettő pedig az ECN (Explicit Congestion Notification) bitek.
- **Total Length:** Ez a 16 bites mező írja le egy IP csomag teljes méretét, a csomag fejlécét és a becsomagolt adatot is beleértve.
- **Identification:** Egy 16 bites egyedi azonosító, amely azonosítja az ugyanahhoz csomaghoz tartozó darabokat, amelyek a feldarabolás során keletkeztek.
- **Flag:** 3 bites érték, amellyel a feldarabolás karakterisztikáit tudjuk testre szabni. Az első bit értéke mindig nulla. A második bittel (DF: Don’t fragment) beállítható, hogy a csomag egyáltalán ne legyen feldarabolva. Ha egy csomagot feldarabolunk, akkor az összes darab esetében a harmadik bit (MF: More fragments) értéke egy, kivéve az utolsóét, ezzel megjelölve a csomag utolsó darabját.
- **Fragment Offset:** Ez a 13 bites mező kijelöli egy töredék pozícióját az egyazon csomaghoz tartozó töredékekhez képest.
- **Time-to-live (TTL):** Egy 8 bites bináris érték, amely korlátozza az IP csomagok élettartamát. Az IPv4 csomag küldője állítja be a csomag kezdeti TTL értékét. Ez az érték minden egyes alkalommal egyel csökken, amikor a csomag áthalad egy routeren. Abban az esetben, ha egy routerhez érkező csomag nullás TTL értékkel rendelkezik, akkor a router eldobja a csomagot és értesíti a csomag küldőjét egy ICMP (Internet Control Message Protocol) üzenettel, hogy a csomag nem ért célba, hiszen a TTL mező nullára redukálódott.
- **Protocol:** Ez a 8 bites mező határozza meg, hogy milyen szállítás rétegbeli protokoll van használatban. Például TCP protokoll esetében ez a szám 6, UDP esetében pedig 17.
- **Header Checksum:** Ebben a 16 bites mezőben a fejléchez tartozó ellenőrzőösszeg található. Segítségével megállapítható, hogy a fejléc hibás vagy sem.
- **Source IP Address:** Ebben a 32 bites mezőben található meg a küldő IP címe. A küldő IP címe mindig egy unicast cím.
- **Destination IP Address:** Ebben a 32 bites mezőben található meg a fogadó IP címe. A célként megjelölt cím lehet unicast, multicast vagy broadcast cím.

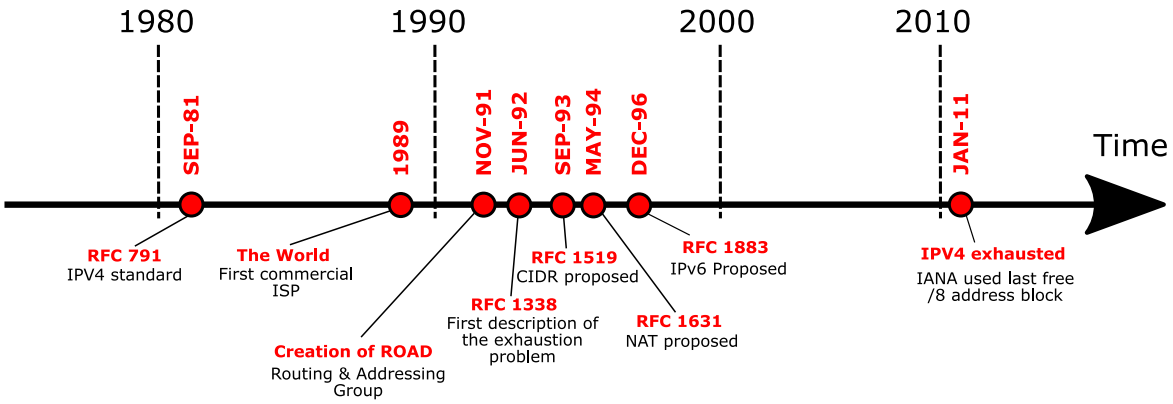
Ezek azok a mezők, amelyek kötelezően szerepelnek minden egyes IP fejlécben. Egy fejléc akkor lehet hosszabb 20 bájtnál, ha az IP Options mezőt is használjuk. Ennek a mezőnek a használata teljesen opcionális, segítségével a küldő például kérelmezheti, hogy a csomag milyen útvonalon haladjon a cél felé. [12][13]

2.3. IPv4 címek kimerülése

Már évtizedekkel ezelőtt nyilvánvalóvá vált, hogy előbb-utóbb elfogynak majd az IPv4-es címtérből kioszthatók címek. Az IPv4-es címek fogyását nagyban felgyorsította az okostelefonok elterjedése és az internet használatára való megnövekedett igény olyan területeken is, ahol korábban az nem volt jellemző. Ennek következtében a kioszthatók IPv4-es címek rohamosan elkezdtek fogyni, amit különféle technikák bevezetésével igyekeztek orvosolni.

Elsőként 1992-ben a CIDR (Classless Inter-Domain Routing) bevezetésével próbálták meg lelassítani a kimerülés folyamatát. A technika alkalmazásával megszűnt a korábban használatos osztályos IP címzés. Használata azt jelentette, hogy az ISP-k eltérhettek a hagyományos A osztályos (/8 prefix), B osztályos (/16 prefix), C osztályos (/24 prefix) hálózati címzésektől. A prefix megadja azoknak a biteknek a számát, amelyek egy IP címben a hálózatot azonosítják. Ennek következtében tényleges igények alapján határozhatták meg a használt prefixet, így megspórolva olyan címeket, amelyeket nem vettek volna használatba a felhasználók. Emellett a CIDR lehetővé tette az összevont útvonalak használatát, amely növelte a forgalomirányítás hatékonyságát.

A pár évvel később megjelenő hálózati címfordítás (Network Address Translation) is azzal a céllal jött létre, hogy lelassítsa az IPv4-es címek fogyását. A hálózati címfordítás során egy vagy több publikus IP címet rendelnek egy vagy több privát alhálózathoz. Ez gyakorlatban azt jelenti, hogy egy privát IP címeket felhasználó alhálózatban a résztvevő hálózati eszközök egy közös publikus IP címen osztoznak és ezen keresztül érhetőek el az interneten. Ennek következtében kevesebb publikus IP cím szükséges ahhoz, hogy egy privát alhálózat tagjait elérjük.



2.3. ábra: Az IPv4-es címtér kimerülésével kapcsolatos események idővonala

Az IP címtér kezeléséért és az IP címek kiosztásáért globálisan az IANA felel, ezen felül létezik öt másik regionális, a kiosztott IP címek bejegyzéséért felelős szervezet (Regional Internet Registry). Az IANA által kezel címtartományból kioszthatő IPv4 címek 2011-ben fogytak el, ez a többi regionális szervezetnél is bekövetkezett pár éven belül.

Regionális szervezet	**Kiszolgált régiók**	**Kimerülés időpontja**
Asia Pacific Network Information Centre (APNIC)	Ázsia, Csendes-Óceáni térség	2011 április
Latin America and Caribbean Network Information Centre (LACNIC)	Latin-Amerika, Karib-térség	2014 július
American Registry for Internet Numbers (ARIN)	Amerikai Egyesült Államok, Kanada, Antarktika	2015 szeptember
African Network Information Centre (AFRINIC)	Afrika	2017 április
Réseaux IP Européens Network Coordination Center (RIPE NCC)	Európa, Oroszország, Nyugat-Ázsia	2019 november

2.2. táblázat: IPv4 címtartományok kimerülésének ideje az egyes regionális bejegyző szervezeteknél

Ezek alapján kijelenthető, hogy nincsenek további lefoglalásra váró IPv4 címek, de ennek ellenére a hálózati címfordítás jelentősen csökkentette a használatban lévő publikus IP címek számát, ezzel elnyújtva az IPv4-es címek kimerülését. Viszont a NAT, a pozitívumai mellett rendelkezik olyan hátrányokkal, amelyek nem kiküszöbölhetőek teljes mértékben, így szükséges volt egy másik megoldás bevezetése. Végső megoldásként az IP protokoll következő verzióját, az IPv6-ot kezdték el fejleszteni 1996-ban, ami jelentősen nagyobb címtérrel rendelkezik, mint az elődje és nem szükséges hálózati címfordítást alkalmazni. [12][14]

3. INTERNET PROTOCOL VERSION 6 (IPV6)

Az 1990-as évek elején az IETF (Internet Engineering Task Force) elkezdte az IPv4 protokoll utódjának a kifejlesztését. Ennek következtében megalapították az IPng (Internet Protocol Next Generation) nevű csoportot, akiknek a feladata a különféle megoldások értékelése volt. A megoldási javaslatok közül az IPv6-ot találták a legmegfelelőbbnek, így a választásuk erre a protokollra esett. A fejlesztés során az elsődleges szempont az IP protokoll címterének méretét övező problémák elhárítása volt, de emellett további funkcionalitással tervezték ellátni az új protokollt. Ez elsősorban annak a függvénye volt, hogy az eredeti probléma megoldása mellett marad-e elég idő arra, hogy a protokollt további új funkciókkal lássák el. Az 1994-es becslések szerint az IPv4-es címek elfogyását 2005 és 2011 közé jósolták, így maradt elegendő idő a fejlesztések elvégzésére. Az RFC 1883-ban definiálták az Internet Protocol Version 6 nevű protokollt 1995-ben. [9]

Az új protokoll legfontosabb változtatása az IP címek címterének kibővítése volt. A protokoll korábbi verziójában alkalmazott 32 bites címtér, az IPv6-ban 128 bitre növekedett. Ez a mennyiség bőven meghaladta a jelenlegi szükségleteket, ezáltal lehetővé tette, hogy az internethez csatlakozó eszközök mind egyedi címmel rendelkezzenek. A megnövekedett címtartomány mellett a címeket hierarchikusan építették fel, ezáltal effektívebbé téve a globális forgalomirányítást. További változtatás az automatikus konfiguráció, amelynek segítségével az az alkalmas hosztok globális IP címet generálhatnak egy IPv6-os routertől kapott hálózati prefix és a hosztok által véletlenszerűen generált adatok alapján. Emellett leegyszerűsítették az IPv6-os csomagok fejlécének a felépítését, a hosszát állandó 40 bájtban határozták meg, amelyből 16-16 bájtot a küldő és fogadó IPv6-os címe foglal el és a maradék 8 bájtban általános információkat tárolnak. Ha további opcionális mezők használatára lenne szükség, akkor azok az IPv6 esetében nem a fő fejlécben tárolódnak, hanem úgynevezett kiegészítő fejlécekben. [15]

3.1. IPv6 címek felépítése

Az IPv6-os címek 128 bit hosszú logikai címek, ami azt jelenti, hogy összesen 2^{128} (340,282,366,920,938,463,463,374,607,431,768,211,456) cím áll rendelkezésre. A 128 bites címek nyolc darab 16 bites hexadecimális blokkokra, más néven hextettekre oszlanak, amiket kettőspontok választanak el egymástól. Az IPv6-os címek nem érzékenyek kis- és nagybetűkre, tehát bármilyen írásmód megfelelő.

A könnyebb kezelhetőség érdekében lehetőség van a címeket lerövidíteni bizonyos szabályok mentén. A szabályok megengedik, hogy a blokkokból elhagyjuk a vezető nullákat és azt, hogy az egymást követő nullás blokkokat vagy a címen belüli vezető vagy záró nullákat kettő darab kettősponttal helyettesítsük. Az utóbbit egyszer tehetjük meg egy IPv6 címen belül, mert csak így kikövetkeztethető az eredeti cím.

Teljes IPv6 cím	Rövidített IPv6 cím
2001:DB8:0000:0000:0202:B3FF:FE1E:8329	2001:DB8::202:B3FF:FE1E:8329

3.1. táblázat: Lerövidített példa IPv6 cím

Az IPv4-ben használt CIDR jelölési forma – tehát amikor egy prefix számmal adjuk meg az IP cím hálózatazonosító részét – az IPv6 esetében is használatos. Az IPv6-os címet követően egy perjellel elválasztva megadhatjuk a prefix hosszát. A prefix hossza nagyon gyakran 64 bit, ebben az esetben az IPv6 cím maradék 64 bitje az interfész azonosítóját jelöli ki.

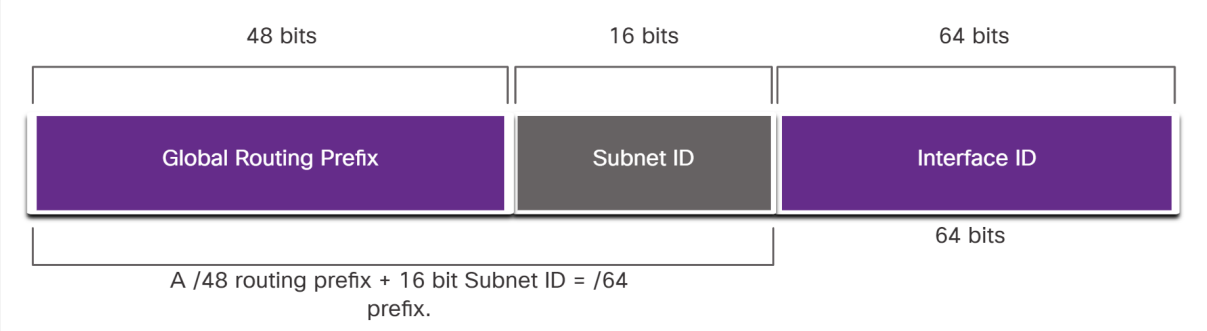
Az IPv6 esetében is három fő címzési módszert különböztetünk meg, ezek az unicast, multicast és anycast. Különbség az IPv4-hez képest az, hogy nem létezik broadcast címzési módszer, viszont lehetőséget ad az anycast címzésre. Egy unicast IPv6 cím egyértelműen kijelöli egy IPv6-os eszköz egyik interfészét. Ezek az interfészek tipikusan legalább kettő unicast címmel rendelkeznek, Global Unicast és Link-local címekkel. A multicast üzenetek segítségével itt is egy „egy-a-többhöz” típusú kapcsolat valósítható meg, az IPv6 protokollnál az ilyen típusú üzenetek gyakoribbak. Végül az anycast, amely egy olyan cím, ami több eszköznek is kiosztható, tehát több eszköz osztozik egy közös unicast címen. Abban az esetben, ha erre a címre érkezik egy csomag, akkor ezt a csomagot csak a legközelebbi eszköz kapja meg, ami a megadott IP címmel rendelkezik. [12]

3.1.1. Global Unicast IPv6 címek

A Global unicast IPv6 címek leginkább a publikus IPv4 címekhez hasonlíthatóak. Mindegyik cím egyedi és az internetes forgalomirányításban részt vehetnek. A publikus IPv4-es címekhez hasonlóan ezeket a címeket is az IANA és a további öt regionális szervezet kezeli és osztja ki. Jelenleg a global unicast címek csak a 2000::/3 tartományból kerülhetnek ki.

A globálisan használt unicast IPv6 címek három részre bonthatóak fel: Global routing prefix, subnet ID (alhálózati azonosító) és Interface ID (interfész azonosító) részre. A Global routing prefix részt az internet- és más szolgáltatók osztják ki a szervezetek számára, gyakran 48 bites global routing prefix címet alkalmaznak, de ez eltérhet szolgáltatónként. Mivel a Global routing prefix és a Subnet ID adja együttesen meg a prefixet, így a választott global routing prefix hossza meghatározza a Subnet ID hosszát.

Az IPv4-gyel ellentétben itt nem az IP cím gépazonosító részéből kell elvenni a biteket, hogy alhálózatokat formáljunk, hanem erre a célra dedikált bitek (Subnet ID) vannak fenntartva.



3.1. ábra: Tipikus global unicast IPv6 cím felbontás

Az ábrán látható tipikus felosztás esetén egy IPv6 cím első négy hextett-je az alhálózatot azonosítja a maradék négy pedig egy gép hálózati csatolóját. Az interfész azonosító rész gyakran egy 64 bit hosszú cím, amely egy alhálózat hosztjait egyértelműen azonosítja. [12]

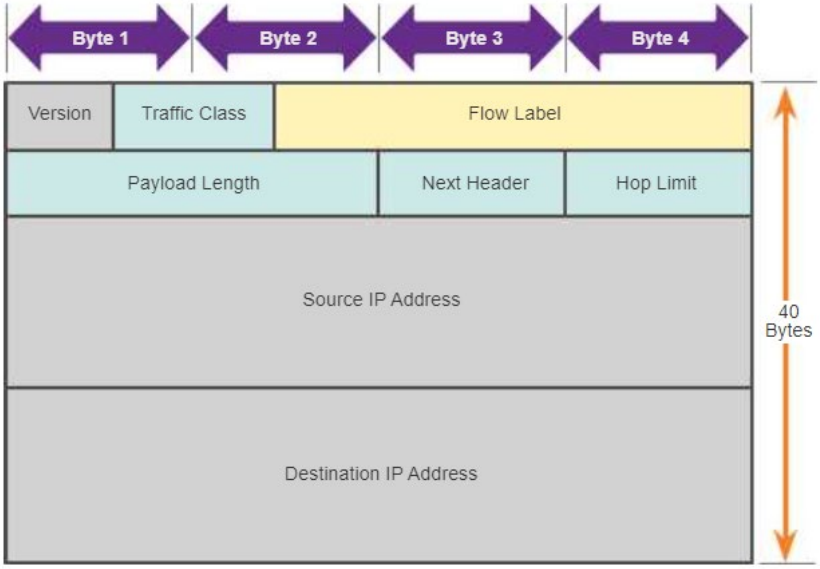
3.1.2. Link-local IPv6 címek

Az IPv6-os link-local címek lehetővé teszik az IPv6-os protokollt használó eszközök számára, hogy azonos alhálózaton belül egymással kommunikáljanak. A csomagokat, amelyekben forrásként vagy célként link-local címek vannak megadva, azokat a routerek nem továbbítják azon az alhálózaton kívülre, ahonnan érkezett a csomag. A link-local címek a fe80::/10 tartományból kerülhetnek ki.

Minden IPv6-os hosztnak rendelkeznie kell egy link-local címmel, ezzel ellentétben globális unicast címek használata nem kötelező. Ha manuálisan nem állítunk be link-local címet, akkor a hosztok dinamikusan készítik el maguknak azt az EUI eljárással, amely során a kliens, a MAC címét felhasználva generál egy link-local címet. [12]

3.2. Az IPv6 fejléc felépítése

Az IPv6-os fejlécből az IPv4-eshez képest az Internet Header Length, Identification, Flags, Fragment Offset, Header Checksum mezőket távolították el. Az Internet Header Length mező, azért került eltávolításra, mert az IPv6-os fejlécek hosszát állandó 40 bájtban határozták meg, míg ez a szám IPv4 esetében 20 és 60 bájt közé eshet attól függően, hogy az opcionális mezőt igénybe vesszük vagy sem. Az IPv6-os fejlécben nincsen definiálva IP Options mező, a helyét a kiegészítő fejlécek vették át. Az Identification, Flags és Fragment Offset mezők mind a csomagok feldarabolásával kapcsolatos beállítások, ezek IPv6 esetében átkerültek a kiegészítő fejlécekbe. Ha szükséges a feldarabolás, akkor a küldő hoszt ezeket az értékeket egy kiegészítő fejlécben beállítja. A Checksum mező eltávolításával a csomagok feldolgozása gyorsabbá vált, mivel nem szükséges minden egyes routernek kiszámolna a csomagok új ellenőrzőösszegét az áthaladások során. Mivel a szállítás rétegbeli protokollok is elláthatják ezt a feladatot, így az IPv6 protokollban az ellenőrzőösszeg számítást a TCP és UDP protokollok látják el kötelezően.



3.2. ábra: Az IPv6 csomag fejléce

A fejlécben a Traffic Class mező felváltotta a korábbi Type of Service/Differentiated Services mezőt, ezek az IPv6-ban másképp vannak megvalósítva. Az IPv4 fejlécben szereplő Protocol Type mezőt átnevezték Next Header-re és a Time-to-live mezőt pedig Hop Limitre. Ezek mellett bevezetésre került egy új mező, a Flow Label.

- **Version:** 4 bites mező, megadja az IP protokoll verzióját, jelen esetben ez a szám 6.
- **Traffic Class:** 1 bájtos mező, az IPv4-hez hasonlóan itt is megadhatóak, a valós-idejű továbbítást igénylő adatoknak a speciális paraméterei. Használatával az érintett routerek megkülönböztethetik az IPv6-os csomagokat osztályok és prioritás szerint.
- **Flow Label:** Ez a 20 bites mező azonosítja, azokat a csomagokat, amelyek ugyanazon adatfolyamnak a részei. Használatával a routerek beazonosíthatják az ugyanahhoz az adatfolyamhoz tartozó csomagokat, ezáltal effektívebbé téve a továbbítást, ugyanis nem szükséges minden csomag fejlécét újra feldolgozni. A küldő és a fogadó IP címe, valamint a Flow Label mező egyértelműen azonosítják az egy adatfolyamhoz tartozó csomagokat.
- **Payload Length:** Egy 2 bájtos mező, amely megadja az IPv6-os csomag payload-jának a méretét. A payload az a része egy csomagnak, ami az adott rétegben beillesztett fejléc után következik. A kiegészítő fejlécek szintén a payload részét képzik, így azok is beleszámítanak a méretbe. Különbség az IPv4-es csomagokkal szemben, hogy az ott használt Total Length mezőben a csomag teljes méretét adják meg, tehát a számolásba beletartozik a fejléc hossza is.
- **Next Header:** 1 bájt hosszú mező, amely az IPv4-es Protocol Type mezőt váltotta fel. Alapesetben az eredetihez hasonlóan itt is a felsőbb rétegbeli szállítási protokollok adhatóak meg a hozzájuk tartozó számokkal, azonban, ha használunk IPv6-os kiegészítő

fejléceket, akkor ebben a mezőben a következő kiegészítő fejléc típusához tartozó szám fog megjelenni. A kiegészítő fejlécek az eredeti IPv6 fejléc és a TCP/UDP fejlécek között találhatóak meg.

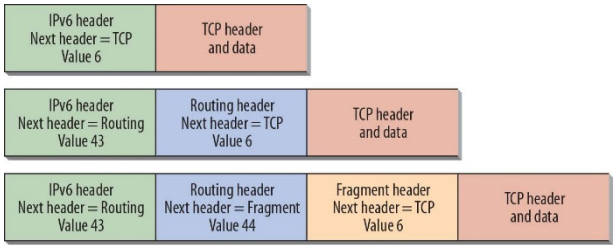
- **Hop Limit:** 1 bájt hosszú mező, amely ugyanúgy működik, mint az IPv4 fejlécében használatos Time-to-live mező. Eredetileg az IPv4-es TTL mezőben másodpercben kifejezve adták meg, hogy egy csomagot meddig továbbíthatnak a routerek, mielőtt eldobnák azokat. Viszont később ezt az értéket routerek közötti ugrásszámban adták meg, így az IPv6-os fejlécben ezt a mezőt átnevezték Hop limitre, hogy az jobban tükrözze a mezőnek a funkcióját. [15]

3.2.1. Kiegészítő fejlécek

Az IPv4-es fejlécek minimum 20 és maximum 60 bájtosak lehetnek. Egy IPv4 fejléc csak akkor lehetett hosszabb 20 bájtnál, ha használták az IP Options mező beállításait pl.: Biztonsági beállítások, időbélyegzés. Ezek a paraméterek nagyon ritkán voltak használatosak, mivel jelentősen visszavetette a csomagok továbbításának a sebességét.

Ezt a problémát úgy oldották meg, hogy maga a fő fejléc viszonylag kis méretű és ha az opcionális mezőket szeretnénk használni, akkor azoknak a mezőknek az értékeit a kiegészítő fejlécekben adhatjuk meg. A kiegészítő fejlécek csak akkor kerülnek beszúrásra egy csomagba, ha szükségesek és gyakran csak a célállomás dolgozza fel őket, a közbülső eszközök pedig nem. Ennek következtében az IPv6-os csomagok feldolgozása jelentősen gyorsabb. A jelenlegi IPv6 specifikáció hat darab kiegészítő fejlécet határoz meg, amit az IPv6-os eszközöknek tudniuk kell kezelni. Ezek a Hop-by-hop Options, Routing, Fragment, Destination Options, Authentication és Encapsulating Security Payload fejlécek. Egy IPv6-os csomagban több kiegészítő fejléc is alkalmazható, az egymás után következő fejlécek a Next Header mező segítségével találhatóak meg. A kiegészítő fejlécek feldolgozása szigorúan olyan sorrendben történhet, ahogyan azok a csomagban szerepelnek. A Hop-by-hop Options kiegészítő fejléc használata esetén ez a fejléc kell legyen az első beszúrt elem a fő fejléc után, továbbá kivételt képez abban is, hogy ezt a típusú kiegészítő fejlécet minden köztes eszköznek fel kell dolgoznia.

A kiegészítő fejlécek által nyújtott rugalmassággal, a jövőben fejlesztett kiegészítő fejlécek is hozzáadásra kerülhetnek az IPv6-os csomagokhoz anélkül, hogy az eredeti fejlécet módosítani kellene. [15]



3.3. ábra: A kiegészítő fejlécek felépítése

3.3. Neighbor Discovery Protocol

Az NDP (Neighbor Discovery Protocol, Szomszéd felderítő protokoll) protokoll segítségével a hosztok feltérképezhetik a helyi hálózatokon található routereket és a további fontos csomópontokat. Ezektől az eszközöktől a hosztok az internet használatához fontos konfigurációs információkat szereznek be. A Neighbor Discovery Protocol üzenetei több feladatot is megvalósítanak, ezek:

- IPv6 címek automata konfigurációja SLAAC-kel (Stateless Address Autoconfiguration)
- A hosztok beszerzik a hálózatban használatos prefixet, útvonalakat és egyéb konfigurációs információkat
- IPv6 címduplikáció ellenőrzése (Duplicate Address Detection, DAD) egy alhálózatban
- A hosztok megismerhetik az alhálózatban lévő más hosztok adatkapcsolati rétegbeli címét
- Számontartja, hogy mely szomszédok elérhetőek és melyikek nem (Neighbor Unreachability Detection, NUD)
- Detektálja a második rétegbeli címváltozásokat

Az NDP öt különböző ICMPv6 üzenettípust definiál: Router Solicitation (RS, router keresés), Router Advertisement (RA, router hirdetés), Neighbor Solicitation (NS, szomszéd keresés), Neighbor Advertisement (NA, szomszéd hirdetés) és a Redirect (Átirányítás) üzenetet. Ezek az üzenetek a következő funkciókat valósítják meg:

- **Router Solicitation:** A hosztok ilyen típusú üzeneteket küldenek a hálózatban lévő összes routernek, amelyben kérik, hogy a routerek RA üzeneteket küldjenek a csomagban megjelölt címre. Erre a típusú üzenetekre azért van szükség, mert a routerek csak periodikusan küldenek RA üzeneteket és az RS üzenetek használatával a hosztnak nem kell várakoznia az RA üzenetek megérkezéséig.
- **Router Advertisement:** A routerek időközönként ilyen típusú üzeneteket küldenek ki a hálózatukon, illetve a Router Solicitation üzenetekre küldhetik válaszul, amelyekkel értesítik a hálózatban található hosztokat a jelenlétükről. Ezek mellett hirdetik az állapotmentes automata IP cím konfiguráció (SLAAC) folyamatához szükséges információkat. Ezeket az üzeneteket az FF02::01 multicast címet felhasználó multicast csoport (összes IPv6 eszköz) tagjai kapják meg.
- **Neighbor Solicitation:** Ezzel a típusú üzenettel a kliensek lekérdezhetik a címzettként megjelölt hoszt MAC címét. Emellett részt vesz a korábban aktív szomszédok állapotának megállapításában (NUD) és a címduplikáció detektáló folyamatban (DAD).
- **Neighbor Advertisement:** A kliensek az NA üzeneteket az NS üzenetekre küldik válaszként. Továbbá ezek az üzenetek akkor is kiküldésre kerülnek, ha változás történik a küldő cím konfigurációjában, valamint ebben jelezhetik a hosztok a DAD procedúra keretein belül, ha már alkalmazzák a lekérdezett IP címet.
- **Redirect:** Működésük megegyezik az ICMP protokollban található redirect csomaggal, segítségével hálózati forgalmat lehet átirányítani routerek között. [15][16]

3.3.1. Stateless Address Autoconfiguration (SLAAC)

A SLAAC (Stateless Address Autoconfiguration, Állapotmentes automatikus címkonfiguráció) egyike az IPv6 protokoll új funkcióinak. A funkció segítésével egy hálózat hosztjai automatikusan konfigurálhatnak IPv6 címeket saját maguk számára. Az IPv4 protokoll esetében a címeket manuálisan lehetett megadni vagy DHCP szerverek használatával szerezhettek be a hosztok IPv4 címeket.

Az IPv6 címek automatikus konfigurációja két féle módon történhet, állapottartó (stateful) és állapotmentes (stateless) konfigurációban. A SLAAC segítségével az állapotmentes konfiguráció valósítható meg. Használatával a hosztok manuális konfigurációjára nincsen szükség, csak a routerek igényelnek minimális konfigurációt, viszont külön szerverre nincsen szükség. A hosztokon helyben elérhető és a routerek által terjesztett információk alapján képesek generálni címeket saját maguk részükre. A routerektől kapják meg a konfigurációhoz szükséges prefixet, amely definiálja, hogy melyik alhálózathoz tartoznak, valamint ezen kívül egyéb, a hosztokat érintő paramétereket (MTU, cím élettartam, konfigurációs zászlók). A hosztok a hálózati prefixet kiegészítik az általuk generált interfész azonosító címmel, a két címet kombinálva a hosztok képesek lesznek a Global unicast IPv6 címük aktivációjára. Abban az esetben, ha az alhálózatban nincsen jelen olyan router, amely prefixeket terjesztene, akkor a hosztok csak link-local címeket fognak tudni generálni. A link-local címek segítségével csak az alhálózatban lévő eszközökkel képes kommunikálni egy hoszt.

Az állapottartó konfiguráció a DHCPv6 szerverek segítségével valósítható meg. Alapvetően egy IPv6 cím háromféleképpen konfigurálható a két módszer segítségével. Az első esetben egyedül a SLAAC használatos, a DHCPv6 nem aktív, valamint a második eset ennek pont a fordítottja, azaz a DHCPv6 aktív, míg a SLAAC nem. A harmadik módszer a két protokoll együttműködését valósítja meg. Ebben az esetben egy hoszt az IPv6 címet a SLAAC módszerrel generálja, viszont a további információkat a router által megjelölt DHCPv6 szervertől szerzi be (például DNS szerver címe).

Az IPv6 címekkel a hosztok egy bizonyos időtartamig rendelkeznek, pontosabban a lifetime (élettartam) mezőben meghatározott ideig. Amikor ez az időzítő lejár akkor az érintett címek érvénytelenné (invalid) válnak. Ennek következtében a hosztoknak új címeket kell generálniuk, amelynek során a Duplicate Address Detection (Címduplikáció ellenőrzés, DAD) procedúra lép életbe. Segítségével a hosztok olyan IP címeket generálhatnak, amelyet a hálózatukban lévő hosztok még nem használnak.

Egy IPv6 címnek a következő állapotai lehetnek:

- **Tentative Address (Kérdéses cím):** Ilyen típusú IPv6 címek, amelyek még nem lettek ténylegesen aktiválva egy interfészen, viszont a címgeneráció már megtörtént. Tulajdonságuk, hogy a DAD procedúra által még nem lettek jóváhagyva, azaz nem megerősíthető az egyéniségük. Egy ilyen címmel rendelkező hoszt nem képes rendes kommunikációra a hálózatban, de ezek a címek használatosak az automata konfigurációs folyamat során. Csakis a SLAAC folyamán felhasznált Neighbor

Discovery üzeneteket lehet feldolgozni az ilyen típusú címekkel, minden más erre a címre érkező csomagot eldobnak a hosztok.

- **Preferred Address (Preferált cím):** Ezeknek a típusú címeknek az egyediségét már korábban megerősítettek az autokonfiguráció során és megfelelően hozzárendelésre kerültek az adott interfészekhez. Limitációk nélkül használhatóak, egészen az élettartamuk lejáratáig.
- **Deprecated Address (Elavult cím):** Az elavult címek használata nem ajánlatos, de további használatuk nem tiltott. Egészen a cím tényleges élettartamáig használhatóak, a meglévő kapcsolatokat nem szakítja meg, viszont új kapcsolatok kialakítására már nem alkalmas.
- **Valid Address (Érvényes cím):** Érvényes címnek nevezik mind a Preferred és Deprecated címeket.
- **Invalid Address (Érvénytelen cím):** Egy érvényes cím, akkor válik érvénytelenné, amikor annak letelik az élettartama. Amint egy cím érvénytelenné válik, a továbbiakban interfészhez nem rendelhető. [15]

3.3.2. Duplicate Address Detection (DAD)

A Duplicate Address Detection az előző fejezetben említett címduplikációt ellenőrző metódus. Működése szorosan összekapcsolódik a SLAAC operációjával. Használatával a hosztok csak egyedi címeket generálhatnak.

Az automata konfiguráció lépéseit az alábbi táblázat részletezi:

Lépés	Művelet
Link-local címgenerálás	Az első lépés során egy link-local cím generálódik az fe80-as prefix-szel, amelyhez a hosztok hozzácsatolják a generált interfész azonosítójukat. Az így kapott link-local cím egy tentative address lesz.
Csatlakozás a multicast csoportokhoz	A hoszt a következő multicast csoportokhoz csatlakozik: all-nodes (FF02::1) és a kérdéses cím solicited-node multicast csoportjához.
DAD procedúra	Egy Neighbor Solicitation üzenet kerül kiküldésre, amiben a célcím mezőben a kérdéses cím van feltüntetve. Ezeknek a csomagoknak a küldő IP címeként végig-nullás cím (::) van szerepeltetve, a fogadó címeként pedig a kérdéses cím solicited-node multicast címe. Ilyen üzenetek segítségével valósul meg a DAD folyamat, amivel ellenőrizhető, hogy a cím használatban van vagy sem. Ha egy hoszt rendelkezik a releváns címmel, akkor azt egy Neighbor Advertisement üzenetben jelzi. Abban az esetben, ha az interfész azonosító cím generálása az EUI-64 eljárással történt, akkor a procedúra leáll. Ha a generált interfész azonosító, azonban véletlenszerűen generált számokból állt, akkor a Neighbor Advertisement üzenet hatására újabb címeket fog generálni,

	ennek pontos menete operációs rendszer függő. Ha több alkalommal sem minősül egyedinek a cím, akkor ez a cím a továbbiakban manuálisan lesz konfigurálható. Abban az esetben, ha a kérdésre nem érkezik válasz, azaz egyik hoszt sem használja ezt a címet, akkor a tentative address státusza preferred address státuszra változik. Eddig a pontig a hosztok és routerek automata konfigurációja megegyezik, viszont a következő lépést csak a hosztok hajtják végre.
Routerek felderítése	Annak érdekében, hogy a hosztok megtalálják a hálózaton aktív IPv6 routereket, Router Solicitation üzeneteket küldenek ki az FF02::2-es multicast címre, amelyre a routerek fel vannak iratkozva. A válaszban megismerik az alapértelmezett átjáróként alkalmazandó router IP címét és az általa terjesztett prefix-et, amely alapján a SLAAC folyamat elkezdődhet más címek esetében is.
Routerek válasza	A hosztok által küldött Router Solicitation üzenetekre a routerek Router Advertisement üzenetekkel reagálnak. Minden olyan üzenetben, ahol prefix van szerepeltetve, valamint az automata címgenerációra vonatkozó (Autonomous Configuration) zászló aktív, abban az esetekben a hosztok a prefix-et és a saját maguk által generált interfész azonosítókat kombinálva létrehozzák az aktív címüket.

3.2. táblázat: Automata konfigurációs folyamat menete

Minden címet szükséges a DAD procedúra segítségével megerősíteni. Abban az esetben, ha a további címek interfész azonosító része megegyezik az elsőként konfigurált link-local cím interfész azonosító címével, akkor a harmadik lépést (címduplikáció ellenőrzés) nem szükséges újra elvégezni, ha mindkét cím az automatikus címgeneráció folyamán készült. [15][16]

4. IPV6 BIZTONSÁG

Az IPv4 kifejlesztése során nem kaptak nagy szerepet a biztonsági tényezők, ugyanis az internet korai napjaiban csak olyan felhasználók – leginkább kutatók – fértek hozzá az internethez, akikről feltételezték, hogy nincsenek rosszindulatú szándékaik. Ebből az okból kifolyólag az eredeti IP infrastruktúra nem tartalmaz semmilyen biztonsági keretrendszert, amelyet minden alkalmazás igénybe vehetne. Ha a biztonság fontos szempont volt egy-egy folyamat során, akkor azt az alkalmazás rétegbeli protokollokba épített autentikációs és autorizációs megoldásokkal tudták megvalósítani. Sok évvel azután, hogy az IPv4 használata széleskörben elterjedt, megjelent az IPsec protokoll csomag, amelynek segítségével megvalósíthatóvá vált egy végpontól végpontig terjedő biztonságos kapcsolat felépítése a hálózati rétegben. Mivel az IPv4-et eredetileg nem készítették fel az IPsec használatára, így utólag kellett megoldani a kettő együttműködését, amely jelentős teljesítménybéli problémákat okozott, ezáltal nem terjedt el a használata. Ellentétben az IPv6-tal, ahol a kezdetektől fogva elvárt volt az alapvető biztonsági megoldások implementálása a protokoll alapjaiba. Az ezzel kapcsolatos szabványoknak megfelelően az IPv6-nak kötelezően rendelkeznie kellett IPsec implementációval, kiegészítő fejlécek formájában. [15]

Az IPsec implementálása mellett további biztonsági intézkedéseket tettek, amivel igyekeztek a következő IPv4-es sérülékenységeket is kiküszöbölni:

- **Fragmentációs támadások elleni intézkedés:** Olyan támadások, amelyek kihasználják azt, hogy egyes rendszerek hogyan kezelik a nagy IP csomagokat. Ezt a sérülékenységet úgy küszöbölték ki IPv6 esetében, hogy közbenső eszközök számára nem engedélyezett, hogy feldarabolják a szállított csomagokat, hanem ezt csak a feladó teheti meg ha igény van rá. Ezentúl, ha az összeállított csomagok kevesebb mint 1280 oktett hosszúak akkor az eszközök eldobják azt a csomagot, mivel minden adatkapcsolati rétegbeli protokoll képes kell legyen minimum 1280 bájt továbbítására, anélkül, hogy fel kellene darabolni a csomagokat. Ezeknek a szabályoknak a betartásával nagyobb védelem alakítható ki a fragmentációs támadások ellen.
- **Szórási viharok:** Az IPv4-ben jelenlévő szórási viharokat kiküszöbölték az IPv6 esetében. IPv6-ban nem érhető el broadcast címzési módszer, ezt helyettesítik a multicast és anycast módszerek. A szórásos támadások elhárítása érdekében az RFC 2463 azt javasolja, hogy ICMPv6-os válaszüzeneteket nem szabad küldeni válaszként olyan üzenetekre, amelyekben cél címnek egy multicast IP cím van megadva vagy egy második rétegbeli multicast vagy broadcast cím. [17]

4.1. Implementációs problémák

Tekintve, hogy a natív IPv6 implementációk viszonylag újak, így azzal a problémával kell szembenéznünk, hogy nem áll rendelkezésre elég olyan IPv6-ot támogató sérülékenységelemző eszköz, amellyel felmérhetnénk az IPv6-os hálózatunk biztonságát. Ezeket az eszközöket azzal a céllal használják, hogy megkeressék a hálózatokban az esetleges sérülékenységeket vagy hibás beállításokat, majd a felderített problémákat orvosolják. További probléma, hogy ha ezek

az eszközök mégis rendelkeznek IPv6 támogatással, akkor sem lehetünk abban biztosak, hogy megfelelően működnek, ugyanis még nem telt el elegendő idő ahhoz, hogy megbizonyosodhassanak a fejlesztők afelől, hogy a program megfelelően működik minden esetben. [15]

4.2. IPv6 sérülékenységek

Az IPv4 protokoll esetén használatos ARP (Address Resolution Protocol) protokoll funkcionalitását az IPv6 esetében az ICMPv6 protokoll látja el. Az NDP (Neighbor Discovery Protocol) protokollban is fellelhetőek az ARP protokoll biztonsági problémái, de ezek kiküszöbölhetőek az IPsec vagy a SEND (Secure Neighbor Discovery) protokollok használatával. Szóba jönnek például az átirányításos támadások (támadó eszközök megakadályozzák, hogy a csomagokat az eredetileg meghatározott címzettek megkapják), a Denial of Service (DoS, szolgáltatás megtagadással járó támadások) támadások és az elárasztásos támadások (jelentéktelen, nagy mennyiségű forgalom átirányítása az áldozat felé) is.

Ezek mellett a Duplicate Address Detection (DAD) és a Router Advertisement (RA) csomagok is felhasználhatóak rosszindulatú szándékokkal. Az ezeket a csomagokat felhasználó támadások hasonlóak az IPv4-es környezetben megvalósított ARP és DHCP protokollt kihasználó támadásokhoz. A DAD csomagok segítségével megállapítható, hogy egy azonos helyi hálózaton két különböző hoszt ugyanazzal az IP címmel rendelkezik, még mielőtt az IPv6 cím jóváhagyásra kerülne. [15]

4.2.1. Több kioszthető IP cím sérülékenység

Mivel egy hálózati interfész akármennyi IPv6 címmel rendelkezhet, így ezt kihasználva DoS támadásokat tudnak indítani a támadók. Ha például egy hoszt a DAD működése során azt állapítja meg, hogy a használni kívánt IP cím már használatban van az alhálózaton, akkor az IP cím már csak manuálisan adható meg. Ezt kihasználva egy támadó állomás ugyanabban az alhálózatban beállíthat magának több ezer címet és ezáltal lehetetlenné teszi, hogy a hosztok automatikusan link-local címeket kapjanak és emiatt nem lesznek alkalmasak IPv6-os kommunikációra. Ennek a kiaknázására készültek olyan IPv6-os hacker eszközök, amelyekkel ezt a hatást sokkal könnyebben el lehet érni, szimplán csak egy szoftverről van szó, ami olyan üzeneteket küld ki, amiben jelzi a hosztnak, hogy egy bizonyos cím már használatban van. [18]

4.2.2. EUI-64 sérülékenység

Egy IPv6-os cím SLAAC-kel (Stateless Address Autoconfiguration) való generálása során a felhasználó megkapja egy, az alhálózatában elhelyezkedő routertől a hálózatban használt prefixet, és abban az esetben, ha az intérfész azonosítót az EUI-64 folyamattal generálja a hoszt, akkor az interfész azonosítója könnyedén kitalálható lesz, ugyanis az EUI-64 módszer olyan interfész azonosítót generál, amelyben megjelenik a hoszt teljes 48 bites MAC címe. Ezáltal megállapíthatóvá válik egy hoszton telepített operációs rendszer típusa és ez alapján a támadók a felderített operációs rendszerhez kapcsolódó ismert sérülékenységeket használhatnak fel támadások indításához. [18]

4.2.3. Multicast sérülékenység

Az IPv6 protokoll nem rendelkezik broadcast csomagtovábbítási módszerrel, helyette multicast-tel valósíthatóak meg az „egy-a-többhöz” típusú kommunikációk. Például multicast címzést használnak az NDP és DHCP (Dynamic Host Configuration Protocol) protokollok is. Ha egy támadó adatcsomagokat küld egy multicast csoportnak és csoport tagjai válaszolnak az üzenetre, akkor a támadó a válaszokban található információkat felhasználhatja egy későbbi támadás során. A támadónak nem szükséges ismernie az eszközök pontos címeit, mivel csak a csoport multicast címét adja meg, ami széleskörben ismert. A támadó a visszaküldött adatok alapján több információt tudhat meg a hálózatban elhelyezkedő routerekről és a DHCPv6 szerverekről. Ezek az eszközök olyan adatokat tárolnak a hálózatról, amelyek alapján a támadó felderítheti a hálózat topológiáját. Ezeket az adatokat általában szomszédsági táblázatokból vagy DHCPv6 szerverek által generált naplófájlokból nyeri ki a támadó. [18]

4.2.4. Kiegészítő fejlécek sérülékenysége

Mivel az eredeti IPv6 protokoll specifikációja nem korlátozza a kiegészítő fejlécek használatát, így azokat rosszindulatú célokkal is fel lehet használni. A kiegészítő fejlécek egy láncolt listát alkotva szerepelnek a csomagban a fő fejléc után. Egy támadó készíthet egy olyan IPv6-os csomagot, ami megfelel a specifikációban leírtaknak és emellett közel végtelen hosszú kiegészítő fejlécek láncolatát tartalmazza. Egy ilyen csomag akár egy szolgáltatás megtagadással járó támadás hatását képes replikálni a köztes eszközökön, amelyeken a csomag áthalad a szállítás során. Ezentúl sok kiegészítő fejléc összeláncolásával megoldható, hogy a tűzfalak vagy hálózati behatolás érzékelő rendszerek ne detektálják a rosszindulatú forgalmat. A csomag megnövekedett mérete miatt a fragmentálás során az első darabba csak a kiegészítő fejlécek férnek el és maga a payload csak a második részlettől kezdő töredékekben jelenik meg, ezáltal kijátszva azokat a tűzfalakat, amelyek csak a csomagok első töredékét vizsgálják. [18]

4.2.5. Fragmentálási sérülékenység

A fragmentálás (feldarabolás) az IP csomagok kisebb részekre való felbontás folyamata, erre azért van szükség, hogy azokon a hálózatokon is szállítható legyen a csomag, amik nem képesek nagyobb csomagok szállítására. Az IPv6-os csomagokat csak a végpontok darabolhatják fel és állíthatják össze, a közbenső routerek számára ez nem engedélyezett. Ezt egy támadó felhasználhatja arra, hogy támadást indítson vagy elrejtse a támadását. Például felhasználhatja azt, hogy egy végpont milyen metodikával állítja össze a csomag darabjait és úgy darabol fel egy csomagot, hogy az egyes darabok átfedésben legyenek. Egy másik típusú fragmentálási támadás keretein belül a támadó elküldi egy csomagnak a darabjait, kivéve a legutolsót, így a címzett hoszt várni fogja a soha meg nem érkező csomagot és ennek következtében felesleges erőforrásokat használ fel. [18]

4.2.6. Neighbor Discovery sérülékenység

A Neighbor Discovery Protocol eleinte nem tartalmazott semmiféle megelőző biztonsági mechanizmust. Emiatt, ha egy támadó hozzáférést szerzett a célpont hálózatához, akkor a protokoll üzeneteit felhasználva több típusú támadást is megvalósíthatott. Például a Router Advertisement csomagok kreálásával egy támadó kiadhatja magát egy IPv6-os routernek a

hálózaton. Ezeket a csomagokat arra is felhasználhatja, hogy a hálózatot ilyen típusú üzenetekkel árasztja el és mindegyikben más prefixet hirdet, ami ahhoz vezet, hogy a kliensek mind új címeket generálnak a különböző hirdetett prefixekhez, ezzel leterhelve a hosztokat. Ezek a támadások DHCPv6 használatával sem szüntethetőek meg, mert a routernek álcázott támadó hoszt kényszeríteni tudja a hálózat többi tagját, hogy ne használja DHCP-t a hálózati címek beszerzéséhez.

4.2.7. IPv6 bevezetését támogató technikák sérülékenységei

A korábban IPv4-es protokollt alkalmazó hálózatokról az IPv6-os hálózatokra történő váltás egy fokozatos folyamat, ennek következtében, amíg nem történik meg a teljes átállás, addig ennek a két protokollnak együtt kell működnie. Emiatt létrehoztak különféle mechanikákat, amelyek segítségével az IPv4-es és IPv6-os eszközök kommunikálhatnak egymással. A legjelentősebb átállást segítő mechanikák az alagút (tunneling) és dual-stack technológiák. Tekintve, hogy széleskörben még nem használatosak a natív IPv6 implementációk és a legtöbb hálózatban mindkét protokoll jelen van, így megkerülhetetlen ezeknek a technológiáknak a használata. Ennek köszönhetően ezek a mechanikák eddig új és ismeretlen fenyegetéseket jelenthetnek, a korábbi IPv4-es sérülékenységeket is felhasználva. [18]

4.3. Védelmi megoldások

Az IPv4-es hálózatok esetén a végpontól végpontig tartó átláthatóság nem valósul meg a hálózati címfordítás szükségessége miatt. Az IPv6 esetén ez nem jelent problémát, mivel a megnövekedett címtér lehetőséget ad arra, hogy minden egyes hosztot egyedi IPv6 címmel lássunk el. Bevett szokássá vált, hogy az emberek hálózati címfordítás és privát IP címek használatával próbálják elrejteni a hálózatuk topológiáját a külvilág felől. Viszont az IPv6 alkalmazásával a cél az, hogy helyreállítsák ezt a láthatóságot, emiatt is szükséges egy új védelmi terv megalkotása és a megfelelő védelmi mechanizmusok implementálása.

IPv4 hálózatok esetében gyakori megoldásnak számított, az, hogy a hálózat határán elhelyeztek egy tűzfalat, amely képes a kimenő és bejövő forgalom szűrésére, valamint gyakran hálózati címfordítást is alkalmaztak. Egy IPv6-os hálózat védelmének biztosítása során célszerű figyelembe venni, hogy úgy alakítsunk ki egy hathatós védelmi rendszert, hogy közben ne korlátozzuk a végpontok elérhetőségét. [15]

A védelem kialakítása során problémát jelenthet, hogy az IPv6-os interfészek több logikai címmel is rendelkezhetnek, amelyek gyakran változhatnak. Tehát nem célszerű olyan biztonsági szabályokat, ACL-eket létrehozni, amelyek cím alapján szűrnek.

4.3.1. IPv6 tűzfal szabályok

Abban az esetben, ha egy dual-stack hálózat védelmét szeretnénk kialakítani, akkor figyelembe kell vegyük az IPv4 és az IPv6 protokollt érintő fenyegetéseket. Ennek következtében, úgy kell kialakítanunk a tűzfal szabályokat, hogy az egyik protokoll használóinak se korlátozza az alapvető funkcionalitását. Például IPv4-es hálózatokban gyakran teljesen letiltják az ICMP forgalmat, viszont az ICMPv6 tiltása esetén előfordulhat, hogy alapvető funkciókat tesznek tönkre. Érdemes úgy kialakítani a védelmet, hogy a hálózat határán lévő tűzfal a kintről befelé

érkező forgalmat szűrje a hálózaton belül állandóan használatos privát IP címek alapján. Emellett célszerű a hosztokon is szoftveres tűzfalakat alkalmazni a nagyobb védelem érdekében, valamint a kritikus eszközökön olyan statikus címeket beállítani, amelyek nem találhatóak ki könnyedén. A Layer 3-as (hálózati réteg) tűzfalak soha nem továbbíthatnak link-local címekkel ellátott csomagokat, emellett képesnek kell lenniük arra, hogy forrás és cél cím, kiegészítő fejlécek és felsőbb rétegbeli protokoll információk alapján szűrjék az adatforgalmat.

4.3.2. Secure Neighbor Discovery

Az eredeti Neighbor Discovery specifikációban az IPsec használatát javasolták, amellyel a protokoll támadási felületét minimalizálni lehet, de több okból sem volt praktikus a használata. Az IPsec hátránya, hogy manuális konfigurációt igényel, amelyben számos biztonsági paramétert kell meghatározni.

A Secure Neighbor Discovery (SEND) amelyet eredetileg az RFC 3971-ben definiáltak, egy védelmi mechanizmus, amelyet azzal a céllal alkottak meg, hogy a Neighbor Discovery protokoll védelmét támogassa. A SEND-et olyan hálózatokban célszerű használni, ahol a fizikai védelem nem biztosított, mint például vezetéknélküli hálózatokban, ahol nagyobb eséllyel hajtható végre a protokollt kiaknázó támadás.

Az RFC dokumentumban több biztonsági szempontot is rögzítettek. Először is szükséges a routerek megbízhatóságát igazolni, mielőtt azok felvehetnék az alapértelmezett routerek szerepét. Egy köteléket szükséges kiépíteni egy hoszt (trust anchor) és a router között, ezt nevezik Certification Path-nak (Tanúsítvány útvonal). További két üzenetfajtát határoztak meg, ezek a Certification Path Solicitation és Certification Path Advertisement üzenetek. Ezeknek az üzeneteknek a segítségével alakítható ki az útvonal a router és a megbízható hoszt között. A hálózatban szereplő hosztok konfigurációjában több trust anchor bejegyzés is szerepel. A hosztok, a routerek és csomópontok hitelességét a trust anchor hosztokon keresztül erősíthetik meg.

Jelenleg a protokoll nem támogatja a statikusan beállított, valamint a SLAAC-kel generált IP címek használatát. Kriptografikusan generált IP címeket (Cryptographically Generated Address, CGA) használnak a hálózatokban, amelyekkel megállapítható, hogy az ND üzenet küldője valóban az, akinek vallja magát az üzenetben. Mielőtt egy cím aktiválható lenne, minden hoszt egy privát és publikus kulcspárt generál, amelyek alapján létrehozható a CGA cím. Ezek a címek, olyan IPv6 címek, amelyeknek az interfész azonosító része kriptografikusan, egyirányú hash művelettel lettek generálva a publikus kulcs és a kiegészítő paraméterek alapján. Annak érdekében, hogy a publikus kulcs és annak paraméterei továbbíthatóak legyenek, definiáltak egy új opcionális ND mezőt, amelyet CGA mezőnek hívnak.

Ezek mellett további három opcionális mezőt definiáltak. Az RSA Signature (RSA aláírás) mezőt, amellyel biztosítható a Neighbor és Router Discovery üzenetek biztonsága. Valamint a Timestamp (Időbélyeg) és Nonce mezők kerültek hozzáadásra, amelyek segítségével megelőzhetőek a visszajátszásos támadások.

Mivel a legtöbb gyártó nem implementálta a SEND protokollt, így az nem terjedt el széles körben. Dual-stack környezetekben, ahol párban használják az IP protokollokat, a SEND nem alkalmazott, mivel az IPv4 sem biztosított, így csak egyedül IPv6-ot használó hálózatokban érdemes alkalmazni. [15]

4.3.3. RA Guard

Az RA Guard-ot (Router Advertisement Guard) az RFC 6105 definiálja. Az RA Guard célja, hogy az adatkapcsolati rétegben szűrje a Router Advertisement üzeneteket megadott feltételek alapján, mielőtt azok célba érhetnének. Meg lehet határozni, hogy csak bizonyos előre definiált forrásokból engedélyezzen ilyen típusú üzeneteket vagy éppen letiltható bizonyos interfészeken az RA üzenetek fogadása.

Az RA Guard hatékonysága nagyrészt attól függ, hogy az adatkapcsolati rétegben működő eszközök milyen mértékben tudják felismerni a Router Advertisement üzeneteket. Gyakorlatban előfordulhat, hogy ezek az eszközök képtelenek felismerni ezeket a csomagokat, főleg, ha Fragmentation Header-t (Töredék fejléc) alkalmaznak egy csomagban. Amikor csomagokat több részbe darabolnak, akkor a második rétegbeli eszközök nem minden esetben tudják megfelelően szűrni azokat. Ennek következtében az RFC 6980-ban előírják, hogy az említett fejlécek használata a Neighbor Discovery Protocol üzeneteivel együtt tilos. [15]

4.3.4. IPsec

Elméletileg az IPsec bármilyen IP csomag biztosítására használható, de valójában az implementációja nem egyszerű feladat. Az IPsec felhasználható az NDP védelmére, azon belül is például a DAD és a SLAAC procedúra védelmére. A Neighbor Discovery protokoll üzenetei (3.3 fejezetben részletezve) többféle címzést is alkalmaznak. A legtöbb típus unicast forráscímet alkalmaz, azonban a Router- és Neighbor Solicitation üzenetek meghatározatlan, végig nullás címet is alkalmaznak (::). Az unicast címek mellett a multicast címek is használatosak, például az összes hoszt megcímzésére (FF02::1) és az összes aktív router megcímzésére (FF02::2) is ilyen típusú címeket használnak.

Az eredeti NDP és SLAAC specifikációja az IPsec Authentication Header (Hitelesítési fejléc) által alkalmazott SA-k (Security Associations, Biztonsági paraméterek) alkalmazását javasolja. Ennek ellenére a leírás nem tartalmazza, hogy pontosan melyik SA-kat érdemes alkalmazni, valamint nincsen definiálva, hogy a paramétereket érintően hogyan jöjjön létre a megállapodás, valamint a használandó kulcscsere protokoll (IKEv1 vagy IKEv2) sincs megadva.

A leírtak alapján az implementáció több problémába is ütközhet. Elsősorban a kulcscsere kérdése a legégetőbb. Lehetetlen az IKE (Internet Key Exchange) protokollokat felhasználni anélkül, hogy ördögi körbe kerülnénk. Az IKE protokoll UDP protokollt alkalmaz, amelynek szüksége van arra, hogy a Neighbor Discovery protokoll megfelelően működjön, amely pont azokat az ICMPv6 üzenetek használja, amelyeket az IPsec-nek védenie kellene. Emellett az IKE nem működőképes multicast üzentekkel. További probléma, hogy sok biztonsági paramétert kell előzetesen egyeztetni, amely jelentős többletterhelést róhat a hálózatra. Az

említettek miatt az IPsec használatát már nem javasolják, hanem csak feltüntetik, mint lehetséges opció. [16]

A fentebb említettek ellenére az IPsec alkalmazható a SLAAC procedúra és ND protokoll védelmére, ha a biztonsági paramétereket előre le lehet egyeztetni. Az egyik megoldás, hogy átmeneti IPv6 címeket alkalmazunk, amelyek esetében már meg vannak határozva a biztonsági paraméterek és ezeknek segítségével lehetne konfigurálni hosszabb élettartamú címeket, majd azokhoz egyeztetni a szükséges paramétereket.

Gyakorlatban ez azt jelenti, hogy több limitáció árán, de lehetséges az IPsec implementálása. Például manuálisan kell kezelnünk a kulcsokat az IKE helyett, valamint szükséges minden hálózati kártya MAC címét előre ismernünk. További megkötés, hogy az üzenetek visszajátszását detektáló mechanizmust deaktiválni szükséges és hogy bizonyos IPsec paramétereket egyeztetés nélkül kell megadni. [23]

5. IPV6 TÁMADÁSI FORMÁK

Ebben a fejezetben bemutatom azokat a támadási formákat, amelyek az IPv6 fejlesztése során bevezetett új mechanikák sérülékenységeit használják fel és nem voltak jelen a protokoll korábbi verziójában.

5.1. Neighbor Discovery Protokollt alkalmazó támadások

A 3.3-as fejezetben tárgyalt Neighbor Discovery Protokoll által használt ICMPv6-os üzeneteket (RS, RA, NS, NA) a támadók fel tudják használni különféle támadások indítására. Ezen csomagok rosszindulatú felhasználásával megvalósíthatóak például Denial of Service, MITM (Man in the middle, Közbeékelődéses támadás) típusú támadások is. Egy Man in the middle típusú támadás során egy támadó két fél közé helyezi magát és a két hoszt közötti forgalmat úgy befolyásolja, hogy mindkét fél azt gondolja, hogy a valós címzettel kommunikál, de valójában a támadóval és ezt felhasználva különböző támadásokat indíthat.

5.1.1. Neighbor Solicitation/Advertisement Spoofing

Ez a támadás típus az NDP protokoll Neighbor Solicitation és Neighbor Advertisement üzeneteit használja fel. Segítségével MITM támadások valósíthatóak meg

A Neighbor Solicitation és Neighbor Advertisement üzenetek használatával a hosztok lekérhetik egymás IPv6 címeihez tartozó MAC címeit. Mivel ez a folyamat nagyjából megegyezik az IPv4-es ARP által alkalmazott folyamattal, így a sérülékenységek is hasonlóak. A támadó a parasite6 eszközzel készít egy olyan hamisított Neighbor Advertisement csomagot, amiben a saját MAC címét szerepelteti és engedélyezi az override (felülírás) zászlót. Ennek következtében, ha egy host Neighbor Solicitation üzenetet küld azzal a céllal, hogy egyik szomszédjának megszerezze a MAC címét, akkor mind a támadó és az szomszéd gép válaszként visszaküld egy Neighbor Advertisement üzenetet, amiben jelzik, hogy a megjelölt IP címhez a saját MAC címük tartozik. Tekintve, hogy a támadó az üzenetében engedélyezte a felülírást, így a kérés indítója az adott IP címhez a támadó MAC címét fogja hozzárendelni a szomszédsági táblájában. Miután ez megtörtént a támadó lehallgathatja a két hoszt között zajló kommunikációt. [19]

5.1.2. Duplicate Address Detection támadás

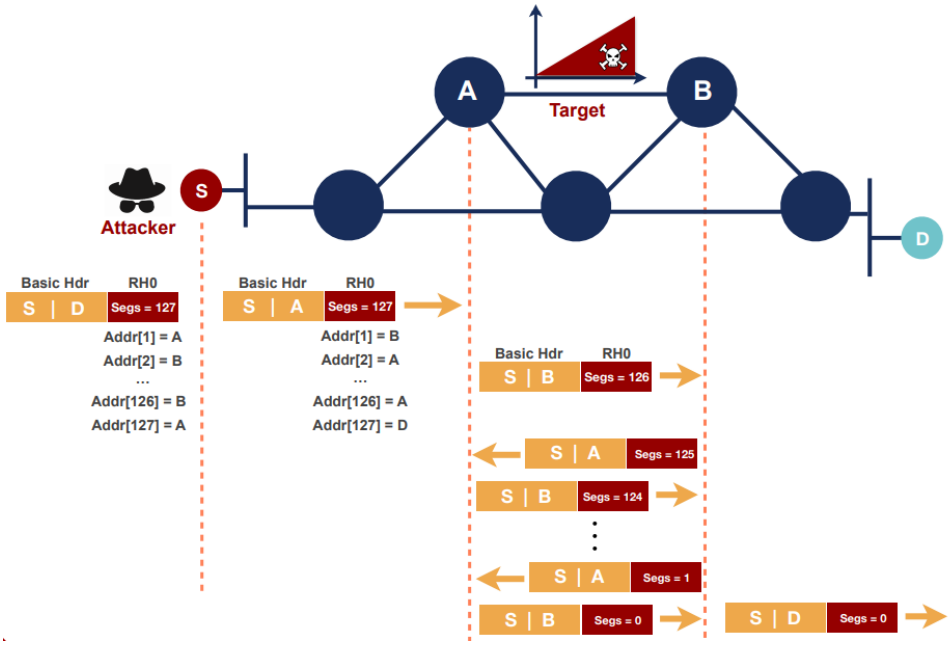
A DAD procedúra az előző támadáshoz hasonlóan NA és NS üzeneteket használ fel. Az áldozat hoszt még mielőtt az automatikusan generált IP cím konfigurációját jóváhagyná küld egy DAD (Duplicate Address Detection) üzenetet, amelyben megbizonyosodik arról, hogy más eszköz nem rendelkezik ezzel a címmel a hálózaton. Ugyanezen a hálózaton egy rosszindulatú támadó egy Neighbor Advertisement üzenettel válaszol, amiben jelzi, hogy ez a cím már használatban van. Ennek hatására a hoszt újabb és újabb címeket generál, amelyre a támadó mindig azt a választ küldi vissza, hogy már foglalt a cím. Egy pár próbálkozás után a hoszt felhagy a cím generálásával, így a hoszt nem lesz képes az interfész azonosító cím automata konfigurációjára, ezáltal nem fog tudni IPv6-os kommunikációban részt venni. Ezáltal a támadó egy szolgáltatás megtagadással járó támadást valósíthat meg. [19]

5.2. Kiegészítő fejlécekkel kapcsolatos támadások

A kiegészítő fejlécek segítségével opcionális paraméterek adhatóak meg egy IPv6 csomagban. Az IPv6-os csomagon belül a fő fejléc után és a szállítási rétegbeli protokollok fejlécei előtt helyezkednek el. Két kiegészítő fejléc típus lehet érdekes biztonsági szempontból: a Routing Header type 0 (nullás verziójú forgalomirányító kiegészítő fejléc) és a Fragment Header (csomagok feldarabolásához használt kiegészítő fejléc). [20]

5.2.1. Routing Header Type 0 támadás

A Routing header type 0 egy elavultnak számító (RFC5095) verziója a forgalomirányító kiegészítő fejlécnek, de egyes korábbi IPv6-os implementációk részét képzik. Ez a fejléc tartalmaz egy olyan mezőt, amelyben szerepeltetve vannak azoknak a távoli csomópontoknak a címei egy listában, amelyen keresztül át kell haladjon a forgalom mielőtt a címzetthez érkezne. Ha a támadó ebben a listában két címet felváltva szerepeltet, akkor az elküldött csomag két távoli csomópont között oda-vissza utazik. Emiatt a két távoli csomópont túlterhelődhet és emiatt a küldött csomag nem érkezik meg a címzetthez, ezzel megvalósítva egy DoS típusú támadást. [20][21]



5.1. ábra: Routing Header Type 0 támadás

5.2.2. Fragment headert alkalmazó támadások

Ezt a kiegészítő fejlécet abban az esetben alkalmazzák a küldő hosztok, ha olyan csomagot szeretnének küldeni, amelyek meghaladják a Path MTU (Maximum Transmission Unit) értéket. A Path MTU meghatározza, hogy mekkora csomagokat küldhet a feladó. Mivel az útvonalon elhelyezkedő routerek különféle MTU értékekkel rendelkezhetnek, így a szűk keresztmetszet a legkisebb MTU értékkel rendelkező router lesz.

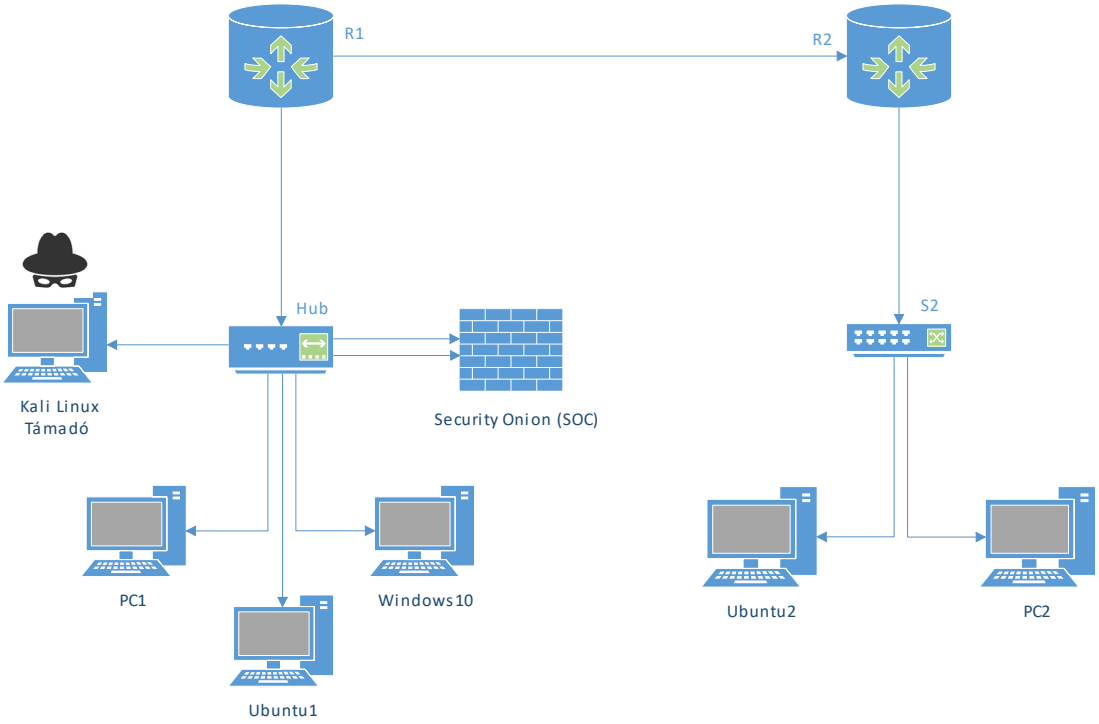
- **Hiányzó darab támadás**: A fejlécet felhasználva a támadó feldarabol egy csomagot majd az összes töredéket elküldi a célpontnak kivéve az utolsót (M zászló jelöli az utolsó darabot). Emiatt a fogadó várakozni fog, egészen addig amíg meg nem kapja a csomag utolsó részletét, ezzel felesleges erőforrásokat felemésztve. [21]
- **Fragment overlapping támadás:** Egy másik támadási módszer során a csomag darabjait úgy küldi el a támadó, hogy azok átfedésben legyenek egymással, ezt úgy éri el, hogy szándékosan helytelen fragment offset értékeket határoz meg a csomagokban. Eredetileg az IPv6 specifikáció nem tiltotta meg az átfedő darabokkal rendelkező csomag összeállítását, annak ellenére, hogy így egyes tűzfalak megkerülhetőek voltak. A hibás csomagok összeállítása különféle hibákat okozhatnak a fogadó eszközön. Erre legelterjedtebb módszer a TCP SYN zászló felülírása. [20]
- **Atomic fragment támadás:** Ez a támadás olyan IPv6-os csomagokat használ fel, amelyek ugyan tartalmazzák a Fragment fejrészt, de ennek ellenére nem feldarabolt csomagok darabjait képzik. Ezt a támadó felhasználhatja úgy, hogy hamisít egy ICMPv6 PTB (Packet Too Big) üzenetet, amelyben jelzi, hogy a használt MTU kisebb mint 1280 bájt. Az RFC 2460 előírja, hogy az IPv6 minimum 1280 bájtos MTU értékekkel működhet és abban az esetben, ha ennél kisebb értékkel rendelkező üzenetet kap egy router, akkor nem kell az üzenet méretét csökkentenie, hanem helyette ezt a csomagot el kell látnia egy Fragment fejléccel. Ezt kihasználva többféle támadás is megvalósítható. Először is sok IPv6 implementációban az atomic fragmenteket tényleges feldarabolt forgalomnak érzékelik az eszközök, ezért megkísérlik a csomagok összeállítását, ami mindig hibával zárul, emiatt felesleges erőforrásokat fog felhasználni a célpont. Továbbá léteznek olyan eszközök, amelyek eldobják az IPv6-os csomagokat, ha azok fragmentáltak, ilyen eszköz például egy BGP forgalomirányító protokollt alkalmazó router. Tehát ha egy támadó küld egy hamisított PTB üzenetet egy BGP routernek, akkor az automatikusan fragmentációs fejléccel fogja ellátni az üzeneteket és emiatt az útvonalon következő BGP router el fogja dobni ezeket a csomagokat. Ezzel a módszerrel Denial of Service típusú támadások valósíthatóak meg. [22]

6. RENDSZERTERV

Ebben a fejezetben megfogalmazásra kerül a dolgozatban megvalósítandó virtuális hálózat és annak topológiájának a terve. Ezen kívül szó lesz a végrehajtani kívánt támadások típusáról és az alkalmazott védelmi módszerek lehetőségeiről.

6.1. Virtuális hálózat

A megvalósítandó virtuális hálózatot a GNS3 (Graphical Network Simulator 3) nevű programmal fogom szimulálni. A GNS3 lehetőséget ad a generikus hálózati eszközök mellett a Cisco által készített eszközök élethű szimulációjára is. Segítségével egy virtuális hálózatban tudom kezelni mind a virtualizált hosztokat, emulált Cisco switcheket és routereket, valamint az SOC rendszert, a Security Oniont.



6.1. ábra: Szimulált hálózat tervezett felépítése

6.1.1. Hálózati topológia elemei

- Switch: Cisco IOSvL2 (VIRL)
- Routerek: Cisco IOSv (VIRL)
- Hub
- SOC: Security Onion
- Hosztok: Kali Linux, Windows 10, Ubuntu, virtuális PC

A hálózati topológia összesen hét darab számítógépet tartalmaz, ezek közül a Kali Linux nevű PC tölti majd be a támadó szerepét. Ezen a számítógépen egy Kali Linux operációs rendszer van installálva, valamint a THC IPv6 eszközkészlet. A többi számítógépen pedig Microsoft Windows 10 és Ubuntu operációs rendszerek futnak, valamint a PC1 és PC2 nevű számítógépeken egy virtualizált rendszer fut. A hosztok megfelelő statikus IPv4 és IPv6 címekkel vannak ellátva.

Ezek mellett a hálózati topológia egyik legfontosabb eleme a SOC (Security Operations Center, Biztonsági Művelet Központ) is helyet kap. A SOC szerepét a Security Onion fogja betölteni a szimulált hálózatban. Segítségével elemezhetjük az indított támadások részleteit, valamint megfelelő konfigurációval a behatolás érzékelő rendszerek – amelyek a Security Onion részei – detektálni tudják az indított támadásokat.

6.2. Megvalósított támadások

A támadó számítógép (Kali Linux) a THC (The Hacker's Choice) IPv6-hoz készült hacker eszközkészletét fogja alkalmazni a támadások során. Az eszközkészlet segítségével különféle támadások valósíthatóak meg például: MITM támadások a Neighbor Solicitation/Advertisement üzenetek használatával (parasite6), DoS támadások a DAD üzenetek felhasználásával (dos-new-ip6), Router megszemélyesítési támadások (fake-router6) és hálózati felderítésre is lehetőséget ad a multicast csoportok kihasználásával. A felsoroltakon belül a dos-new-ip6 és fake-router6 scriptek által megvalósított támadásokat fogom elvégezni.

6.2.1. Denial Of Service támadás Duplicate Address Detection üzenetekkel

A topológiában található R1 router az IPv6-os forgalomirányításért felelős. Ez a router küldi ki Router Advertisement üzeneteket a helyi hálózaton lévő gépeknek. A támadó számítógép a hubon keresztül a routerhez csatlakoztatott interfészén megkezdi a DoS támadást a dos-new-ip6 nevű eszközzel. A támadott számítógép bekapcsolásra kerül, majd megkezdődik az automata IP konfiguráció (SLAAC) folyamata.

Ennek során a hoszt Router Solicitation üzeneteket küld ki annak érdekében, hogy megtalálja a hálózatához kapcsolódó IPv6-os routert, amely elküldi számára az automata konfigurációhoz szükséges adatokat a Router Advertisement csomagok segítségével. Miután a hoszt lekérte a konfigurációhoz szükséges adatokat az IPv6-os routertől és mielőtt még jóváhagyná a kapott prefix és a generált interfész azonosító által meghatározott címet, a hoszt Neighbor Solicitation üzeneteket küld ki hálózaton, amivel lekéri, hogy a hálózaton lévő hosztok valamelyike már rendelkezik a sajátjával azonos IP címmel. A támadó szintén megkapja ezeket az üzeneteket és egy Neighbor Advertisement üzenettel válaszol, amelyben jelzi, hogy ez a cím már foglalt. Ennek következtében a hoszt új címet generál, majd a hálózaton újra Neighbor Solicitation üzeneteket küld ki, amire a támadó ismét válaszol, hogy foglalt a cím, így a hoszt nem fog tudni érvényes IP címet generálni saját maga számára és ennek hatására a hoszt nem lesz képes más IPv6-os eszközökkel kommunikálni.

6.2.2. Denial Of Service támadás Router megszemélyesítésével

A topológiában két alhálózat szerepel, amelyeknek R1 és R2 routerek az alapértelmezett átjárójuk. Mindkét router Router Advertisement üzeneteket küld ki az alhálózatukban, amelyben jelzik a jelenlétüket. A támadó jelen esetben a fake-router6 nevű eszközt fogja alkalmazni. A támadás során a támadó olyan Router Advertisement üzeneteket küld ki, amelyben a Router preference érték (preferált router) nagyobb, mint az R1 által küldött üzenetekben. Az R1 router alapértelemezetten közepes preferencia értéket állít be a csomagokban, a támadó pedig magas prioritást. Ennek következtében a támadó fogja betölteni az alapértelmezett router szerepét a hálózaton, mivel az általa küldött üzenetek nagyobb preferencia értékkel bírnak. Emellett a támadó által küldött Router Advertisement csomagok nagyobb Router Lifetime értékkel rendelkeznek. Ez a mező definiálja, hogy az adott router hány másodpercig számít érvényes alapértelmezett routernek.

Miután a támadó számítógépe átvette az alapértelmezett router szerepét, utána az R1 alhálózatában lévő Windows10 hoszt és R2 alhálózatában lévő Ubuntu2 képtelen lesz egymással kommunikálni.

6.3. Támadás detektálása a Security Onion eszközeivel

A Security Onion egy Threat Hunting céllal létrejött, ingyenes és nyílt Linux disztribúció, amely képes rendszerek és hálózatok monitorozására és az ezzel kapcsolatos naplófájlok kezelésére. Három fő funkciójaként képes az áthaladó adatcsomagok elkapására, a hoszt- és hálózat alapú behatolás-érzékelő rendszerek kezelésére és a különféle fenyegetések által kiváltott riasztások elemzésére. Olyan biztonsági eszközöket foglal magába, mint például: Kibana, TheHive, Zeek, Wazuh, Suricata stb.

6.3.1. Behatolás érzékelő rendszerek szerepe

A Security Onion SOC részét képzi a Zeek és a Suricata nevű behatolás érzékelő rendszerek. Mindkét rendszer rendelkezik IPv6-os támogatással, tehát képesek az IPv6-os támadásokat egy részét felfedezni.

A Zeek egy hálózat alapú behatolás-érzékelő rendszer, de szabályalapú detektálás helyett viselkedésalapú érzékelést alkalmaz. Szkriptek segítségével határozza meg, hogy milyen adatokat naplózzon és hogy mikor váltson ki riasztásokat. Továbbá képes csatolmányokat vírus elemzés céljából továbbítani, hozzáférést letiltani rosszindulatú helyekhez és akár le is tud kapcsolni számítógépeket, amelyek megsértik a biztonsági előírásokat.

A Suricata egy aláírásos módszert alkalmazó hálózat alapú behatolás-érzékelő rendszer. Egyik előnye a Zeek-el szemben, hogy natívan támogatja a több feldolgozószálon történő feldolgozást. Így az egyes adatcsomag folyamokat külön processzormagokhoz tudja rendelni és ezáltal párhuzamosan tudja elemezni azokat. Továbbá a Snort szabályok alkalmazását is támogatja.

Egy 2019-es tanulmány szerint, a Zeek és Suricata hálózati behatolás érzékelő rendszerek 62 ismert IPv6-os támadások közül viszonylag csak keveset tudtak érzékelni. A Zeek esetében 22

detektált támadásról számolnak be, ami alapján 35%-os hatékonysággal képes az IPv6-os támadásokat felismerni. A Suricata használata esetén a támadások közül 29 támadást tudott detektálni, ami 46%-os hatékonyságot jelent. [23]

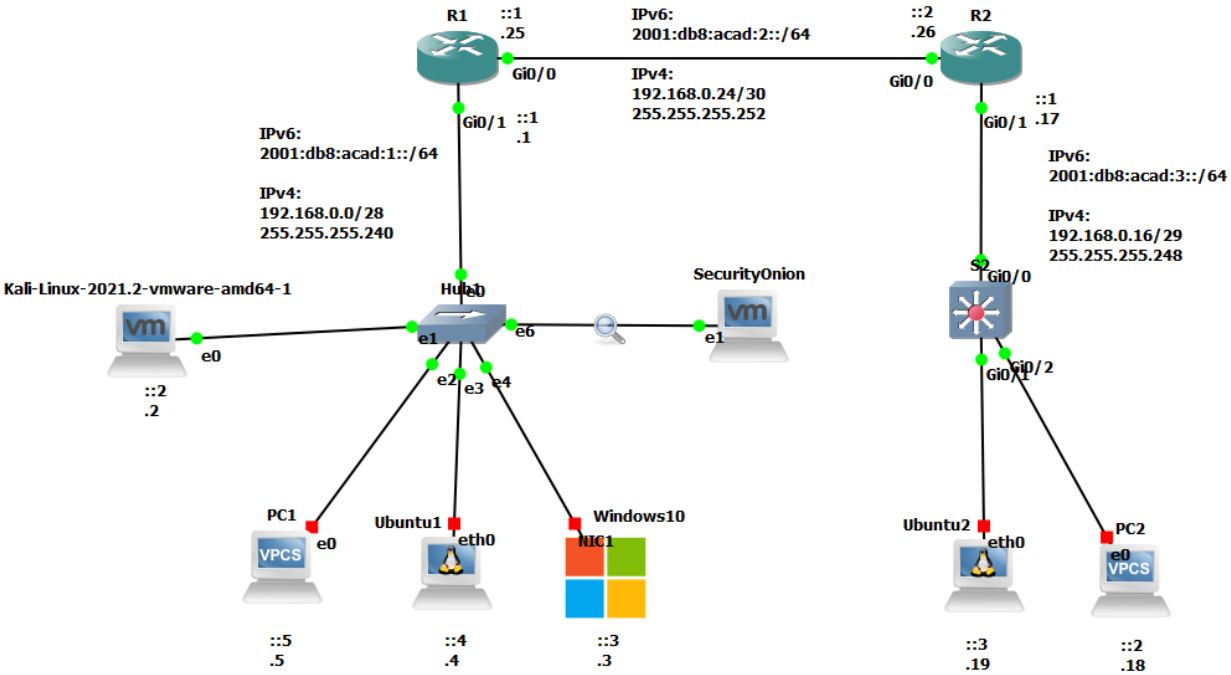
Megkísérlem a hálózati behatolás érzékelő rendszereknek az IPv6-os támadásokkal szembeni hatékonyságát növelni új szabályok és különféle beállítások meghatározásával, annak érdekében, hogy a rendszer képes legyen automatikusan detektálni a kivitelezett támadásokat.

7. MEGVALÓSÍTÁS

Ez a fejezet taglalja a kialakított virtuális hálózat tagjait, azoknak pontos típusát és ismérveit, valamint az érintett eszközök konfigurációját. Megtörténik az alkalmazott rendszerek alapvető tesztelése, hogy működéséről bizonyosságot nyerjek. Ezek mellett szó lesz a ténylegesen megvalósított támadásokról és ezeknek a detektálásának a körülményeiről.

7.1. Virtualizált környezet kezdeti konfigurációja

A hálózat a GNS3 ingyenes szoftver 2.2.29-es verziójának segítségével lett megvalósítva a virtuális környezetben. A hálózat eszközeinek nagy része VMware Workstation-ben virtualizált virtuális gépeken fut. A hálózatnak eleme több Cisco által kiadott IOSv rendszerére épülő switch és router. A hálózatban részt vesz egy hub is, aminek segítségével továbbítható a Security Onion monitoring (megfigyelő) portjához a hálózat forgalma. Ezek mellett a hálózatban található számos másik hoszt, ezek között szerepelnek az Ubuntu, Windows 10 operációs rendszerrel ellátott gépek, egyéb virtuális hosztok és a támadó számítógép, amin pedig Kali Linux 2021.2 verziójú rendszer van installálva. Az alábbi ábrán látható a fent leírt hálózat topológiája:



7.1. ábra: GNS3-ban megvalósított hálózat topológiája

7.1.1. Cisco routerek kezdeti konfigurációja

Az alábbi táblázatban látható az R1 és R2 nevű routerek kezdeti konfigurációja, amelyben néhány alapvető beállítás mellett szerepel a hálózati interfészek IP címzése, valamint az alapértelmezett útvonalak beállítása és további egyéb beállítások.

R1 router konfigurációja	R2 router konfigurációja
!	!
version 15.6	version 15.6
service timestamps debug datetime msec	service timestamps debug datetime msec
service timestamps log datetime msec	service timestamps log datetime msec
service password-encryption	service password-encryption
!	!
hostname R1	hostname R2
!	!
enable secret 5 1el69$5QPsY2EqontZO/NRUrtlW/	enable secret 5 $1$2ZTc$q.I8hX/rHQ67iaCeGXU5t/
!	!
no ip domain lookup	no ip domain lookup
ip cef	ip cef
ipv6 unicast-routing	ipv6 unicast-routing
ipv6 cef	ipv6 cef
!	!
!	!
interface GigabitEthernet0/0	interface GigabitEthernet0/0
ip address 192.168.0.25 255.255.255.252	ip address 192.168.0.26 255.255.255.252
duplex auto	duplex auto
speed auto	speed auto
media-type rj45	media-type rj45
ipv6 address FE80::1 link-local	ipv6 address FE80::2 link-local
ipv6 address 2001:DB8:ACAD:2::1/64	ipv6 address 2001:DB8:ACAD:2::2/64
!	!
interface GigabitEthernet0/1	interface GigabitEthernet0/1
ip address 192.168.0.1 255.255.255.240	ip address 192.168.0.17 255.255.255.248
duplex auto	duplex auto
speed auto	speed auto
media-type rj45	media-type rj45
ipv6 address FE80::1 link-local	ipv6 address FE80::1 link-local
ipv6 address 2001:DB8:ACAD:1::1/64	ipv6 address 2001:DB8:ACAD:3::1/64
!	!
!	!
no ip http server	no ip http server
no ip http secure-server	no ip http secure-server
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0	ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0
!	!
ipv6 route ::/0 GigabitEthernet0/0	ipv6 route ::/0 GigabitEthernet0/0
!	!
banner motd ^CUnauthorized access is prohibited!^C	banner motd ^CUnauthorized access is prohibited!^C
!	!
line con 0	line con 0
exec-timeout 0 0	exec-timeout 0 0

```
 privilege level 15
 password 7 1511021F0725
 logging synchronous
 login
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 password 7 104D000A0618
 login
 transport input none
line vty 5 15
 password 7 104D000A0618
 login
 transport input none
!
```

```
 privilege level 15
 password 7 070C285F4D06
 logging synchronous
 login
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 password 7 030752180500
 login
 transport input none
line vty 5 15
 password 7 030752180500
 login
 transport input none
!
```

7.1. táblázat: R1 és R2 routerek konfigurációs beállításai

7.1.2. Használt hálózati címzés

A hálózatban IPv4 és IPv6 címzés is használatban van. A valóságnak megfelelően általános megvalósítás, ugyanis ritkán fordul elő az, hogy egy hálózatban csak IPv6 címzés használatos. Az alábbi táblázat első fele írja le a hálózatban lévő routerek IPv4 és IPv6 címeit, a második fele pedig az egyéb hosztok IP címzését.

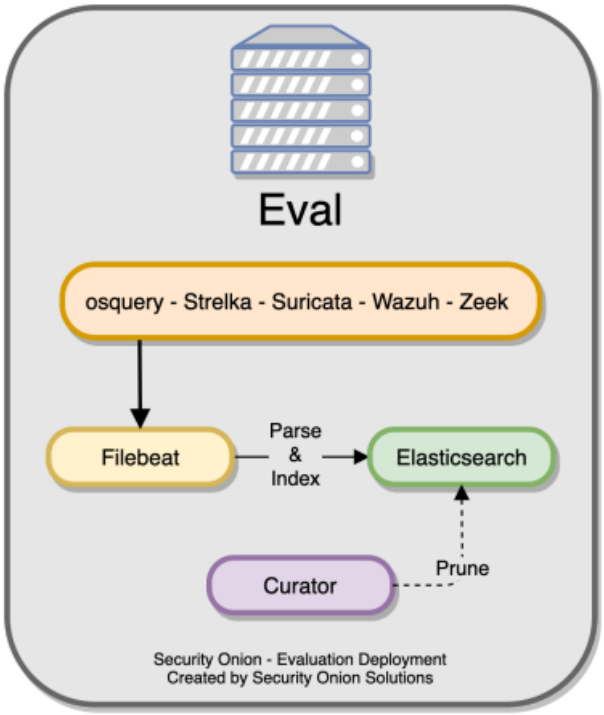
Hoszt	Interface	IPv4 cím	IPv6 cím	MAC
R1	Gi0/1	192.168.0.1/28	2001:db8:acad:1::1/64	0c:aa:e6:e9:5d:01
R1	Gi0/0	192.168.0.25/30	2001:db8:acad:2::1/64	
R2	Gi0/0	192.168.0.26/30	2001:db8:acad:2::2/64	0c:aa:e6:f5:2b:00
R2	Gi0/1	192.168.0.17/29	2001:db8:acad:3::1/64	
Kali Linux 2021.2	eth0	192.168.0.2/28	2001:db8:acad:1::2/64	00:0c:29:14:4e:96
PC1	eth0	192.168.0.5/28	2001:db8:acad:1::5/64	00:50:79:66:68:00
Ubuntu1	eth0	192.168.0.4/28	2001:db8:acad:1::4/64	Változó
Windows10	NIC1	192.168.0.3/28	2001:db8:acad:1::3/64	0c:aa:e6:b9:01:00
Security Onion	eth0	192.168.0.10/28	2001:db8:acad:1::10/64	00:0c:29:4b:4c:c3
Security Onion	eth1 (monitoring mode)	-	-	00:0c:29:4b:4c:cd
Ubuntu2	eth0	192.168.0.19/29	2001:db8:acad:3::19/64	Változó
PC2	eth0	192.168.0.18/29	2001:db8:acad:3::18/64	00:50:79:66:68:01

7.2. táblázat: Hálózatban használt IPv4/IPv6 és MAC címek

7.1.3. Security Onion kezdeti konfigurációja

A Security Onion 2.3.100-as verziója lett installálva egy CentOS7 Linux alapú disztribúcióra egy virtuális gépre. A Security Onion installáció választható architektúrái a következőek:

Import, Evaluation, Standalone és Distributed. A választásom az Evaluation felépítésre esett, mert funkcionalitás szinten az összes lehetséges opció elérhető, ugyanúgy, mint a Standalone esetében, viszont kevésbé erőforrás-igényes. Az összes szolgáltatás, telepítésre került, ezek a következőek: Grafana, Osquery, Wazuh, Playbook, Strelka. Hátránya a Standalone változathoz képest, hogy a logok begyűjtése és feldolgozása kevésbé skálázható, így valós környezetben nem ajánlott a használata.



7.2. ábra: Security Onion evaluation architektúrájának felépítése

A hivatalos dokumentáció alapján az Evaluation változat hardverigénye a következő: 4 CPU mag, 12 GB ram és 200 GB tárhely. Ezek mellett szükséges legalább kettő hálózati interfész használata. A kettő közül az egyiken keresztül a Security Onion webes adminisztrációs felülete érhető el, a második interfész pedig a monitoring port szerepét tölti be.

Ezen információk felhasználásával készítettem el VMware-ben a virtuális gépet, amelyen a Security Onion fog futni. Hozzáadásra került egy NAT típusú interfész, amelyen keresztül a gazdagépen érhető el a Security Onion webes adminisztrációs felülete. Ezen kívül további két interfészt adtam hozzá a virtuális géphez. Ezek az interfészek a GNS3 hálózatához csatlakoznak a megfelelő virtuális VMware hálózaton keresztül. A két interfész közül az egyik a monitoring port, amelyen keresztül lehallgathatóvá válik a környezet hálózati forgalma, hiszen egy hub-hoz csatlakozik, amely minden portján kiküldi a rajta áthaladó csomagokat. A másik interfészen keresztül pedig hagyományos hálózati forgalom bonyolítható le.

7.2. Alapvető működés megerősítése

A szakdolgozat ezen fejezetében a különböző rendszerek alapvető működését erősítem meg tesztek lebonyolításával. A tesztelendő rendszerek közé tartozik a virtualizált hálózat és a

Security Onion szenzorai. A tesztek kiterjednek a hálózat megfelelő működésére, a Security Onion komponensek megfelelő üzemelésére. Ezen belül kiemelt szerepet kap a monitoring hálózati port tesztelése.

7.2.1. Hálózat alapvető funkcionalitásának megerősítése

A hálózat megfelelő működése és az IPv4 és IPv6 címek, valamint a forgalomirányító bejegyzések megfelelő konfigurációja megerősíthető a *ping/ping6* és *traceroute* parancsok kiadásával a megfelelő állomásokon. Az említett parancsok a megfelelő ICMP és ICMPv6 típusú csomagok kiküldésével tesztelik a hálózatot, oly módon, hogy visszaigazolják, hogy ezek a csomagok rendben megérkeztek az egyes célállomásokhoz vagy sem. A hálózat tesztelési procedúrája során a következő parancsokat adtam ki az alábbi formában.

Forrás	**Cél**	**Eredmény**
Security Onion (2001:db8:acad:1::10)	R1 (2001:db8:acad:1::1)	Sikeres ping6 (ping6 2001:db8:acad:1::1)
Kali Linux (2001:db8:acad:1::2)	Ubuntu2 **(**2001:db8:acad:3::3**)**	Sikeres ping6 (ping6 2001:db8:acad:3::3)
Ubuntu1 (192.168.0.4)	PC2 **(**192.168.0.18**)**	Sikeres traceroute 1 192.168.0.1 (192.168.0.1) 2 192.168.0.26 (192.168.0.26) 3 192.168.0.18 (192.168.0.18)

7.3. táblázat: Hálózat tesztelési célzattal kiadott parancsok és eredményeik

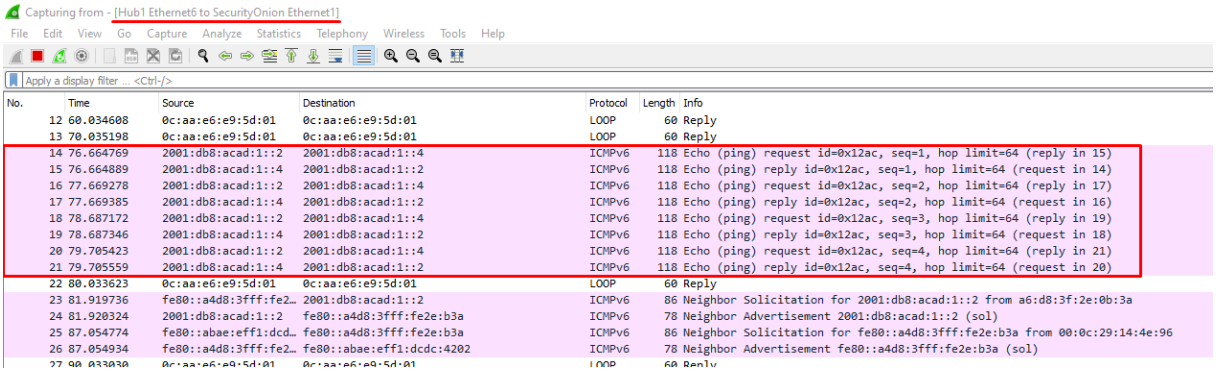
7.2.2. Security Onion alapvető funkcionalitásának megerősítése

A Security Onion virtuális gép bekapcsolása során mindenképpen szükséges a megfelelő NAT interfészt engedélyezni, amelyen keresztül elérhető válik a gazdagép számára a webes adminisztrációs felület. A GNS3 minden esetben felülírja a meglévő interfészek konfigurációját és inaktív módba állítja a NAT típusú interfészeket. Tapasztalatom szerint elengedhetetlen az említett interfész manuális engedélyezése, annak érdekében, hogy a Security Onion megfelelően működjön. A komponensek megfelelő működése a beépített *sudo so-status* parancs kiadásával erősíthető meg, amely kilistázza, hogy az alkalmazott szolgáltatások megfelelően betöltődtek a bekapcsolást követően.

```
[bdaniel@securityonionc ~]$ sudo so-status

Checking Docker status

    Docker ---------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]

Checking container statuses

    so-aptcacherng ------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-curator ----------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-dockerregistry ---------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-elastalert -------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-elasticsearch ----------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-filebeat ---------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-fleet ------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-grafana ----------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-idstools ---------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-influxdb ---------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-kibana ------------------------------------------------------------------------------------------------------------------------------------------------ [ OK ]
    so-kratos ------------------------------------------------------------------------------------------------------------------------------------------------ [ OK ]
    so-mysql ------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-nginx ------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-playbook ---------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-redis ------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-sensoroni --------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-soc --------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-soctopus ---------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-steno ------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-strelka-backend --------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-strelka-coordinator ----------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-strelka-filestream ------------------------------------------------------------------------------------------------------------------------------------ [ OK ]
    so-strelka-frontend -------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-strelka-gatekeeper ------------------------------------------------------------------------------------------------------------------------------------ [ OK ]
    so-strelka-manager --------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-suricata ---------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-telegraf ---------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-wazuh ------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]
    so-zeek -------------------------------------------------------------------------------------------------------------------------------------------------- [ OK ]

[bdaniel@securityonionc ~]$
```

7.3. ábra: sudo so-status parancs konzolos kimenete

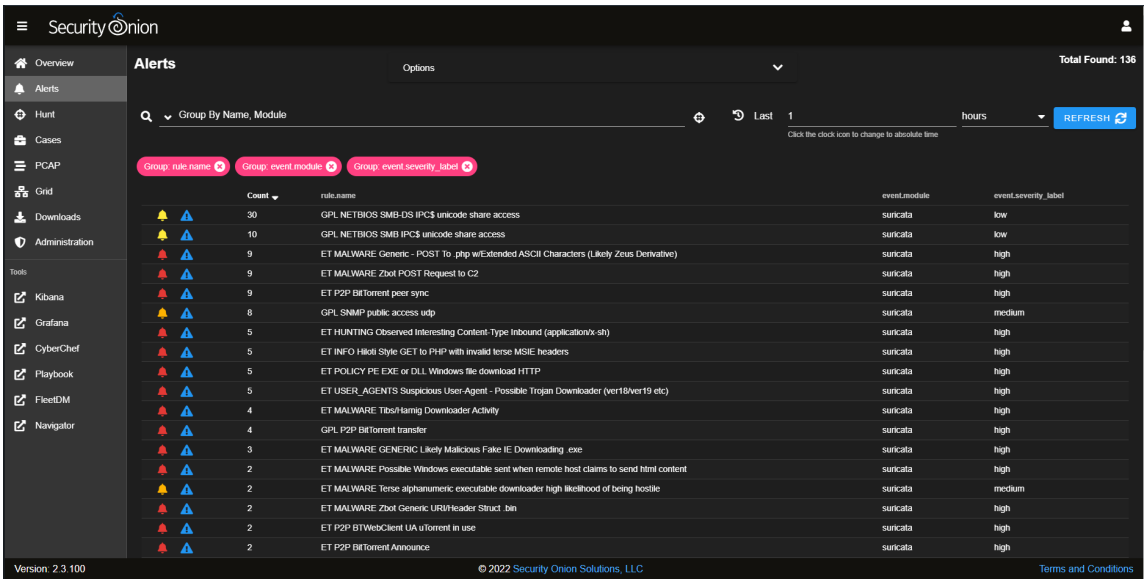
Fontos ellenőrizni, hogy a Security Onion monitoring interfészéhez rendben megérkeznek a Hub-on keresztülhaladó csomagok. Ezt a Wireshark segítségével fogom megtenni, amely képes egy vonalon elkapni a kimenő és bejövő csomagokat (packet capture). A teszt során a Kali Linux-os hoszt fogja megpingelni az Ubuntu1 nevű hosztot. Az alábbi kép készítését megelőzően a következő parancsot adtam ki: *ping6 -I 2001:db8:acad:1::2 2001:db8:acad:1::4*, ez a parancs az említett hosztok IPv6 címeit tartalmazza.



7.4. ábra: A Security Onion hoszt és Hub közötti vonalon aktív packet capture eredménye

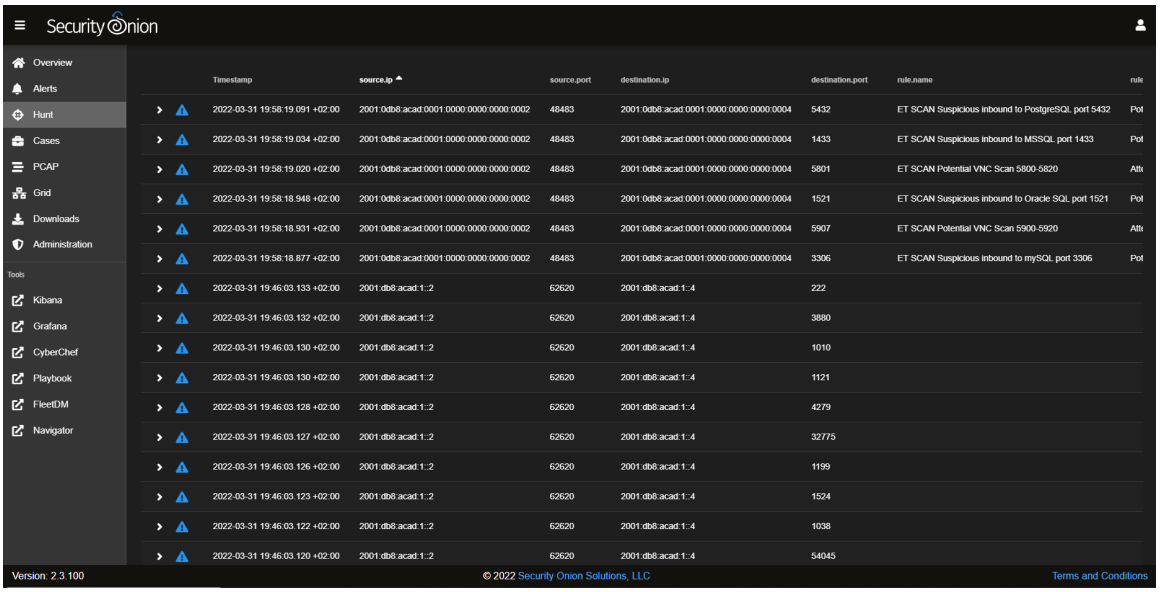
A fentebbi kép alapján látható, hogy a Security Onion monitoring hálózati interfészén megfelelően megjelenik az alhálózaton végbemenő hálózati forgalom, így képes lesz ezen a portján hallgatózni.

Továbbá tesztelendő, hogy a Security Onion megfelelően felismeri a fenyegetéseket és helyesen elkészíti a riasztásokat. Ez a funkcionalitás a beépített *so-test* paranccsal ellenőrizhető, amely korábbról ismert támadások során generált forgalmakat (pcap) játszik vissza a Security Onion monitoring portján és ennek következtében meg kell jelenjenek a támadások által kiváltott riasztások a webes adminisztrációs felületen az „Alerts” fül alatt.



7.5. ábra: Security Onion webes felületén megjelenő riasztások

Ezt követően a Kali Linux hosztról egy portfeltérképezést indítok az Ubuntu1 gép címére a következő parancs kiadásával: *nmap -6 -v 2001:db8:acad:1::4.* A parancs eredményeképpen a támadó a konzolban listázva láthatja az adott címen nyitott hálózati portokat. Ennél viszont lényegesebb, hogy a Security Onion a „Hunt" nevű aloldalán bejegyzést hoz létre, hogy gyanúsnak vélhető esemény történt, valamint a megfelelő riasztásokat is létrehozza. A Security Onion keresőjében kiadva a ** AND source.ip:"2001:0db8:acad:0001:0000:0000:0000:0002" | groupby observer.name* parancsot, láthatjuk a támadó IPv6 címéről küldött gyanús forgalmakat, amely az alábbi képen meg is jelenik, a bejegyzések között megtalálhatóak azok amelyekben jelzi, hogy érzékelte a port feltérképezést. Ezáltal bizonyítást nyert, hogy az érzékelő rendszerek megfelelően működnek az alapértelmezett szabályrendszer felhasználásával.



7.6. ábra: Port feltérképezést követően megjelenő bejegyzések

7.3. Megvalósított támadások

Az alábbi alfejezetekben részletezem a végrehajtott Denial of Service típusú támadásoknak az ismérveit és azoknak a hatásait. Továbbá részletesen elemzem, hogy a támadások milyen gyengepontokat használnak fel és hogy hogyan térnek el az alkalmazott üzenetek a valósaktól.

7.3.1. Denial Of Service támadás Duplicate Address Detection üzenetekkel

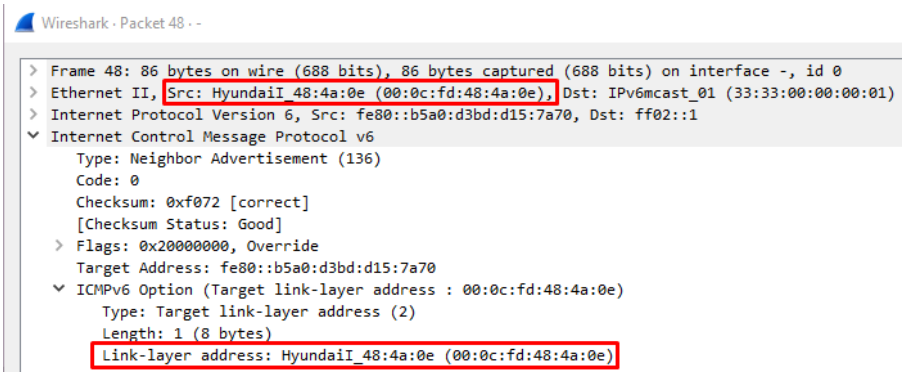
A helyi hálózatban az R1-es Cisco Virl képfájlt futtató router látja el az IPv6-os forgalomirányítás feladatait. Abban az esetben, ha egy újonnan bekapcsolt eszköz kerül a topológiába, akkor az adott eszköz először is lekéri a hálózatban aktív IPv6-ot támogató routerek jegyzékét, jelen esetben az egyik ilyen router szerepét az R1 fogja betölteni. Miután a célpont összegyűjtötte a kellő információkat az automatikus IP cím konfigurációhoz a routertől, utána a hoszt a kapott információk alapján megpróbál IPv6 címeket generálni saját maga számára. Mielőtt véglegesíthetné ezeket a címeket, a hálózatban kiküld egy Neighbor Solicitation üzenetet, amelyben jelzi, hogy az adott címet szeretné használni és várja a választ a hálózat további részvevőitől, hogy már aktív a kiválasztott cím vagy sem. Mivel a támadó is megkapja ezeket az üzenetek, így reagálni is tud egy Neighbor Advertisement üzenettel, amelyben jelzi, hogy a generált címet már alkalmazza és emiatt kénytelen lesz a célpont új címet generálni, egészen addig, míg megszakad az automata IP cím konfigurációs folyamat és nem próbálkozik további címek generálásával és ezáltal képtelen lesz az IPv6 hálózattal kommunikálni.

A támadott gép szerepét a Windows10 nevű gép fogja betölteni. A hálózat részét képző routerek, switchek és hubok vannak bekapcsolva, valamint a támadó, Kali Linux hoszt és a Security Onion nevű hosztok aktívak a támadást megelőzően. A támadó gépén szükséges kiadni a *sudo atk6-dos-new-ip6 eth0* parancsot, aminek következtében minden új Duplicate Address Detection típusú Neighbor Solicitation üzenetre egy Neighbor Advertisement üzenetet fog küldeni válaszul, amiben jelzi, hogy a generált cím már foglalt. Miután bekapcsolásra kerül a Windows10 PC az alábbi ábrán látható hálózati forgalom zajlik le. Az említett ábrán látható forgalom a mellékletként csatolt *DAD_tamadas.pcapng* fájlban található meg.

No.	Time	Source	Destination	Protocol	Length	Info
4	17.202172	fe80::1	ff02::1	ICMPv6	118	Router Advertisement from 0c:aa:e6:e9:5d:01
5	17.928099	2001:db8:acad:1:6140:bacb:d33:36…	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for 2001:db8:acad:1::1 from 0c:aa:e6:b9:01:00
6	17.928252	fe80::b5a0:d3bd:d15:7a70	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for fe80::1 from 0c:aa:e6:b9:01:00
7	17.932263	2001:db8:acad:1::1	2001:db8:acad:1:6140:bacb:d33:3681	ICMPv6	86	Neighbor Advertisement 2001:db8:acad:1::1 (rtr, sol, ovr) is at 0c:aa:e6:e9:5d:01
8	17.933069	fe80::1	fe80::b5a0:d3bd:d15:7a70	ICMPv6	86	Neighbor Advertisement fe80::1 (rtr, sol, ovr) is at 0c:aa:e6:e9:5d:01
9	18.000760	2001:db8:acad:1:6140:bacb:d33:36…	ff02::1:ff00:1	ICMPv6	86	Neighbor Solicitation for 2001:db8:acad:1::1 from 0c:aa:e6:b9:01:00
10	18.001884	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
12	18.002494	2001:db8:acad:1::1	2001:db8:acad:1:6140:bacb:d33:3681	ICMPv6	86	Neighbor Advertisement 2001:db8:acad:1::1 (rtr, sol, ovr) is at 0c:aa:e6:e9:5d:01
13	18.025084	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
15	18.025766	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
19	18.074909	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
21	18.076800	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
31	18.104863	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
32	18.105202	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
36	18.155838	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
37	18.156141	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	90	Multicast Listener Report Message v2
42	18.171993	::	ff02::1:ff15:7a70	ICMPv6	78	Neighbor Solicitation for fe80::b5a0:d3bd:d15:7a70
43	18.172284	::	ff02::1:ff00:3	ICMPv6	78	Neighbor Solicitation for 2001:db8:acad:1::3
44	18.172372	::	ff02::1:ff15:7a70	ICMPv6	78	Neighbor Solicitation for 2001:db8:acad:1:b5a0:d3bd:d15:7a70
45	18.172458	::	ff02::1:ff33:3681	ICMPv6	78	Neighbor Solicitation for 2001:db8:acad:1:6140:bacb:d33:3681
46	18.172539	fe80::b5a0:d3bd:d15:7a70	ff02::2	ICMPv6	62	Router Solicitation
47	18.172965	fe80::b5a0:d3bd:d15:7a70	ff02::16	ICMPv6	190	Multicast Listener Report Message v2
48	18.179298	fe80::b5a0:d3bd:d15:7a70	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::b5a0:d3bd:d15:7a70 (ovr) is at 00:0c:fd:48:4a:0e
49	18.179549	2001:db8:acad:1::3	ff02::1	ICMPv6	86	Neighbor Advertisement 2001:db8:acad:1::3 (ovr) is at 00:0c:2e:62:bc:9c
50	18.179688	fe80::b5a0:d3bd:d15:7a70	ff02::1	ICMPv6	86	Neighbor Advertisement fe80::b5a0:d3bd:d15:7a70 (ovr) is at 00:0c:fd:48:4a:0e
51	18.180076	2001:db8:acad:1:b5a0:d3bd:d15:7a…	ff02::1	ICMPv6	86	Neighbor Advertisement 2001:db8:acad:1:b5a0:d3bd:d15:7a70 (ovr) is at 00:0c:0a:9a:7d:8a
52	18.180170	2001:db8:acad:1::3	ff02::1	ICMPv6	86	Neighbor Advertisement 2001:db8:acad:1::3 (ovr) is at 00:0c:2e:62:bc:9c
53	18.180430	2001:db8:acad:1:6140:bacb:d33:36…	ff02::1	ICMPv6	86	Neighbor Advertisement 2001:db8:acad:1:6140:bacb:d33:3681 (ovr) is at 00:0c:49:3d:5e:8b
54	18.180540	2001:db8:acad:1:b5a0:d3bd:d15:7a…	ff02::1	ICMPv6	86	Neighbor Advertisement 2001:db8:acad:1:b5a0:d3bd:d15:7a70 (ovr) is at 00:0c:0a:9a:7d:8a
55	18.180943	2001:db8:acad:1:6140:bacb:d33:36…	ff02::1	ICMPv6	86	Neighbor Advertisement 2001:db8:acad:1:6140:bacb:d33:3681 (ovr) is at 00:0c:49:3d:5e:8b

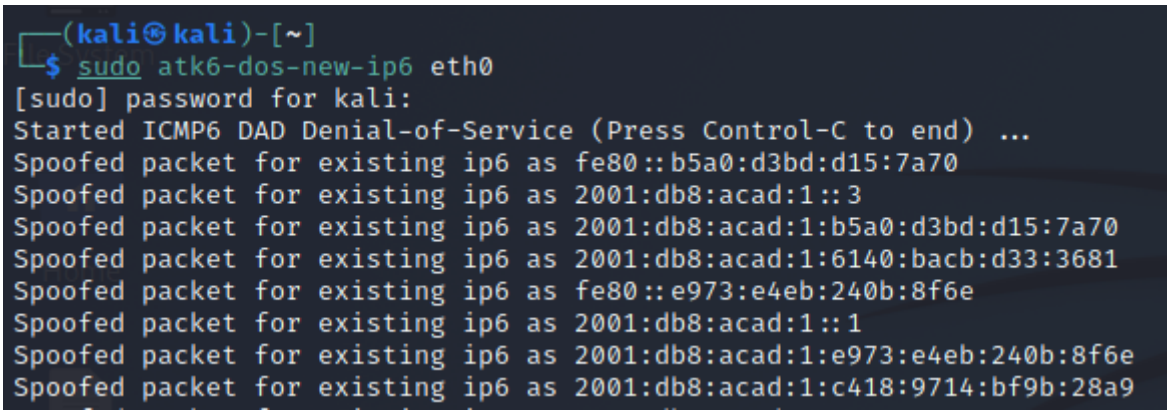
7.7. ábra: DAD támadás során keletkezett keretek

A 4-es számú Router Advertisement üzenetben az R1-es router hirdeti az automata cím konfigurációhoz szükséges információkat, mint például a prefixet ami jelent esetben *2001:db8:acad:1::/64* lesz. A 5-9 számú keretekben a Windows10-es hoszt megcímzi az IPv6-os alapértelmezett átjáró címeit solicited node multicast cím használatával (FF02::01::FF00:1) és az R1 router reagál is Neighbor Advertisement üzenetekkel ezekre a kérésekre, amelyben jelzi, hogy a kért IPv6 címek a saját MAC címén érhetőek el (0c:aa:e6:e9:5d:01). A 42-45 számú üzenetekben a Windows10 hoszt kiküldi a Neighbor Solicitation üzeneteket a hozzátartozó négy IPv6 címhez, amivel ellenőrzi, hogy ezek foglaltak-e már. Ezekben a csomagokban forráscímként meghatározatlan cím van feltüntetve (::), az ilyen típusú Neighbor Solicitation üzenetek kimondottan a DAD procedúra során alkalmazottak. Erre válaszképpen a 48-55 számú Neighbor Advertisement üzenetek érkeznek. Ezeket a csomagokat a támadó küldi válaszul, amelyekben jelzi, hogy az érintett IPv6 címek mind foglaltak.



7.8. ábra: Hamisított MAC címet tartalmazó Neighbor Advertisement csomag

A csomagok közelebbi megvizsgálásával feltűnik, hogy ezek a csomagok mind véletlenszerű, nem létező forrás MAC címről érkeznek, amelyeket nem birtokolnak valós hosztok a hálózatban. Ilyen hamis csomag látszik a fenti ábrán, ez a 48-as számú csomag.



7.9. ábra: Kali Linuxon indított atk6-dos-new-ip6 program kimenete

A Kali Linuxon indított program a támadás során konzolos kimeneten mutatja, hogy mely címek nevében küldött ki eddig hamis Neighbor Advertisement csomagokat. Miután a Windows10 hoszt nem tudta a képen látható címeket (*fe80::b5a0:d3bd:d15:7070,*

2001:db8:acad:1::3, 2001:db8:acad:1:b5a0:d3bd:d15:7a70, 2001:db8:acad:1:6140:bacb:d33:3681) aktiválni, utána nem sokkal később a célpont újra próbálkozik új IPv6 címeket (*2001:db:acad:1:e973:e4eb:240b:8f6e, 2001:db:acad:1:c418:9714:bf9b:28a9*) generálni, de hasonlóan hamis válaszüzeneteket küld a támadó és ennek következtében a célpont képtelen lesz használható címeket generálni.

```
C:\Windows\system32>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : MSEDGEWIN10
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection #2
   Physical Address. . . . . . . . . : 0C-AA-E6-B9-01-00
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:db8:acad:1::3(Duplicate)
   Link-local IPv6 Address . . . . . : fe80::e973:e4eb:240b:8f6e%14(Preferred)
   Autoconfiguration IPv4 Address. . : 169.254.40.240(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 2001:db8:acad:1::1
                                       fe80::1%14
   DHCPv6 IAID . . . . . . . . . . . : 118270694
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-24-23-10-85-00-0C-29-7B-A4-A5
   DNS Servers . . . . . . . . . . . : 2001:db8:acad:1::1
   NetBIOS over Tcpip. . . . . . . . : Enabled

C:\Windows\system32>ping 2001:db8:acad:1::1

Pinging 2001:db8:acad:1::1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 2001:db8:acad:1::1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

7.10. ábra: Windows10 hoszton (célpont) kiadott ipconfig /all kimenete

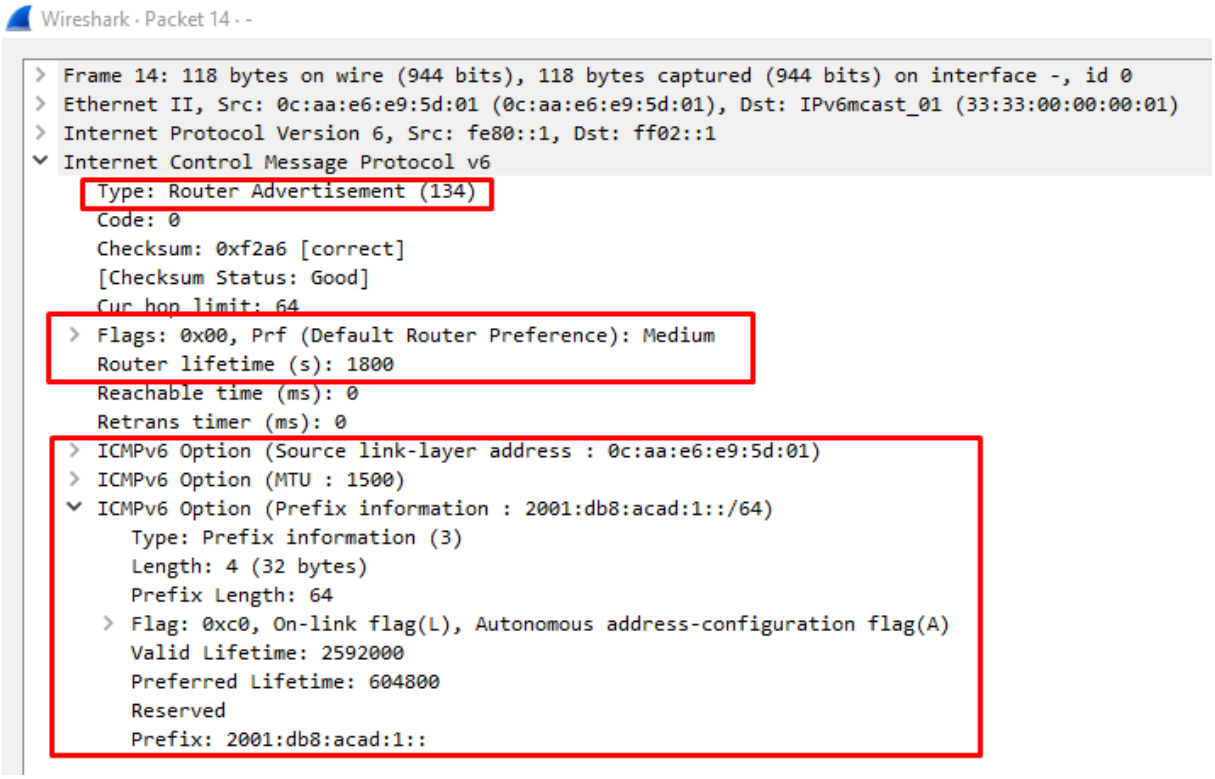
Windows 10 hoszton kiadva az *ipconfig /all* parancsot a fentebbi képen látható kimenet képződik, amelyen látszódik, hogy a manuálisan megadott *2001:db8:acad:1::3* IPv6 címet nem tudta használatba venni, mert duplikált címnek tekinti annak következtében, hogy a támadó jelezte hamis válaszüzenetekben, hogy az érintett címeket már használják.

Annak ellenére, hogy a támadó a generált link-local IP címek nevében is küld ki hamis Neighbor Advertisement üzeneteket, tapasztalatom szerint a Windows10 hoszt pár alkalommal megpróbálkozik új link-local cím generálásával, de végül egy ponton túl nem generál már újabb címeket, ennek ellenére ezzel a címmel a helyi hálózaton fog tudni küldeni üzeneteket, ellentétben az Ubuntu1 hoszt-tal, amely nem fog rendelkezni aktív link-local címmel.

A Security Onion az alapértelmezett szabálykészlete alapján egyik behatolás érzékelő modulja sem nem képes detektálni ez efféle, Neighbor Discovery Protocol-on alapuló támadásokat, így nem hoz létre riasztásokat sem. Ezért célom egy olyan megoldást alkalmazni, amely az ilyen típusú támadásokat képes detektálni és az olvasható naplófájlokat továbbítani a Security Onion részére, ahol tovább elemezhető a gyanús forgalom.

7.3.2. Denial Of Service támadás Router megszemélyesítésével

A támadott hálózatban az R1 nevű router az alapértelmezett IPv6-os router. A Neighbor Advertisement üzenetekhez hasonlóan a Router Advertisement üzenetek célcímeként az összes IPv6-ra alkalmas hoszt multicast (all nodes multicast) címe jelenik meg. A Neighbor Discovery protokoll két féle routerekkel kapcsolatos üzenetet definiál. Ezek a Router Solicitation és Router Advertisement üzenetek. A hálózatban aktív routerek periodikusan küldenek Router Advertisement üzeneteket, amelyekben a saját, illetve a hálózatban használatos paramétereket hirdet. Például a Default Router Preference-t, Router lifetime-ot (Router élettartam) és a router MAC címét. Továbbá az alhálózatban használatos IPv6-os prefixet, MTU-t, a konfigurációs flageket, amelyben a címgenerálás módszere van meghatározva. Ezek a paraméterek láthatóak az alábbi képen, amelyen az R1 által küldött Router Advertisement üzenet tartalma látható.

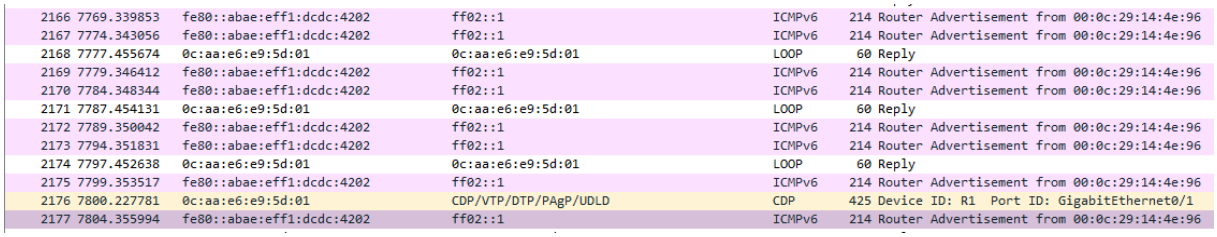


7.11. ábra: Router Advertisement üzenetekben meghatározott paraméterek

A második üzenettípust, a Router Solicitation üzeneteket pedig az újonnan bekapcsolt hosztok küldik, amelyben kérik az aktív routerektől, hogy küldjék el a Router Advertisement üzeneteket a küldött üzenetre válaszul, amelyek alapján automatikusan IPv6 címeket generálthatnak az üzenetben megszabott paraméterek alapján.

A támadó hoszton a támadás a *sudo atk6-fake_router6 eth0 2001:db8:acad:1::2/64* parancs segítségével indítható el, amelyben meg kell adni, hogy a PC melyik interfészén és milyen hálózati prefix-et szerepeltessen a csomagban, jelen esetben ez megegyezik a ténylegesen használt prefix-szel. A támadás megkezdését követően a hálózati forgalmat elemezve látható, hogy a *fe80:abae:eff1:dcdc:4202* link-local címről érkeznek a hamis Router Advertisement

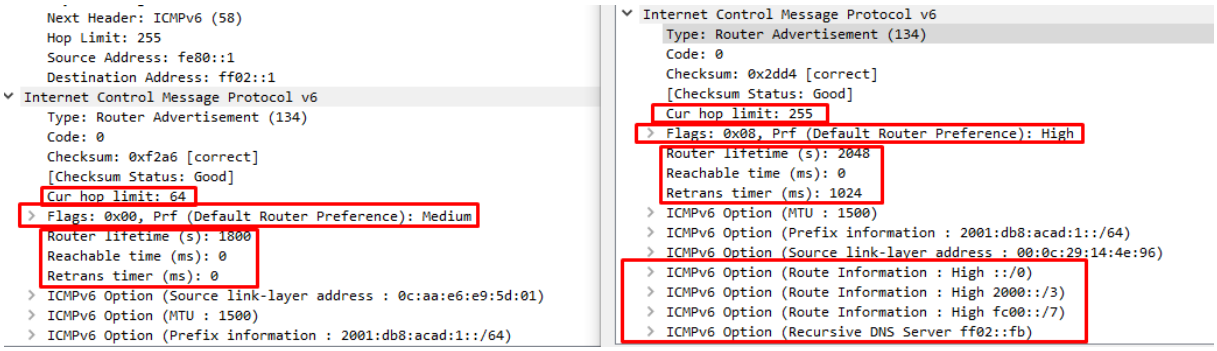
üzenetek öt másodpercenkénti gyakorisággal. Ez a támadó PC link-local címe, ezt a címet fogják a hosztok az alapértelmezett router címeként kezelni, az R1 router címe mellett. Az alábbi ábrán látható packet capture megtalálható a *RA_tamadas.pcapng* nevű mellékletben



2166	7769.339853	fe80::abae:eff1:dcdc:4202	ff02::1	ICMPv6	214 Router Advertisement from 00:0c:29:14:4e:96
2167	7774.343056	fe80::abae:eff1:dcdc:4202	ff02::1	ICMPv6	214 Router Advertisement from 00:0c:29:14:4e:96
2168	7777.455674	0c:aa:e6:e9:5d:01	0c:aa:e6:e9:5d:01	LOOP	60 Reply
2169	7779.346412	fe80::abae:eff1:dcdc:4202	ff02::1	ICMPv6	214 Router Advertisement from 00:0c:29:14:4e:96
2170	7784.348344	fe80::abae:eff1:dcdc:4202	ff02::1	ICMPv6	214 Router Advertisement from 00:0c:29:14:4e:96
2171	7787.454131	0c:aa:e6:e9:5d:01	0c:aa:e6:e9:5d:01	LOOP	60 Reply
2172	7789.350042	fe80::abae:eff1:dcdc:4202	ff02::1	ICMPv6	214 Router Advertisement from 00:0c:29:14:4e:96
2173	7794.351831	fe80::abae:eff1:dcdc:4202	ff02::1	ICMPv6	214 Router Advertisement from 00:0c:29:14:4e:96
2174	7797.452638	0c:aa:e6:e9:5d:01	0c:aa:e6:e9:5d:01	LOOP	60 Reply
2175	7799.353517	fe80::abae:eff1:dcdc:4202	ff02::1	ICMPv6	214 Router Advertisement from 00:0c:29:14:4e:96
2176	7800.227781	0c:aa:e6:e9:5d:01	CDP/VTP/DTP/PAgP/UDLD	CDP	425 Device ID: R1 Port ID: GigabitEthernet0/1
2177	7804.355994	fe80::abae:eff1:dcdc:4202	ff02::1	ICMPv6	214 Router Advertisement from 00:0c:29:14:4e:96

7.12. ábra: Hamis Router Advertisement üzenetek

Közelebbről megvizsgálva a hamisított Router Advertisement üzeneteket, több különbözőség is feltűnhet az eredeti, R1 által küldöttekhez képest. A legelső különbség a Current hop limit mező értékének a különbözősége, az eredeti csomagban ez az érték 64, míg a hamisítottban 255. Ez az érték írja elő a hálózat hosztjainak számára, hogy ilyen hop limit értékkel küldhetnek kimenő irányba üzeneteket. A legjelentősebb különbség az, hogy a Default Router Preference mezőben az R1 által használt medium érték helyett high van szerepeltetve. Olyan hálózatban, ahol több router is jelen van, ez a mező döntőszerepű az aktuális router megválasztásában. A hosztok a magasabb Router Preference értékkel rendelkező routert fogják az alapértelmezett átjárójukként alkalmazni.



```
    Next Header: ICMPv6 (58)
    Hop Limit: 255
    Source Address: fe80::1
    Destination Address: ff02::1
Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0xf2a6 [correct]
    [Checksum Status: Good]
    Cur hop limit: 64
  > Flags: 0x00, Prf (Default Router Preference): Medium
    Router lifetime (s): 1800
    Reachable time (ms): 0
    Retrans timer (ms): 0
  > ICMPv6 Option (Source link-layer address : 0c:aa:e6:e9:5d:01)
  > ICMPv6 Option (MTU : 1500)
  > ICMPv6 Option (Prefix information : 2001:db8:acad:1::/64)
```

```
Internet Control Message Protocol v6
    Type: Router Advertisement (134)
    Code: 0
    Checksum: 0x2dd4 [correct]
    [Checksum Status: Good]
    Cur hop limit: 255
  > Flags: 0x08, Prf (Default Router Preference): High
    Router lifetime (s): 2048
    Reachable time (ms): 0
    Retrans timer (ms): 1024
  > ICMPv6 Option (MTU : 1500)
  > ICMPv6 Option (Prefix information : 2001:db8:acad:1::/64)
  > ICMPv6 Option (Source link-layer address : 00:0c:29:14:4e:96)
  > ICMPv6 Option (Route Information : High ::/0)
  > ICMPv6 Option (Route Information : High 2000::/3)
  > ICMPv6 Option (Route Information : High fc00::/7)
  > ICMPv6 Option (Recursive DNS Server ff02::fb)
```

7.13. ábra: Eredeti RA (bal) üzenet és a hamisított (jobb) különbségei

További különbség, hogy Router lifetime mező 2048 másodpercre van állítva, míg az eredetiben ez az érték 1800, amely ennek a mezőnek az alapértelmezett értéke. Ez a mező megadja, hogy egy routert meddig kezelnek alapértelmezett routerként a hosztok. Abban az esetben, ha az aktuális router a megszabott időn belül nem küld újabb Router Advertisement üzenetet, amivel frissítheti ezt az időzítőt, akkor a továbbiakban a hosztok ezt a routert nem fogják aktív routerként kezelni. A megnövelt értékkel a támadó célja az, hogy ha valamilyen indokból kifolyólag a helyi hálózaton nem terjed az üzenete, akkor minél tovább szerepeljen a hosztok alapértelmezett átjárójaként.

Természetesen a Prefix opcióban a megjelölt MAC cím a támadó sajátjára mutat, ezen kívül sokkal nagyobb Valid lifetime (érvényes élettartam) és Preferred lifetime (preferált élettartam)

értékeket szerepeltet. Ezek a mezők szabják meg, hogy a hosztok által generált IP címek meddig érvényesek. A Preferred lifetime kötelezően kisebb vagy egyenlő értékkel rendelkezik, mint a Valid lifetime mező. A Preferred lifetime mezőben meghatározott időérték letelte után egy cím elavulttá (deprecated) válik és időszerűvé válik új cím generálása, de egészen addig még használható, amíg a Valid lifetime időtartama le nem telik.

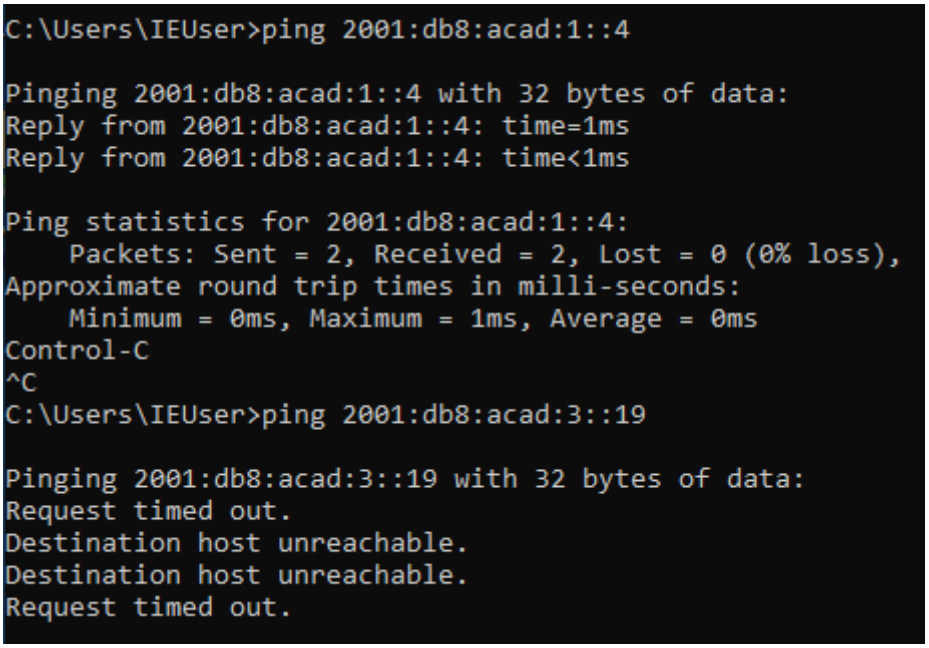
Végül a Retransmission timer (újraközvetítési időzítő) mező értéke is eltér, amelyben a router megszabhatja a hosztok számára, hogy milyen gyakorisággal küldhetnek Neighbor Solicitation üzeneteket a hálózatban. További eltérés, hogy a hamisított üzenetben több opcionális fejrész is van szerepeltetve, mint például a Route Information (Útvonal információ) és Recursive DNS server opciók. A Route Information-ben feltüntetett címeket (::/0, 2000::/3, fc00::/7) olyan hálózatoknak jelöl meg a küldő router, amelyekhez ismer útvonalat. Ezekkel a címekkel tulajdonképpen lefedhető a Global Unicast (2000::/3) és Unique Local (fc00::/7) IPv6 címek tartománya és ezek mellett a megadott ::/0 című alapértelmezett útvonal tovább erősíti a támadónak azt a célját, hogy megadott hamis alapértelmezett routerhez irányítsa a hálózat forgalmát.

A részletezett csomagok miatt a támadó felveszi az alapértelemezett router szerepét a hálózatban a nagyobb preferencia beállítás miatt és ennek következtében a kifelé irányuló csomagok nem jutnak túl a hálózat peremén ugyanis az összes a támadóhoz érkezik, aki ezeket a csomagokat nem továbbítja. Annak ellenére, hogy a támadó router bír a nagyobb preferenciával, attól függetlenül a hosztok mindkét router által kiküldött Router Advertisement üzenetek alapján is generálhatnak címeket. Emiatt csak a távoli hálózatok elérhetősége szűnik meg, a helyi kommunikáció nem. A támadás megkezdését követően a Windows 10 hoszton a bejegyzett alapértelmezett útvonalok közé bekerül a korábban említett támadó link-local címe.

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Network Connection #2
   Physical Address. . . . . . . . . : 0C-AA-E6-B9-01-00
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2001:db8:acad:1::3(Preferred)
   IPv6 Address. . . . . . . . . . . : 2001:db8:acad:1:4803:c88e:93e3:28f0(Preferred)
   Temporary IPv6 Address. . . . . . : 2001:db8:acad:1:d9d8:832b:48af:2435(Preferred)
   Link-local IPv6 Address . . . . . : fe80::4803:c88e:93e3:28f0%14(Preferred)
   Autoconfiguration IPv4 Address. . : 169.254.40.240(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.0.0
   Default Gateway . . . . . . . . . : 2001:db8:acad:1::1
                                       fe80::1%14
                                       fe80::abae:eff1:dcdc:4202%14
   DHCPv6 IAID . . . . . . . . . . . : 118270694
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-24-23-10-85-00-0C-29-7B-A4-A5
   DNS Servers . . . . . . . . . . . : 2001:db8:acad:1::1
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

7.14. ábra: Windows 10 hoszton megjelenik a támadó alapértelmezett routerként

Mivel a csomagban aktiválva van az automata konfigurációs zászló, így a hosztok a hálózaton a hamis üzenetben feltüntetett prefixet is fogják alkalmazni, így ezek az információk alapján is fognak címeket generálni a hosztok. Jelen esetben mindkét router az eredeti prefixet hirdeti, így nem generálnak más alhálózatból származó IP címeket.



7.15. ábra: Támadás eredményeként elérhetetlenné válnak a távoli hálózatok

Az eredmény a fentebbi képen is látható, hogy az alhálózatban használatos 2001:db8:acad:1::4 címet sikeresen tudja pingelni a Windows 10 hoszt viszont a távoli, R2 alhálózatában lévő 2001:db8:acad:3:19-es címmel rendelkező Ubuntu2 hosztot nem. Ezek alapján a szolgáltatás megtagadással járó támadás sikeres volt.

7.4. Támadások detektálása

Ebben a fejezetben taglalom, hogy az előző alfejezetben megvalósított támadásokat milyen módszerekkel tudtam detektálni. Alapvetően az észleléshez a Suricata-t és az NDPMon szoftvereket használtam fel. Az egyénileg létrehozott Suricata szabályokat mutatom be, valamint, hogy a tesztelések alatt megszerzett információkból, hogyan tudtam növelni a detektálási hatékonyságot.

7.4.1. Duplicate Address Detection üzeneteken alapuló támadás detektálása

A szabályalapú detektáló rendszerekben lehetséges létrehozni olyan szabályokat, amelyek bizonyos pontossággal jelezhetik, hogy aktív támadás van folyamatban. A jelenlegi esetben leginkább azokat az információkat használtam fel, hogy például egy adott típusú csomagot (DAD támadás esetén NA üzeneteket) számtalanszor küld ki a hálózaton a támadó hoszt, azonos forráscímről.

Hasonló szabály megvalósítható egy Suricata szabály segítségével, ami a következőképpen nézhet ki:

```
alert icmp any any -> any any (itype: 136; msg:"Security Onion:Neighbour
Advertisement flooding"; threshold: type threshold, track by_src, count 2,
seconds 1; sid:1000001; rev:1;)
```

A szabály az alábbi táblázatban leírtak szerint értelmezhető:

Parancs	**Értelmezés**
Művelet (alert**)**	A Suricata egy riasztást fog generálni abban az esetben, ha a szabályban leírtak teljesülnek.
Protokoll (icmp**)**	Az icmp valamint az icmpv6 típusú forgalmat fogja figyelembe venni.
Forrás és cél címe, port-ja, forgalom iránya (any any -> any any**)**	Jelen esetben bármilyen forrásból érkező, bármilyen célba tartó csomagokat elemez.
ICMP üzenettípus száma (itype: 136**)**	ICMP üzenettípus adható meg a hivatalos szabványban megadott típusszámmal, jelen esetben ez 136 azaz, Neighbor Advertisement típusú üzeneteket vizsgál.
Üzenet (*msg:"Security Onion:Neighbour Advertisement flooding"*)	Riasztáshoz tartozó, felhasználó által megadott egyéni üzenet.
Küszöbérték (*threshold: type threshold***)**	Ezen kulcsszó használatával megadhatjuk, hogy minimum hány észlelés esetén generál riasztást.
Küszöbérték paraméterei (*track by_src, count 2, seconds 1***)**	A küszöbérték átlépést a forrás cím alapján állapítja meg. Jelen esetben ezt a küszöbértéket akkor lépi át, ha egy adott forrás címről, korábban definiált csomagok érkeznek, olyan módon, hogy azok egy másodpercen belül legalább kétszer jutnak célba.
Azonosító (*sid:1000001***)**	Az itt megadott azonosítószám alapján azonosítható be egyértelműen a szabály.
Verzió (*rev:1***)**	Ez a mező adja meg a szabály verziószámát, amit a készítője, egyel növel, ha változtatásokat végez a szabályon.

7.4. táblázat: DAD detektálásra használt szabály elemei és leírása

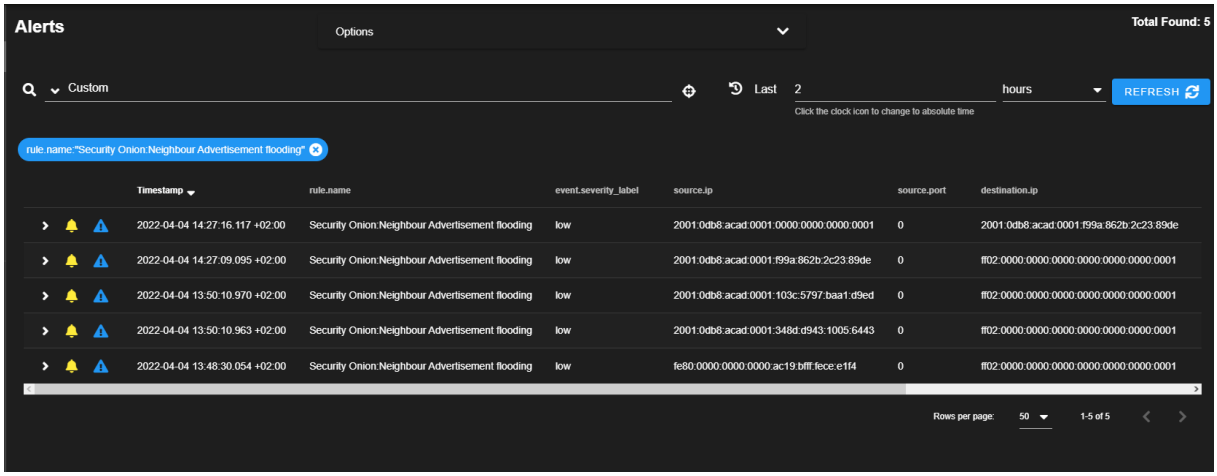
Első tesztek alapján a szabály megfelelően aktiválódott, amikor a Kali Linux DAD támadást indított az újonnan bekapcsolt Ubuntu1 hoszt ellen. Az említett hoszt nem tudta így aktiválni az IPv6 címeit. Tesztelési jelleggel végrehajtottam az Ubuntu1 és Windows10 hosztokkal a DAD procedúrát, ahol ugyanazt az IPv6 címet próbálják alkalmazni az érintett hosztok. A tesztelés során a Windows10 vette fel a célpont szerepét. Az Ubuntu1 megfelelően jelezte egy NA üzenetben, hogy az IP cím már használatban van, viszont a szabály ennek hatására nem aktiválódott. A tesztelés során feltűnt számomra, hogy a Windows10, amikor új IPv6 címet aktiválna, akkor elsősorban kiküld egy Neighbor Solicitation üzenetet, amiben megkérdezi, hogy az adott IPv6 cím már használatban van vagy sem. Ha erre az üzenetre nem érkezik válaszüzenet, akkor küld egy megegyező Neighbor Advertisement üzenetet, amiben jelzi, hogy ez a cím a saját MAC címén érhető el. Azonban, ez a Linux alapú rendszereknél (Ubuntu, Kali Linux) nem így történik és ha nem érkezik válasz az általuk kiküldött NS üzenetekre, akkor

nem küld ki saját maga válasz üzenetet (NA). Ez a működésbeli eltérés szerencsére nem befolyásolja a riasztások kiváltását és nem generálódnak fals pozitív riasztások. A tesztelés során vizsgált hálózati forgalom a mellékelt *DAD_Win10_teszt.pcapng* fájlban található.

A legtöbb hoszt a statikusan beállított IPv6 cím mellett generál 2-3 másik IPv6 címet a SLAAC procedúra keretein belül. Ezek közül az egyik biztosan a link-local IPv6 cím lesz, a további címek pedig dinamikusan generált global unicast címek. A címek száma és generálásának módja az futtatott operációs rendszertől függ jellemzően. Jelen esetben is az alkalmazott rendszerek eltérést mutattak ilyen téren is.

A korábban bemutatott szabály azért működhet, mert a Kali Linuxon futtatott *atk6-dos-new-ip6* script egy NS üzenetre legalább két azonos NA üzenettel válaszol. Mivel IPv6 címenként kettő válaszüzenetet küld a támadó, így garantálható, hogy a szabályban meghatározott küszöbértéket (1 másodperc alatt 2 csomag) meg fogja haladni a rosszindulatú forgalom.

Ezek alapján elmondható, hogy ezen szabály felhasználása jó indikáció lehet, hogy támadás történt, de egyértelműen nem segít a támadó beazonosításában. Ez amiatt van így, mert a támadó nem valós MAC címeket használ a hamisított üzenetekben, így az nem beazonosítható. Valós környezetben emiatt további manuális vizsgálat lenne szükséges, hogy megbizonyosodhassunk a támadás forrásáról.



7.16. ábra: Security Onion-ben megjelenő riasztások az új szabály hatására

A fentebbi képen az első bejegyzés formájában látható, hogy a router által küldött NA üzeneteket is érzékeli, ezt olyan módon lehetne kiküszöbölni, hogy a szabály megvizsgálja a válaszként küldött NA üzeneteket, hogy azokban nem szerepel a Solicited zászló. Ugyanis ez a zászló akkor aktív, amikor egy olyan NS üzenetre érkezik válaszul ez a csomag, amely nem a DAD folyamat része. Ilyen zászlóval ellátott üzeneteket küldenek a hosztok, ha meg akarják állapítani egy adott IPv6 címhez tartozó MAC címet.

További különbség, hogy a DAD procedúrában használatos NA üzenetek cél címeként az FF02::1-es multicast cím van megadva minden esetben, amellyel az összes IPv6-os állomás megcímezhető egyszerre. Hagyományos NA üzenet esetén pedig a cél cím megegyezik az NS üzenetet küldő hoszt címével, ahogyan ez az ábra első bejegyzésében látható.

Alapvetően háromféle flag létezik egy ICMPv6 üzenet esetében, ezek bitsorrendben: Router, Solicited, Override. A router zászló akkor aktív egy üzenetben, ha az üzenet feladója egy router. A solicited zászló akkor aktív, ha a NA üzenetet kimondottan egy NS üzenetre küldik válaszul, amely jellemzően akkor történik meg, ha a küldő a címzett IPv6 címéhez tartozó MAC címét szeretné megismerni. Az override zászló funkcionalitása pedig az, hogy felülírja a gyorsítótárakban tárolt MAC címeket és ezért az eszközök mindenképpen a megcímzett IPv6 címhez az NA üzenetet küldő hoszt MAC címét fogja társítani.

Célszerű olyan módon módosítani a szabályt, amivel kizárható, hogy olyan NA üzenetet is vizsgáljon, amely nem a DAD procedúra során jött létre. Kutatásom alapján a Suricata erre nem biztosít egyszerű lehetőséget, emiatt szükséges az IPv6 csomagokban az 58-61. bájtokat vizsgálni, amelyek leírják a flag-ek állapotát. Ezek az 58. bájt felső három bitjének feleltethetőek meg, a további bitek egészen a 61. bájtig további, jövőbeli változtatások (reserved) esetére vannak fenntartva. Ezek a bitek az ICMPv6-os fejlécen belül az 5-8. bájton helyezkednek el. Az ezek alapján módosított Suricata szabály az alábbi módon valósítható meg:

```
alert icmp any any -> any any (itype: 136; msg:"Security Onion:Neighbour
Advertisement flooding"; icmpv6.hdr; content:"|20 00 00 00|"; threshold:
type threshold, track by_src, count 2, seconds 1; sid:1000001; rev:2;)
```

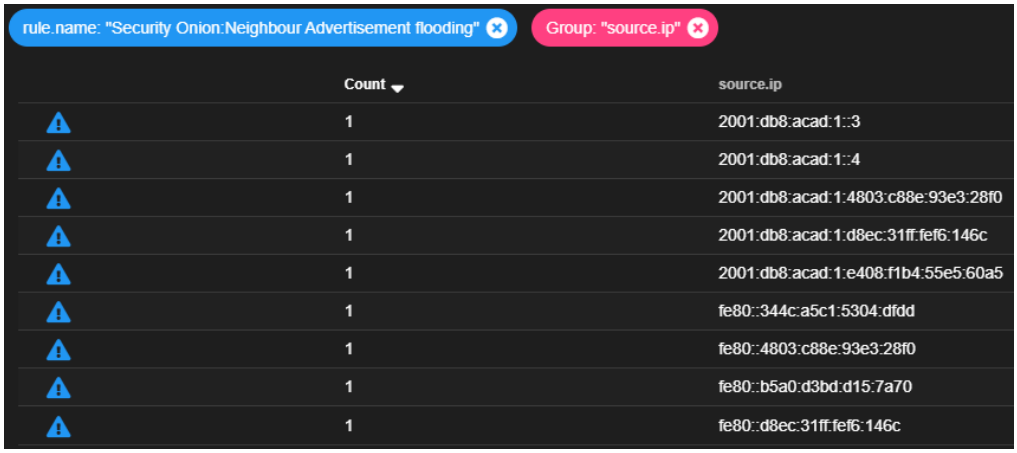
A szabály módosítása során az icmpv6.hdr és content mezők kerültek hozzáadásra, amelyek segítségével megadható, hogy az ICMPv6 fejlécen belül milyen bitsorozati mintát illesszen a csomagokra. A szabályban ez hexadecimális értékként van feltüntetve, amely bináris formában így jeleníthető meg: 00100000 00000000 00000000 00000000. Ezen négy bájtból az első bájt, felső három bitje lényeges az aktív flag-ek szempontjából. Látható, hogy a harmadik bit-nek az értéke egyes, tehát a három közül csak az override zászló lesz aktív. A DAD procedúra során küldött NA üzenetek rendszerint csak ezt a flag-et használják fel, ezáltal megkülönböztethető a hagyományos NA üzenetektől. A korábban említett esetben a solicited zászlót emeltem ki, de azt nem célszerű vizsgálni, hogy a solicited zászló nem aktív, mert úgy nem fedhető le egyszerűen az összes eshetőség.

```
┌──(kali㉿kali)-[~]
└─$ sudo atk6-dos-new-ip6 eth0
Started ICMP6 DAD Denial-of-Service (Press Control-C to end) ...
Spoofed packet for existing ip6 as 2001:db8:acad:1::4
Spoofed packet for existing ip6 as fe80::d8ec:31ff:fef6:146c
Spoofed packet for existing ip6 as fe80::4803:c88e:93e3:28f0
Spoofed packet for existing ip6 as 2001:db8:acad:1::3
Spoofed packet for existing ip6 as fe80::344c:a5c1:5304:dfdd
Spoofed packet for existing ip6 as 2001:db8:acad:1:4803:c88e:93e3:28f0
Spoofed packet for existing ip6 as 2001:db8:acad:1:e408:f1b4:55e5:60a5
Spoofed packet for existing ip6 as 2001:db8:acad:1:d8ec:31ff:fef6:146c
Spoofed packet for existing ip6 as fe80::b5a0:d3bd:d15:7a70
```

7.17. ábra: DAD támadás konzolos kimenete

A támadást megindítását követően a támadó konzolján megjelennek (fenti ábra) azok az IPv6 címek, amelyeknek a nevében küldött NA üzeneteket a támadó. A változtatás hatására a korábban említett legitim router által küldött üzenetek nem fognak riasztást eredményezni,

ahogyan ez a riasztási felületről készült képen is látható. Összehasonlítva a támadó konzolos kimenetét és az alább látható ábrán a detektált támadásokat, megállapítható, hogy a Security Onion megfelelően detektálta az összes elküldött hamis üzenetet.



7.18. ábra: Szabály módosítása után megjelenő riasztások

Alapvetően kijelenthető, hogy a DAD folyamaton alapuló támadásokat lehetséges detektálni a bemutatott környezetben hagyományos megoldásokkal is, ezek alatt a szabályalapú és aláírásokon alapuló detektálási módszereket értem. Azonban ezek a megoldások akkor tudnak igazán effektívek lenni, ha a szabályok definiálását megelőzően ismerjük a támadások paramétereit.

Fontos megemlíteni, hogy nem minden esetben azonosítható a támadás indítója, például a DAD támadás esetében is ez a helyzet. Ez abból fakad, hogy a támadó a valós forgalommal megegyező csomagokat küld ki a hálózaton és forrás MAC címnek hamis, generált címeket használ fel minden alkalommal és a megszemélyesítendő gép leendő IPv6 címét szerepelteti forrás IP címként.

A hagyományos módszerek mellett léteznek az anomália alapú detektálási módszerek, valamint a különféle mesterséges intelligenciát alkalmazó alternatívák. Az anomália alapon detektáló rendszerek alapvetően úgy működnek, hogy a hálózat normális működése közben feltérképezi a hálózatot, adatokat gyűjtenek a hálózatban kiküldött csomagokból, amelyből adatbázisokat építenek fel és ha a detektálás során ettől eltérő működést érzékelnek, akkor különféle riasztásokat váltanak ki. Az egyik ilyen anomália érzékelésen alapuló szoftver például az NDPMon, amely mellett döntöttem, hogy alkalmazni fogok annak érdekében, hogy aktívan lehessen detektálni az NDP protokollon alapuló támadásokat, ezek között szerepel a DAD procedúrát felhasználó támadások és hamis RA üzenetek felderítése.

Az NDPMon szoftver segítségével detektálhatóvá válik az Neighbor Discovery Protocol (NDP) sérülékenységeit kiaknázó támadások nagyrésze. A szoftver anomália alapú meghatározás elvén működik, tehát szükséges egy alapértelmezett, szokványos környezetet megismernie, amelyben nincsen jelen rosszindulatú forgalom. A program egy betanulási fázis során feltérképezi a hálózatot és adatbázisokban eltárolja az adott alhálózati információkat. Ilyen

például az aktív IPv6-képes router információi, azaz az a router, amely bizonyos időközönként Router Advertisement üzeneteket küld a hálózatban. Az ebben a csomagban fellelhető mezőket tárolja el, mint például: router link-local IPv6 címe, router lifetime, MTU, prefix, zászlók stb. Ezek mellett egy másik adatbázisban tárolja a szomszédok adatait, tehát egy táblázatot hoz létre, amiben párban tárolja az adott hoszt MAC címét és az aktív IPv6 címeit, valamint ezen kívül a bejegyzés dátumát és a MAC cím első 24 bitje alapján meghatározott hálózati kártya gyártóját. Ha egy NDP üzenetet gyanúsnak vél, akkor syslog naplófájlt generál, a gyanús forgalom adataival.

Az NDPMon szoftver a Security Onion virtuális gépre került telepítésre. A szoftver installációját és konfigurációját követően a *sudo ndpmon -L* paranccsal indítható el a betanulási fázis, amelynek során a szoftver összegyűjti a szükséges információkat a hálózat elvárható működéséről. Ezeket az adatokat a helyben futtatott weboldalon meg is tekinthetjük.

Probe	
Name	ens34
Type	interface
Countermeasures enabled	0
Router	
MAC Address	c:aa:e6:e9:5d:1
Link Local Address	fe80::1
Hop Limit	64
Reserved Flags	0
Router Lifetime	1800
Reachable Timer	0
Retransmit Timer	0
MTU	1500
Volatile	1
Prefixes Announced	
Prefix	2001:db8:acad:1::/64
Reserved Flags	192
Valid Lifetime	2592000
Preferred Lifetime	604800
IPv6 Global Addresses	

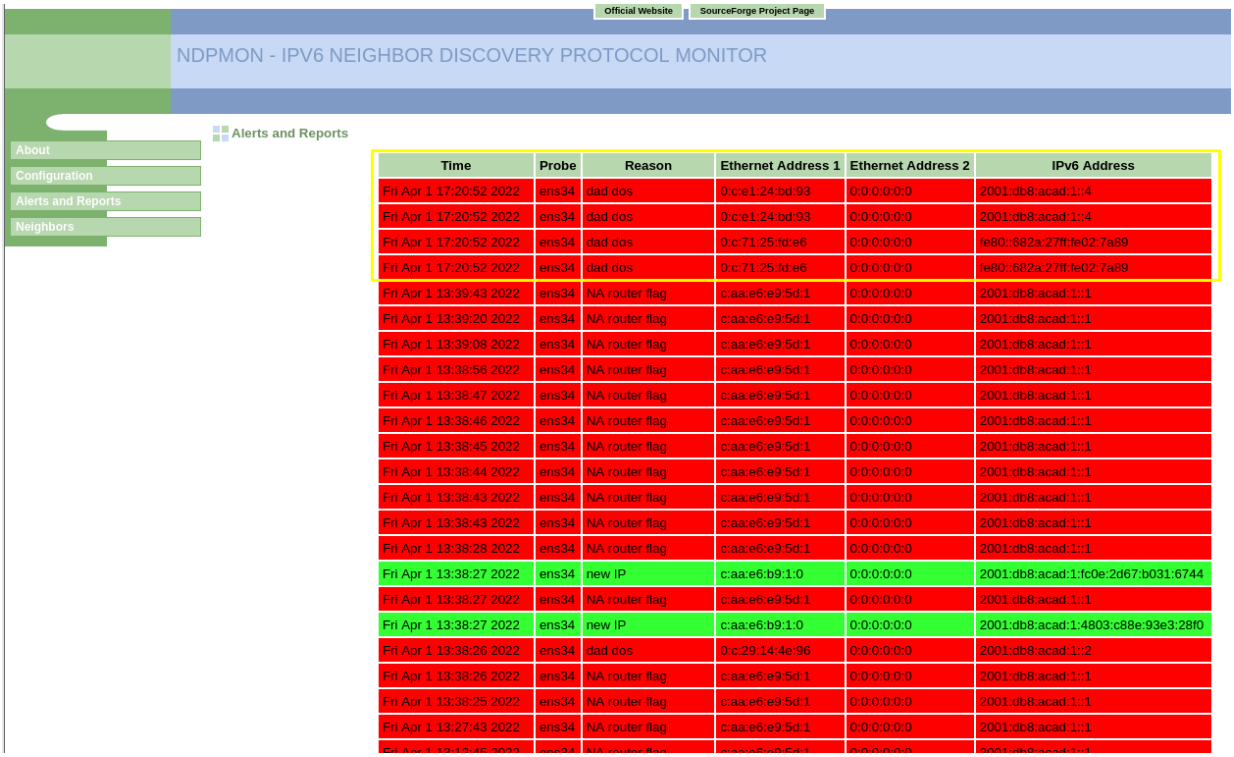
7.19. ábra: NDPMon webes adminisztrációs oldalán megjelenő információ a Router Advertisement csomagok alapján

Az alábbi ábra alapján látszik, hogy a szoftver feltérképezte a szomszédok listáját, megjelenik benne a Kali Linux, Windows10, Security Onion, Ubuntu1 hosztoknak a MAC címei és a hozzájuk tartozó statikusan megadott és dinamikusan generált IPv6-os címek. Ezek mellett a fentebbi ábrán pedig az látható, hogy a router által Router Advertisement üzenetekben hirdetett paramétereket is helyesen tárolta el az NDPMon.

Probe Name	MAC Address	IPv6 Global Addresses	
ens34	0:c:29:4b:4c:cd		
Time	**MAC Address (Vendor)**	**Link Local Address**	**IPv6 Global Addresses**
Fri Apr 1 16:51:22 2022	0:50:56:c0:0:3 (Vmware)	fe80::1109:559:b278:3cf3	2001:db8:acad:1:d924:abba:6b2c:8788 (Fri Apr 1 12:29:26 2022 ;Fri Apr 1 16:51:20 2022) 2001:db8:acad:1:1109:559:b278:3cf3 (Fri Apr 1 12:29:27 2022 ;Fri Apr 1 16:51:22 2022)
Fri Apr 1 16:49:40 2022	c:aa:e6:e9:5d:1 (unknown)	fe80::1	2001:db8:acad:1::1 (Fri Apr 1 12:30:48 2022 ;Fri Apr 1 16:43:16 2022)
Fri Apr 1 14:31:27 2022	c:aa:e6:b9:1:0 (unknown)	fe80::4803:c88e:93e3:28f0	2001:db8:acad:1:b016:c100:6835:d32a (Fri Apr 1 12:30:48 2022 ;Fri Apr 1 13:08:12 2022) 2001:db8:acad:1:fc0e:2d67:b031:6744 (Fri Apr 1 13:38:27 2022 ;Fri Apr 1 14:31:17 2022) 2001:db8:acad:1:4803:c88e:93e3:28f0 (Fri Apr 1 13:38:27 2022 ;Fri Apr 1 14:00:49 2022) 2001:db8:acad:1::3 (Fri Apr 1 14:07:42 2022 ;Fri Apr 1 14:08:25 2022)
Fri Apr 1 16:42:56 2022	0:c:29:4b:4c:c3 (Vmware)	fe80::b290:3015:366c:839b	2001:db8:acad:1::10 (Fri Apr 1 12:35:47 2022 ;Fri Apr 1 16:42:56 2022)
Fri Apr 1 14:07:42 2022	42:2d:f3:fc:5e:84 (unknown)	::	2001:db8:acad:1:402d:f3ff:fefc:5e84 (Fri Apr 1 13:57:03 2022 ;Fri Apr 1 14:07:42 2022)
Fri Apr 1 16:47:40 2022	0:c:29:14:4e:96 (Vmware)	fe80::abae:eff1:dcdc:4202	2001:db8:acad:1::2 (Fri Apr 1 14:02:22 2022 ;Fri Apr 1 16:47:40 2022)
Fri Apr 1 16:43:16 2022	5a:62:13:9e:1b:38 (unknown)	fe80::5862:13ff:fe9e:1b38	2001:db8:acad:1:5862:13ff:fe9e:1b38 (Fri Apr 1 16:42:04 2022 ;Fri Apr 1 16:42:09 2022) 2001:db8:acad:1::4 (Fri Apr 1 16:43:16 2022 ;Fri Apr 1 16:43:16 2022)

7.20. ábra: NDPMon által számontartott szomszédok listája

A betanulási fázist követően a *sudo ndpmon* paranccsal elindítva a programot, monitoring módba lép és aktívan rögzíti a riasztásokat a futtató rendszer syslogjában. Miközben a támadó gépén folyamatban lévő DAD támadás folyik, a következő jelenik meg az NDPMon „Alerts and Reports” webes aloldalán.



Time	Probe	Reason	Ethernet Address 1	Ethernet Address 2	IPv6 Address
Fri Apr 1 17:20:52 2022	ens34	dad dos	0:c:e1:24:bd:93	0:0:0:0:0:0	2001:db8:acad:1::4
Fri Apr 1 17:20:52 2022	ens34	dad dos	0:c:e1:24:bd:93	0:0:0:0:0:0	2001:db8:acad:1::4
Fri Apr 1 17:20:52 2022	ens34	dad dos	0:c:71:25:fd:e6	0:0:0:0:0:0	fe80::682a:27ff:fe02:7a89
Fri Apr 1 17:20:52 2022	ens34	dad dos	0:c:71:25:fd:e6	0:0:0:0:0:0	fe80::682a:27ff:fe02:7a89
Fri Apr 1 13:39:43 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:39:20 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:39:08 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:56 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:47 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:46 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:45 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:44 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:43 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:43 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:28 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:27 2022	ens34	new IP	c:aa:e6:b9:1:0	0:0:0:0:0:0	2001:db8:acad:1:fc0e:2d67:b031:6744
Fri Apr 1 13:38:27 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:27 2022	ens34	new IP	c:aa:e6:b9:1:0	0:0:0:0:0:0	2001:db8:acad:1:4803:c88e:93e3:28f0
Fri Apr 1 13:38:26 2022	ens34	dad dos	0:c:29:14:4e:96	0:0:0:0:0:0	2001:db8:acad:1::2
Fri Apr 1 13:38:26 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:38:25 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1
Fri Apr 1 13:27:43 2022	ens34	NA router flag	c:aa:e6:e9:5d:1	0:0:0:0:0:0	2001:db8:acad:1::1

7.21. ábra: NDPMon webes adminisztrációs felületén megjelenő riasztások, táblázat elején a DAD támadás hatására kiváltott riasztások láthatóak

Az említett riasztási felületen látható, hogy helyesen megjelennek a DAD támadásra utaló jelek, ez a felső négy piros háttérrel ellátott bejegyzés, amelyekben a kiváltó okként „dad dos” van

szerepeltetve. A táblázatban zöld háttérrel jelennek azok a bejegyzések, amelyek tájékoztató jellegűek, jelen esetben ezek a bejegyzések azt jelölik, hogy a jelölt MAC címhez kapcsolódóan új IPv6 címek jelentek meg a hálózatban. Egy, a képen nem látható, harmadik színnel, narancssárgával jelöli a gyanús üzeneteket, amelyek szokatlanok, de támadásnak nem minősíthetőek. Ilyen például, ha egy adott IPv6 cím tulajdonosaként egy másik MAC cím tűnik fel a hálózaton. Ebből arra lehetne következtetni, hogy egy rosszindulatú hoszt más eszköz MAC címét hamisítja meg és használja fel üzenetek küldésénél.

Annak érdekében, hogy a riasztásokat központilag lehessen felügyelni, így célszerű az NDPMon által generált syslog fájlokat továbbítani a Security Onion számára. A Security Onion a Filebeat nevű komponensének segítségével gyűjti be a különböző logokat generáló komponenseitől a naplófájlokat. A gyári konfiguráció lehetővé teszi, hogy a Filebeat syslog fájlokat fogadjon a virtuális gép 514-es TCP és UDP portjain. Az rsyslog megfelelő konfigurációjával az NDPmon által írt naplófájl továbbítható az Elasticsearch felé és így megjeleníthetővé válik a Security Onion webes felületén. A következő rsyslog konfigurációval megfelelően továbbíthatóak az érintett bejegyzések:

```
:programname, contains, „NDPMon” @@0.0.0.0:514
```

A konfiguráció célja, hogy a syslog azon bejegyzései, amelyeket az NDPMon program írt, azokat a 514-es TCP porton továbbítsa helyileg, amelyeket a Filebeat fogadhat. A konfiguráció életbelépését követően és az NDPMon-t monitorozó módban elindítva a log fájlok megjelentek a Security Onion felületén.

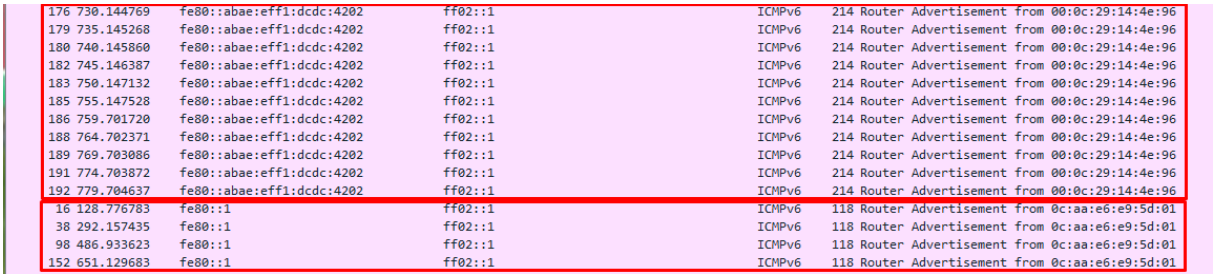
7.22. ábra: Security Onion-ben megjelenő NDPMon syslog bejegyzései

7.4.2. Router Advertisement csomagokon alapuló támadás detektálása

Az előző fejezetben megismert hasonló módokon lehetséges detektálni a Router Advertisement üzenetekre építő támadásokat is. Ebben az esetben is kijelenthető, hogy az alapértelmezett szabálykészlet alapján a Suricata az efféle támadásokat sem képes deteltálni. Hasonlóképpen létrehozható egy Suricata szabály, amely a kiküldött Router Advertisement üzenetek gyakoriságát vizsgálja. Az előző fejezetben felhasznált szabály mintájára a következő szabály hozható létre:

```
alert icmp any any -> any any (itype: 134; msg:"Security Onion:Router Advertisement flooding"; threshold: type threshold, track by_src, count 2, seconds 1; sid:1000002; rev:1;)
```

Különbség az előzőhöz képest, hogy itt a szerepeltetett itype-nak 134-es érték van megadva, azaz a Router Advertisement üzeneteket vizsgálja. Hasonló módon van meghatározva itt is küszöbérték, tehát ha ugyanarról a forráscímről érkezik kettő ilyen típusú üzenet egy másodpercen belül, akkor riasztást vált ki.



7.23. ábra: Eredeti (lent) és hamisított (fent) Router Advertisement üzenetek

A hálózati forgalmat megfigyelvén látható, hogy az R1 local-link címéről (fe80::1) érkező Router Advertisement üzenetek nagyjából 180 másodpercenként ismétlődnek, míg a támadó link-local címéről (fe80:abae:eff1:dcdc:4202) érkező csomagok 5 másodpercenként. Az eredetileg meghatározott érték nem bizonyult megfelelőnek a detektálás szempontjából. Ezek alapján célszerű módosítani a szabályt oly módon, hogy a valós router által küldött csomagokat ne érzékelje rosszindulatú forgalomnak, csak a támadó által küldötteket.

A leírták alapján módosított Suricata szabály így valósítható meg:

```
alert icmp any any -> any any (itype: 134; msg:"Security Onion:Router Advertisement flooding"; threshold: type threshold, track by_src, count 3, seconds 30; sid:1000002; rev:2;)
```

A küszöbértéket úgy módosítottam, hogy 30 másodpercen belül háromszori érzékelés után vált ki riasztást a szabály. Ennek a változtatásnak köszönhetően, aktív támadás közben a riasztás megfelelően végbemegy, ahogyan az alábbi képen is látszódik.

7.24. ábra: Router Advertisement flooding szabály hatására kiváltott riasztások

A szabály megfelelően érzékelte a rosszindulatú forgalmat, azonban, ha a támadó változtat azon, hogy milyen gyakorisággal küld Router Advertisement üzeneteket, akkor elméletileg a detektálás elkerülhető lehetne. Azonban a támadó által alkalmazott *atk6-fake_router6* script alapértelmezetten öt másodperces intervallumot definiál, így a szabály ezeket megfelelően érzékeli.

Az NDPMon-nal való detektálás a korábban megtanult adatokra alapszik. Az előző alfejezetben a 7.18-as ábrán látható, hogy a normál működés közben feljegyezte az R1 által küldött Router

Advertisement üzenetekben található adatokat. Ezek alapján a helyi router címe a fe80::1-es link-local cím és hozzárendelve pedig 0c:aa:e6:e9:5d:01 MAC cím van. Ezek mellett a router a 2001:db8:acad:1::/64-es prefixet hirdeti a hálózatban, valamint megjelennek a korábban tárgyalt egyéb paraméterek is. A leírt paraméterek teljesen megegyeznek a ténylegesen alkalmazott paraméterekkel.

Az NDPMon-t monitorozó módban elindítva és a támadást elindítva Kali Linuxon, az NDPMon webes adminisztrációs felületén meg is jelennek a generált riasztások, „wrong ipv6 router" indokkal.

Time	Probe	Reason	Ethernet Address 1	Ethernet Address 2	IPv6 Address
Mon Apr 25 14:08:31 2022	ens34	wrong ipv6 router	0:c:29:14:4e:96	0:0:0:0:0:0	fe80::abae:eff1:dcdc:4202
Mon Apr 25 14:08:26 2022	ens34	wrong ipv6 router	0:c:29:14:4e:96	0:0:0:0:0:0	fe80::abae:eff1:dcdc:4202
Mon Apr 25 14:08:20 2022	ens34	wrong ipv6 router	0:c:29:14:4e:96	0:0:0:0:0:0	fe80::abae:eff1:dcdc:4202
Mon Apr 25 14:08:15 2022	ens34	wrong ipv6 router	0:c:29:14:4e:96	0:0:0:0:0:0	fe80::abae:eff1:dcdc:4202

7.25. ábra: NDPMon webes felületén hamis RA üzenetek hatására megjelenő riasztások

A bejegyzésekben látható a támadás ideje, ezáltal itt is megbizonyosodhatunk afelől, hogy a Router Advertisement üzenetek valóban öt másodpercenként érkeznek a támadó irányából. Ezek mellett megtalálható a támadó IPv6 és MAC címe is, amelyek valóban megegyeznek a támadó valós címeivel. Az IPv6 cím megegyezik a létrehozott szabály által kiváltott riasztásokban szereplő link-local címmel, azonban a Security Onion-ben megjelenő riasztások nem tartalmazzák a támadó MAC címét, ellentétben az NDPMon által generált riasztásokkal. Ezek a bejegyzések elmentésre kerülnek syslog üzenetekbe, amik továbbításra kerülnek a Security Onion adatbázisába, amelyek a lentebbi ábrán láthatóak.

7.26. ábra: Security Onion rendszerében megtalálhatóak az NDPMon riasztásai

7.5. Egyéb védelmi megoldások

A különféle védelmi megoldások metódusait a 4.3-as fejezetben részleteztem. Az általam szimulált két, Neighbor Discovery Protocol-t érintő támadás esetében leginkább az RA Guard és a csomagszűrő szabályok alkalmazása jelenthetnek valós megoldásokat. Az IPsec megfelelő konfigurálása a korábban leírtak alapján óriási korábbi tapasztalatot igényel és megvalósítása nagyon körülményes, így nem optimális alternatíva, ahogyan a SEND alkalmazása sem, ugyanis az említett PKI infrastruktúra kialakítása problémás a felhasznált virtuális környezetben.

7.5.1. RA Guard konfigurálása

Az RA Guard konfigurációja a szimulált hálózatban az IOS-el ellátott Switch-en lehetséges. Az alábbi parancsok konfigurációs módban való kiadásával megvalósítható egy olyan konfiguráció, amellyel a megadott portokról érkező Router Advertisement csomagokat eldobja.

Parancs	Hatás
`ipv6 nd raguard policy HOST`	Új „HOST” nevű RA Guard szabályrendszer létrehozása.
`device-role host`	A porton csatlakoztatott eszközök típusa. Host típus esetében a Switch eldobja a Router Advertisement és Redirect csomagokat.
`interface range GigabitEthernet0/0 - 3, GigabitEthernet1/0 - 3`	Megadott interfészeken kerül aktiválásra a megadott konfigurációval az RA Guard.
`ipv6 nd raguard attach-policy HOST`	Korábban létrehozott „HOST” szabályrendszer aktiválása.

7.5. táblázat: RA Guard konfigurációjának menete Cisco switchen

A fenti táblázatban megjelenített módon lehet konfigurálni úgy a hálózatban alkalmazott Cisco switchet, hogy azokat a Router Advertisement üzeneteket eldobja a switch, amelyek nem az R1 router irányából érkeznek.

8. EREDMÉNYEK ÉRTÉKELÉSE

Ez a fejezet a kialakított virtuális hálózatban végrehajtott szimulált támadások detektálásának körülményeit és eredményeit mutatja be. Részletezve lesz az indított támadásoknak a sikeressége a támadó szempontjából, valamint az alkalmazott detektálási módszerek megfelelősége.

8.1. Végrehajtott támadások hatása és effektivitása

A szakdolgozatomban két, a Neighbor Discovery protokoll üzeneteire építő támadást valósítottam meg. Ezek a Duplicate Address Detection (DAD) procedúrára építő és a Router Advertisement üzeneteket alkalmazó támadások. Mindkettő módszernek célja egy szolgáltatás megtagadással járó támadás végrehajtása a hálózatban.

A DAD támadással alapvetően a támadó elérte a célját, az újonnan bekapcsolt hosztok képtelenek voltak IP címeket generálni a SLAAC folyamat keretein belül, mivel a támadó minden címduplikációt megerősítő üzenetre olyan üzenettel reagált, amivel jelezte, hogy a hoszt által alkalmazni kívánt cím már foglalt. Emiatt az újonnan bekapcsolt hosztok képtelenek voltak hálózati kommunikációra. A kivételt a Windows 10 rendszer képzi, ugyanis link-local címet minden esetben tudott generálni jellemzően többedik próbálkozásra, még akkor is, ha mindegyik kérésre a támadó válaszol. Ez a link-local cím továbbra is használható helyi kommunikációra, nem úgy, mint az Ubuntu1 hoszt esetében, ugyanis az általa alkalmazni kívánt link-local címet nem tudta használatba venni, így nem volt elérhető a helyi hálózaton. A támadás részleteit a 7.3.1-es fejezetben részleteztem bővebben.

A Router Advertisement üzeneteket alkalmazó támadás célja alapvetően az, hogy egy hamis router vegyen részt a hálózat forgalmában, amelyet a hosztok nagyobb prioritással kezelnek, mint a valós routert. Innentől fogva ez a hamis router tetszőlegesen megadott konfigurációs paramétereket terjeszthetett a hálózatban az említett üzenetek segítségével. A támadás során megfigyelhető volt, hogy a hálózat hosztjainak irányító tábláiban az alapértelmezett útvonalnál megjelenik a támadó link-local címe is a valós router címe mellett, mint hálózati átjáró. A támadó által alkalmazott nagyobb preferencia miatt és a routing kiegészítő fejlécek miatt a legtöbb hoszt ezt a hamis routert alkalmazta alapértelmezett hálózati átjárójaként, ami azt eredményezte, hogy a távoli hálózatok elérhetetlenek lettek a hosztok számára, azzal a kivétellel, hogy a Windows 10-es hoszt másképpen reagált. Egy hosszabb pingsorozatot alatt a hálózati csomagokat megfigyeltem és amikor a Windows 10-es hoszt sikeresnek jelezte a ping-et, akkor valóban a valós átjárót alkalmazta, így az üzenet el is jutott a címzetthez. Tapasztalataim szerint ez viszonylag ritkán következett be, hozzávetőlegesen tíz alkalomból egyszer. Ezen támadás további részleteit a 7.3.2-es fejezetben részleteztem.

8.2. Alkalmazott detektálási módszerek hatékonysága

A felvázolt támadások detektálását két különböző irányból közelítettem meg. Mindkét esetben hálózat alapú behatolás-érzékelő rendszereket alkalmaztam. Az első alkalmazott módszer egy aláírásos módszert alkalmazó hálózat alapú behatolás-érzékelő rendszer, amely a Suricata nevet

viseli és a Security Onion komponensei közé tartozik. A másik alkalmazott módszer egy anomália alapú behatolás-érzékelő rendszer, amely az NDPMon néven ismert.

8.2.1. Szignatúra alapú detektálási módszer hatékonysága

A Security Onion a gyári beállítások szerint a Suricata komponenséhez az Emerging Threats Open elnevezésű, ingyenesen felhasználható szabálykészletet biztosítja. Tapasztalataim szerint az általam végrehajtott támadások egyikét sem képes detektálni a Security Onion a gyári beállítások használata esetén. Emiatt szükséges volt új szabályokat létrehoznom, amelyek képesek lehetnek detektálni az említett támadásokat. Mindkét esetben a szabály a lényeges üzenettípusok gyakori megjelenését vizsgálja a különböző címekről, tehát egy megadott küszöbérték átlépése után riaszt csak a rendszer. Ez akkor valósulhat meg, ha egy hoszt gyakran küldi az érintett üzenettípust, ami a normál működéstől a legtöbb esetben eltér, így feltételezhető, hogy egy rosszindulatú küldőtől származnak az üzenetek.

A Duplicate Address Detection támadás esetén a szabály az NDP protokoll azon ICMPv6 üzeneteit veszi figyelembe, amelyek Neighbor Advertisement típusúak. Az alkalmazott szabály felépítését korábban a 7.4.1-es fejezetben részleteztem. Megfigyelve a hálózat forgalmát, a támadó minden egyes Neighbor Solicitation üzenetre két, szinte egyidőben elküldött Neighbor Advertisement üzenettel válaszolt, ami eltér a szokványos működéstől. Miután a támadó megindította a támadást az első eredmények azt mutatták, hogy a készített szabály alapján a riasztások megfelelően kiváltásra kerültek a támadásoknak megfelelően, viszont előfordult, hogy a router által küldött, DAD procedúrában nem résztvevő Neighbor Advertisement üzenetek hatására is készülhettek riasztások. Emiatt olyan módon módosítottam a szabályon, hogy a DAD procedúra során alkalmazott üzeneteket vegye figyelembe egyedül. Ezt úgy értem el, hogy a releváns csomagokban kizárólag az override zászló jelenlétét vizsgáltam, amely minden esetben aktív, ha a DAD folyamat során küldik a hosztok és más esetekben pedig nem egyedül használatos. Ez a megoldás azért lehet alkalmas, mivel az üzenetek más típusú felhasználásánál, például címfeloldás során a solicited zászló mindig aktív, valamint a router zászló is szerepelhet.

Annak érdekében, hogy teszteljem, hogy normális működés során a szabály aktiválásra kerül vagy sem, több hoszt reakcióját is megfigyeltem a DAD procedúra során. Tehát a tesztelést úgy végeztem el, hogy a „támadó” szerepét egy legitim hoszt vette át, aki már rendelkezett azzal az IP címmel, amit az újonnan bekapcsolandó hoszt szeretett volna alkalmazni. A tesztelés során feltűnt számomra, hogy a Windows 10 hoszt másképp reagál abban az esetben, ha Neighbor Solicitation üzeneteire nem érkezik válasz, tehát abban az esetben, ha a hálózaton egyik hoszt sem jelzi, hogy a cím már használatban van. A működése olyan módon tér el, hogy válaszok hiányában ő maga küldi el a Neighbor Advertisement üzeneteket, amelyben hirdeti, hogy a hozzátartozó IP címek a saját MAC címén érhetőek el. Ez az eltérés a szabályi korábbi verziójában fals riasztások eredményeztek, azonban a végső verzióban nem befolyásolta a riasztások megfelelő kiváltását. A fentebb említettek alapján kijelenthető, hogy a DAD támadások detektálására használatos szabály megfelelően érzékelte a rosszindulatú forgalmat és a megszokott működést nem jelezte tévesen veszélyesnek az elvégzett tesztjeim alapján.

A hamis Router Advertisement üzeneteket felhasználó támadás detektálására készített szabályom hasonló módon az NDP protokollon belül alkalmazott ICMPv6 üzeneteket vizsgál, azon belül is a Router Advertisement üzenetek gyakoriságát. Működésének alapja az, hogy küszöbértékként, azt vizsgálja, hogy egy adott hálózati címről 30 másodpercen belül legalább 3 Router Advertisement üzenet érkezik. A szabály bővebb leírása a 7.4.2-es fejezetben található meg.

A támadás megindítását követően az eredetileg alkalmazott szabály nem bizonyult megfelelőnek. Miután részletesen kielemeztem a hálózati forgalmat, oly módon módosítottam a szabályon, hogy a támadás alapértelmezett paraméterei alapján riasztás lépjen életbe. Átlagos működés során a teszthálózaton áthaladó forgalmat figyelembe véve a szabály megfelelően érzékelte rosszindulatúnak a támadó irányából érkező forgalmat, míg a legitim forgalmat nem jelezte tévesen rossznak az eredetileg meghatározott hálózati architektúrában.

Miután megerősítettem, hogy általános működés közben megfelelően működik a létrehozott szabály, további lehetséges forgatókönyveket teszteltem. Az első teszt keretein belül 2-3 további Ubuntu hosztot adtam hozzá a hálózathoz. A teszt első lépése az volt, hogy az összes alhálózaton lévő hosztot egyszerre indítottam el, ugyanis az automata cím konfiguráció során a hosztok egy Router Solicitation üzentet küldenek ki a hálózaton a routernek, amelyekre Router Advertisement üzenetekkel kell reagálniuk. A tesztelés során előfordult olyan eset, amikor az előbb említett indokokból a valós router által küldött Router Advertisement üzenetek meghaladták a küszöbértéket, így a szabály ez alapján riasztást váltott ki. További tesztelések lennének szükségesek ahhoz, hogy még nagyobb felhasználó szám esetén mennyiben változna a kiküldött üzenetek száma. Ezek alapján a küszöbértéket célszerű lenne olyan módon módosítani, hogy az ilyen típusú forgalmat ne érzékelje.

Összegezve a hamis Router Advertisement üzeneteket detektáló szabály hatékonyságát, a riasztások nem minden esetben felelnek meg a valóságnak attól függően, hogy milyen egy hálózatnak a forgalma, hány hoszt vesz részt stb. A felhasznált küszöbértéket mindig hálózat specifikusan érdemes meghatározni.

8.2.2. Anomália alapú detektálási módszer hatékonysága

A szabályokkal való detektálás az esetek nagy részében megfelelően működött, azonban a szabályok megalkotásánál szükséges ismerni a támadások paramétereit és hálózat általános működését. Az anomália alapon detektáló rendszerek azonban megkönnyítik ezt a folyamatot és egy betanulási fázist igényelnek, amelyek akár komolyabb architekturális változások esetén újból futtathatóak. Emiatt nem szükséges a támadások pontos paramétereit ismerniük, mert tudnak alkalmazkodni a környezethez. Erre a tulajdonságára azonban negatívumként is lehet tekinteni, ha a hálózatban egy szokatlan, de valós változás megy végbe, akkor a rendszer fals pozitív riasztásokat generálhat. További problémát jelenthet a viszonyítási pont definiálása, ha ez nem megfelelően történik, akkor ebben az esetben is előfordulhatnak téves riasztások.

Az általam alkalmazott NDPMon szoftver egy anomália alapú behatolás-érzékelő rendszer, amely a Neighbor Discovery protokoll sérülékenységeit kiaknázó támadásokat képes

felismerni. A szoftver a betanulási fázis folyamán megismeri és eltárolja az alhálózatban található hosztok MAC és IPv6 címeit, azokat a valóságnak megfelelően egymáshoz rendeli. Emellett tárolja a helyi legitim router elérhetőséget, az általa periodikusan küldött üzenetekben terjesztett paramétereket.

A betanulási fázis alatt a szoftver helyesen eltárolta a helyi router paramétereit, viszont a szomszédsági „táblában” tárolt adatok hiányosak voltak, mivel a hálózat tétlen állapotban volt, így az nem generált jelentős forgalmat. Ennek következtében aktivizáltam a hosztokat, hálózati forgalmat generálva, ennek hatására a program az elvárt módon töltötte fel a szomszédsági táblát, az összes használatban lévő MAC címhez hozzárendelte az aktív IPv6 címeket.

Mindkét támadás esetében a rendszer megfelelően elkészítette a bejegyzéseket, amelyek az aktuális támadásra figyelmeztetnek. A bejegyzésekben megjelöli a támadásokat idejét, a kiváltó indokot és a támadó által alkalmazott MAC és IPv6 címeket. A megtanult adatoknak köszönhetően effektívebb alternatíva lehet a szabályalapú detektálásnál. Nem kell egyedül bizonyos üzenetek gyakoriságra hagyatkoznia, mivel a meglévő adatok alapján képes eldönteni egy csomagról, hogy támadó szándékú vagy sem.

A fejezetben említettek alapján elmondható, hogy mindkét detektálási módszer sikeresnek bizonyult az esetek nagyrészében. Azonban valós környezetben való alkalmazás esetén érdemes mérlegelni mindkét módszernek bizonyos előnyeit és hátrányait.

9. TOVÁBBFEJLESZTÉSI LEHETŐSÉGEK

Az alkalmazott detektálási módszerek megfelelőnek bizonyultak a vizsgált támadások esetében. Ennek ellenére így is fordulhat elő olyan eshetőség, amelyet az elvégzett tesztelések nem fednek le, amelynek hatására az alkalmazott rendszerek fals riasztásokat generálthatnak vagy egyáltalán fel sem ismernek bizonyos támadásokat. Ezek ellenére úgy gondolom, hogy annak az eshetősége, hogy mindkét módszer helytelenül érzékeljen minimális. Éppen ezért célszerű lehet egy olyan megoldást létrehozni, amellyel a két rendszer által küldött bejegyzéseket és riasztásokat össze lehetne hasonlítani, amely alapján még nagyobb bizonyosságot nyerhetne a detektálások helyessége. A két metódus riasztásait összehasonlítva felfedezhetőek lennének az eltérések, amelyek további manuális vizsgálatot igényelnének és azok az esetek amikor megegyeznek a bejegyzések paraméterei.

Az előző javaslattal kapcsolatosan célszerű lehet növelni a detektáló rendszerek számát is, amelyek más detektálási elven működnek. Például bővítési célzattal lehetne egy mesterséges intelligenciát alkalmazó behatolás-érzékelő rendszert is használatba venni. A rendszer betanítása sok időt és meglévő adatot vehet igénybe, azonban ezt követően a rendszer akár valós időben képes detektálni a támadásokat viszonylag nagy pontossággal a betanítás sikerességét figyelembe véve. Ezzel ki lehetne küszöbölni a különböző működési metodológiák hátrányait és kihasználni az előnyeiket. A rendszerek által generált eredményeket érdemes lenne összesíteni, amely alapján egy átfogó képet kaphatunk az egyes szenzorok riasztásairól és azok körülményeiről. A végső értékelést célszerű lehet akár egy biztonsági szakemberre bízni vagy akár egy félautomata rendszerben megvizsgálni, amely előre definiált paraméterek alapján megállapíthatja a végső eredményt.

Egy másik lehetőség a rendszer további automatizálását jelentené. A 4.3-as fejezetben említettem olyan védelmi módszereket, amelyeket akár egy-egy kiváltott riasztás esetében alkalmazni lehetne automatikusan a megfelelő eszközökön. Ez a rendszer úgy működhetne, hogy kiváltott riasztás esetén a riasztást tovább kellene propagálni valamilyen formában egy eseménykezelő rendszer felé, amely a riasztások bekövetkezése esetén különféle műveleteket tud ellátni. Ilyen lehet egy védelmi módszer életbe léptetése egy riasztás hatására. Az említett módszerek között szerepel olyan is, mint például a csomagszűrő szabályok vagy az RA Guard, amelyek akár Cisco által gyártott switcheken és routereken is aktiválhatóak. Egy riasztási esemény hatására le lehetne futtatni, olyan Python-ban íródott scripteket – például a Netmiko könyvtár felhasználásával – amelyek képesek SSH kapcsolatokat létrehozni a scriptben meghatározott eszközökkel, amelyen keresztül különféle IOS parancsokat lehetséges kiadni a megfelelő szekvenciában.

Ezzel a módszerrel dinamikusan lehetne kezelni az aktív védelmi módszereket attól függően, hogy a hálózatunk támadás alatt áll vagy sem. Érdemes lehet megvizsgálni a védelmi módszerek azon aspektusát, hogy azok mekkora erőforrást igényelnek az eszközök hardvereitől, ugyanis léteznek olyan védelmi konfigurációk, amelyek jelentős többletterhelést jelenthetnek az alkalmazóik számára. Az említett okokból kifolyólag nem feltétlenül célszerű, hogy egy adott defenzív megoldás mindig aktív legyen a hálózatban, ha arra éppen aktuálisan nincsen szükség.

10.ÖSSZEFOGLALÁS

A szakdolgozatom azzal a céllal készült, hogy ismertesse az IPv6 protokoll sérülékenységeit, főképpen azon sérülékenységeket kiaknázó támadások bemutatásával, amelyek IPv6 specifikusak. Továbbá a célok között szerepelt ezen támadások megfelelő detektálása. Az említett támadásokat a THC IPv6 Toolkit segítségével hajtottam végre, a rendszertervben részletezett virtuális hálózatban. Rengeteg IPv6 specifikus támadás indítható az említett eszközkészlet segítségével, ezen belül én két támadást választottam ki, amelyek kimondottan a Neighbor Discovery protokoll sérülékenységeire alapoznak. A támadások paramétereit és tulajdonságait a szimuláció közben hosszasan elemeztem, hogy ezek alapján megfelelő detektálási szabályokat hozhassak létre a Security Onion eszközeivel és egyéb megoldásokkal. Az elemzés során megismertem a Neighbor Discovery protokoll alapvető működését és azt, hogy az indított támadások a protokoll milyen aspektusát használják fel.

Gyakorlatilag a legtöbb behatolás-érzékelő rendszer rendelkezik alapvető IPv6 támogatottsággal, de ennek ellenére a támogatottsága nem közelíti meg az IPv4 protokollét. Ez a detektálási hatékonyságon is meglátszik, számos IPv6 támadást képtelenek detektálni a rendszerek. Annak érdekében, hogy az indított támadásokat detektáljam, két hálózat alapú behatolás-érzékelő rendszert alkalmaztam. A két rendszer különböző elveken működik, a Suricata szignatúra alapú detektálást alkalmaz, míg az NDPMon anomália alapon történő detektálást valósít meg.

Mindkét támadás észlelésére egyéni Suricata-kompatibilis szabályokat hoztam létre, amelyeket hosszas tesztelés alapján finomhangoltam. A tesztelési fázist megelőzően tanulmányoztam a hálózat általános működése közben és a támadás során lezajló hálózati forgalmat, amelyekben olyan tulajdonságokat kerestem, amely alapján a szabályok képesek lehetnek detektálni a releváns forgalmat. A tesztelés során meghatároztam különféle forgatókönyveket és a hozzájuk tartozó megvalósítandó célokat. Egészen addig módosítottam a szabályokon, amíg azok megfeleltek a felállított követelményeimnek. Összefoglalva, tapasztalatom alapján az elvégzett tesztekben a Suricata szabályok megfelelően működtek és helyesen váltottak ki riasztásokat.

A másik alkalmazott behatolás-érzékelő eszköz, az NDPMon is hasonló hatékonysággal működött, a kijelölt esetekben megfelelően érzékelte a támadásokat és ezen kívül további hasznos információkat szolgáltatott. Annak érdekében, hogy a két rendszer riasztásai és bejegyzései egy helyen legyenek áttekinthetőek, a szoftver által generált naplófájlokat továbbítottam a Security Onion részére, ahol akár értékelhetőek, összehasonlíthatóak a két rendszer által kiváltott riasztások azonosságai. Ez előbb említett felhasználási módszert a 9-es számú fejezetben továbbgondoltam, amely alapján ez a fajta összehasonlítás egy félautomata rendszerben is megtörténhet, ami végeredményében egy aggregált bírálatot állít elő a szenzorinformációk alapján.

Összegezve úgy gondolom, hogy a megvalósított detektáló eszközök az elvárásoknak megfeleltek a végrehajtott tesztek során és kijelenthető, hogy az indított támadásokat megfelelően detektálni tudtam a kialakított tesztkörnyezetben.

11.SUMMARY

Within my thesis I set out to describe the vulnerabilities of the IPv6 protocol, mainly those which are specific to IPv6. In order to do this I have launched network attacks, which are based on these vulnerabilities. Also detecting these attacks were among my set goals. I carried out these attacks with the help of the THC IPv6 Toolkit on the virtual network mentioned in the system design chapter. Using this tool, many IPv6 specific attacks can be launched. From these attacks I chose two, which are based on the vulnerabilities of the Neighbor Discovery Protocol. I have analysed the parameters and the properties of these attacks in order to define detection rules with the tools of Security Onion and other solutions. During my analysis I have familiarized myself with the operation of the Neighbor Discovery protocol and with the relevant vulnerabilities which are used by the launched attacks.

Virtually most intrusion detection tools have basic IPv6 support, but it is still outclassed by IPv4 support. This is also shown by the fact that many IPv6 attacks are undetectable by these tools. In order to detect the launched attacks, I have deployed two network based intrusion detection systems. These two systems work on different basis, the first one, namely Suricata uses signature based detection and the second one, NDPMon implements the anomaly based detection method.

I have created Suricata-compatible rules, which can be used to detect the mentioned attacks. In order to raise the efficiency of these, I have finetuned them during my lengthy testing period. Prior to the testing phase I have analysed the network traffic during usual network operation and during ongoing attacks. While I studied this traffic I looked for properties which could be used in the detection rules in order to detect relevant traffic. During the testing phase I have defined different scenarios with correspondig goals. I had kept modifying the created rules until they satisfied my previously defined requirements. Based on my experience while testing the scenarios the Suricata rules worked properly and they successfully generated alerts.

The other applied intrusion detection tool, NDPMon showed similiar efficiency to Suricata. It has properly detected the attacks in the pre-defined scenarios and also provided other useful information. In the interest of overviewing the alerts and the logs of these systems in one place, I have forwarded the logs generated by this application to Security Onion, where the alerts generated by these two systems could be evaluated based on their differences and similiarities. In the 9th chapter I gave further thoughts to this process and came to the conclusion that this comparison could be made in a semi-automatic fashion, where the system would produce an aggregated judgement based on sensor information.

In conclusion I think that the implemented detection methods satisfied the requirements during the testing period and it can be stated that I have detected the launched attacks successfully in the designed test environment.

12.IRODALOMJEGYZÉK

[1] http://www.agr.unideb.hu/~agocs/informatics/05_h_ecdl/ECDLweb/ecdlweb.uw.hu/m1-32.html utoljára megtekintve: 2021.04.28

[2] Abbate, J.E.: From ARPANET to Internet: A history of ARPA-sponsored computer networks, 1966 – 1988.

[3] https://bluecatnetworks.com/blog/mac-address-vs-ip-address-whats-the-difference/ utoljára megtekintve: 2021.04.28

[4] https://www.stackscale.com/blog/ipv4-exhaustion-solutions/ utoljára megtekintve: 2021.04.28

[5] https://www.stackscale.com/blog/whats-ipv6/ utoljára megtekintve: 2021.04.28

[6] OECD Digital Economy Papers: The Economics of Transition to Internet Protocol version 6 (IPv6), No. 244., 2014, 4-5.old.

[7] https://www.google.com/intl/en/ipv6/statistics.html utoljára megtekintve: 2021.05.01

[8] RFC791 INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION 1981 https://tools.ietf.org/html/rfc791

[9] RFC1883 Internet Protocol, Version 6 (IPv6) Specification, 1995 https://tools.ietf.org/html/rfc1883

[10] Levin, S. L., Schmidt, S.: IPv4 to IPv6: Challenges, solutions and lessons, 2014

[11] RFC1918 Address Allocation for Private Internets, 1996 https://tools.ietf.org/html/rfc1918

[12] https://www.netacad.com/courses/networking/ccna-introduction-networks utoljára megtekintve: 2021.05.02

[13] https://networklessons.com/cisco/ccna-routing-switching-icnd1-100-105/ipv4-packet-header utoljára megtekintve: 2021.05.07

[14] IEEE-USA: Next Generation Internet: IPv4 Address Exhaustion, Mitigation Strategies and Implications for the U.S., 2009

[15] Hagen, S.: IPv6 Essentials, 3rd Edition. *O'Reilly Media, Inc.*, 2014

[16] Frankel, S., Graveman, R., Pearce, J., Rooks, M.: Guidelines for the Secure Deployment of IPv6: Recommendations of the National Institute of Standards and Technology, 2010

[17] Amol, R., Sathyanarayhana, G., Rohan, K., Carlos, E. C.: Study of IPv6 Security Vulnerabilities, 2014

[18] Harith, A. D.: IPv6 Security Vulnerabilities, 2012

[19] Pilihanto, A.: A Complete Guide on IPv6 Attack and Defense, 2011

[20] Ullrich, J., Krombholz, K., Hobel, H., Dabrowski, A., Weippl, E.: IPv6 Security: Attacks and Countermeasures in a Nutshell, 2014

[21] https://www.ripe.net/support/training/material/ipv6-security/ipv6security-slides.pdf utoljára megtekintve: 2021.05.12

[22] https://www.si6networks.com/2017/01/19/a-tale-of-bad-decisions-weird-packets-and-dos-attacks/ utoljára megtekintve: 2021.05.14

[23] Schrötter, M., Scheffler, T., Schnor, B.: Evaluation of Intrusion Detection Systems in IPv6 Networks, 2019

Ábrahivatkozások:

[1] https://www.netacad.com/courses/networking/ccna-introduction-networks

[2] https://en.wikipedia.org/wiki/IPv4_address_exhaustion#/media/File:IPv4_exhaustion_time_line-en.svg

[3] https://www.ripe.net/support/training/material/ipv6-security/ipv6security-slides.pdf 29.dia

[4] https://docs.securityonion.net/en/2.3/_images/eval.png

13.ÁBRAJEGYZÉK

2.1. ábra: Hálózati rétegben történő becsomagolás [1]
2.1. táblázat: Az IPv4-es címosztályokhoz tartozó IP címtartományok és a hozzájuk tartozó alhálózati maszkok, Saját táblázat
2.2. ábra: Az IPv4 csomag fejlécének mezői [1]
2.2. táblázat: IPv4 címtartományok kimerülésének ideje az egyes regionális bejegyző szervezeteknél, Saját táblázat
2.3. ábra: Az IPv4-es címtér kimerülésével kapcsolatos események idővonala [2]
3.1. ábra: Tipikus global unicast IPv6 cím felbontás [1]
3.1. táblázat: Lerövidített példa IPv6 cím, Saját táblázat
3.2. ábra: Az IPv6 csomag fejléce [1]
3.2. táblázat: Automata konfigurációs folyamat menete, Saját táblázat
3.3. ábra: A kiegészítő fejlécek felépítése [1]
5.1. ábra: Routing Header Type 0 támadás [2]
6.1. ábra: Szimulált hálózat tervezett felépítése, Saját ábra
7.1. ábra: GNS3-ban megvalósított hálózat topológiája, Saját ábra
7.1. táblázat: R1 és R2 routerek konfigurációs beállításai, Saját táblázat
7.2. ábra: Security Onion evaluation architektúrájának felépítése [3]
7.2. táblázat: Hálózatban használt IPv4/IPv6 és MAC címek, Saját táblázat
7.3. ábra: sudo so-status parancs konzolos kimenete, Saját ábra
7.3. táblázat: Hálózat tesztelési célzattal kiadott parancsok és eredményeik, Saját táblázat
7.4. ábra: A Security Onion hoszt és Hub közötti vonalon aktív packet capture eredménye, Saját ábra
7.4. táblázat: DAD detektálásra használt szabály elemei és leírása, Saját táblázat
7.5. ábra: Security Onion webes felületén megjelenő riasztások, Saját ábra
7.5. táblázat: RA Guard konfigurációjának menete Cisco switchen, Saját táblázat
7.6. ábra: Port feltérképezést követően megjelenő bejegyzések, Saját ábra
7.7. ábra: DAD támadás során keletkezett keretek, Saját ábra
7.8. ábra: Hamisított MAC címet tartalmazó Neighbor Advertisement csomag, Saját ábra
7.9. ábra: Kali Linuxon indított atk6-dos-new-ip6 program kimenete, Saját ábra
7.10. ábra: Windows10 hoszton (célpont) kiadott ipconfig /all kimenete, Saját ábra
7.11. ábra: Router Advertisement üzenetekben meghatározott paraméterek, Saját ábra
7.12. ábra: Hamis Router Advertisement üzenetek, Saját ábra
7.13. ábra: Eredeti RA (bal) üzenet és a hamisított (jobb) különbségei, Saját ábra
7.14. ábra: Windows 10 hoszton megjelenik a támadó alapértelmezett routerként, Saját ábra
7.15. ábra: Támadás eredményeként elérhetetlenné válnak a távoli hálózatok, Saját ábra
7.16. ábra: Security Onion-ben megjelenő riasztások az új szabály hatására, Saját ábra
7.17. ábra: DAD támadás konzolos kimenete, Saját ábra
7.18. ábra: Szabály módosítása után megjelenő riasztások, Saját ábra
7.19. ábra: NDPMon webes adminisztrációs oldalán megjelenő információ a Router Advertisement csomagok alapján, Saját ábra

7.20. ábra: NDPMon által számontartott szomszédok listája, Saját ábra
7.21. ábra: NDPMon webes adminisztrációs felületén megjelenő riasztások, táblázat elején a DAD támadás hatására kiváltott riasztások láthatóak, Saját ábra
7.22. ábra: Security Onion-ben megjelenő NDPMon syslog bejegyzései, Saját ábra
7.23. ábra: Eredeti (lent) és hamisított (fent) Router Advertisement üzenetek, Saját ábra
7.24. ábra: Router Advertisement flooding szabály hatására kiváltott riasztások, Saját ábra
7.25. ábra: NDPMon webes felületén hamis RA üzenetek hatására megjelenő riasztások, Saját ábra
7.26. ábra: Security Onion rendszerében megtalálhatóak az NDPMon riasztásai, Saját ábra

14.MELLÉKLETEK

A releváns hálózati forgalmakat tartalmazó Wireshark-kal elmentett .pcapng kiterjesztésű packet capture fájlok (szokvanyos_forgalom, DAD_tamadas, DAD_Win10_teszt, RA_tamadas) az alábbi linken érhetőek el: GEZFBP_Bardi_Banati_MELLEKLET.zip