Mérési útmutató a „Hálózati protokollok vizsgálata lehallgatással (SNIFF)” című méréshez

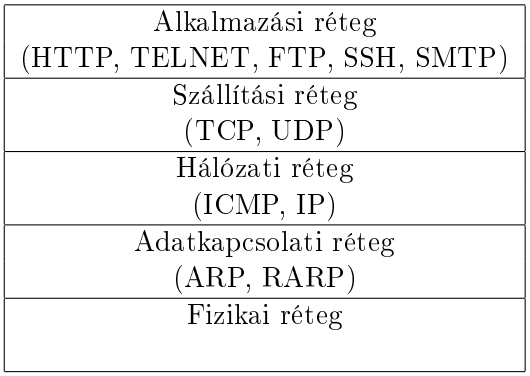
2016. február

A mérést kidolgozta:
Bencsáth Boldizsár és Holczer Tamás

BME, CrySyS Adat- és Rendszerbiztonság Laboratórium

1. Elméleti összefoglaló

A következőkben rövid áttekintést nyújtunk a TCP/IP hálózati architektúráról. Először szót ejtünk a mérés szempontjából is fontos protokollokról, majd néhány fontos hálózati elemet, berendezést mutatunk be.



1. ábra. TCP/IP hálózat rétegei

1.1. Protokollok

IP protokoll

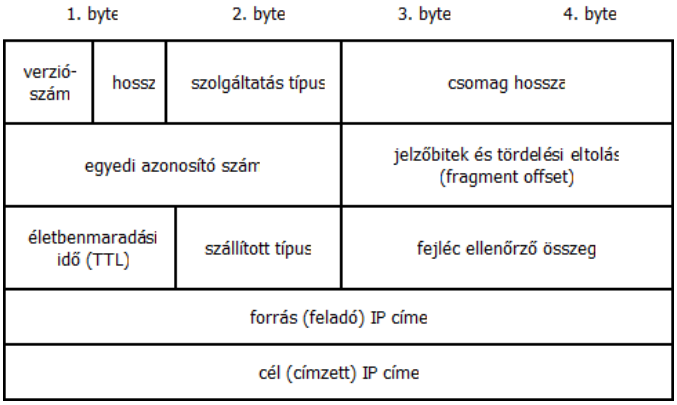
Az *Internet Protocol* (RFC 791) egy összeköttetés-mentes hálózati protokoll, önmagában nem nyújt garanciát arra, hogy az egyes csomagok valaha is célba érnek. Egy IP hálózatban minden eszköz rendelkezik egy ún. IP címmel, ez az ő azonosítója. Egy IP cím csak egy eszköz egy interfészéhez rendelhető, de bizonyos technikákkal (pl.: masquerading és aliasing) elérhető, hogy egy IP címhez több fizikai eszköz tartozzon. Természetesen, egy eszköz (pl.: számítógép) egyszerre több interfésszel is rendelkezhet.

Az IP cím 4 bájtból, vagyis 32 bitből áll (az IPv4 szerint). Általában négy 0 és 255 közötti decimális számmal ábrázoljuk, ezeket egymástól ponttal választjuk el (pl.: `145.157.2.15`). A négy bájtból álló IP címek túl rövidnek bizonyultak, ezért kidolgozták a 128 bitből álló IP címeket (IPv6). A mérés során a ma elterjedtebb IPv4-et használjuk.

Vizsgáljuk meg az IP fejléc tartalmát!

- A verzió mező különbözteti meg, hogy IPv4 vagy IPv6 csomagról van-e szó.
- A szolgáltatás típus mező azt jelzi, hogy a csomag milyen forgalmi paraméterekre érzékeny, de ezt a legtöbb implementáció figyelmen kívül hagyja.

- A DF jelzőbit jelentése do not fragment. Azon csomagokat, melyekben DF=1, a routerek nem darabolhatják fel.
- Az „életben maradási idő” (Time To Live – TTL) feladata a hálózat védelme. A feladó és a címzett között több hálózati eszközön, például routeren (magyarul útvonalválasztón) halad keresztül a csomag, ezen routerek minden továbbküldésnél a csomagban szereplő TTL értéket eggyel csökkentik. Végül, amikor a TTL nullává válik, nem küldik tovább a csomagot. A TTL így gátolja meg, hogy a csomag az idők végezetéig kóboroljon.
- A fejléc integritását ellenőrző összeg védi. A tartalom (payload) védelmére a TCP vagy az UDP fejlécben lévő ellenőrző összeg szolgálhat.
- A szállított típus a szállítási rétegi protokollt adja meg.
- Az opció mező hossza változó lehet, általában üres, az IPv4-ről IPv6-ra való áttérést segítheti, de egyéb célokra is alkalmas (pl.: forrásirányítás, útvonalrögzítés).



2. ábra. IPv4 fejléc

Az egyes hálózati csomópontokon belül a forgalomirányítás gyakorta statikus módon, útvonalválasztó táblák alapján történik. (Ezt minden TCP/IP implementációnak tartalmaznia kell.) Ekkor cél hálózatokat rendelünk hozzá interfészekhez, ez alapján történik az útválasztás. Előfordulhat az is, hogy a cél hálózatot egy ún. átjáróhoz rendeljük hozzá. Ekkor egy olyan gép a címét adjuk meg, amelyik majd tovább irányítja a forgalmat az adott cél hálózat felé.

Egy útvonalválasztó célberendezés esetében az útvonalválasztó tábla sok bejegyzést tartalmazhat, míg egy munkaállomás esetében gyakran csak kettőt: egyrészt a lokális hálózatot, másrészt egy alapértelmezett bejegyzést,

amely az összes többi (lokális hálózaton kívüli) célt egy routerhez irányítja. Ez a router az alapértelmezett átjáró (default gateway) a lokális hálózatból az Internet felé.

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
`Localnet`	`*`	`255.255.248.0`	`U`	`0`	`0`	`0`	`eth0`
`Default`	`router.sch.bme.hu`	`0.0.0.0`	`UG`	`0`	`0`	`0`	`eth0`

3. ábra. Útvonalválasztó tábla

Az adatcsomag irányításakor a berendezés összehasonlítja a csomag IP fejlécében szereplő CÉL címe mezőt az útvonalválasztó tábla CÉL (destination) mezőinek tartalmával. Az összehasonlítás során a berendezés nem feltétlenül egyezést keres. Azt vizsgálja, hogy a csomag uticélja bele tartozik-e a táblázatban definiált valamelyik tartományba. Az a sor (így az az interfész vagy útvonal) kerül kiválasztásra, amelyiknél a táblázat céltartománya a legszűkebb. Több azonos céltartomány esetén a kisebb metrikájú („metric") cím kerül kiválasztásra. Definiálunk átjárót is (amely a helyi hálózaton elérhető berendezés kell, hogy legyen), ekkor a berendezés az adatcsomagot az átjárónak küldi, és a továbbítást az végzi.

ARP

Az *Adress Resolution Protocol* (RFC 826) protokoll az adatkapcsolati rétegbez tartozik. Az IP címeket ethernet MAC címekkel rendeli össze. Tehát, a hálózati réteg címeihez adatkapcsolati rétegbe tartozó címeket rendel.

Ha egy berendezés IP csomagot szeretne küldeni, először megnézi a saját útvonalválasztó tábláját, és ez alapján eldönti, hogy az adott csomagot melyik interfészen kell elküldeni. Ezt követően ARP kérést ad ki ethernet broadcast címre, hogy megtudja, milyen MAC cím tartozik az adott célgép vagy átjáró IP címéhez. Az adott hálózaton minden berendezés megkapja a broadcast üzenetet és feldolgozza azt. Az, amelyiknek az IP címe megegyezik a keresett IP címmel, visszajelez, hogy övé a keresett cím. Mivel az MAC és IP címek közti összerendelés viszonylag ritkán változik, gyakran használnak gyorsítótárat a már megtalált MAC-IP címpárok számára. E címtár neve ARP Cache. Az egyes bejegyzések csak egy bizonyos időtartamig érvényesek, ennek nagysága implementáció-függő. Általában pár perc, de például egyes Cisco útvonalválasztóknál két óra is lehet.

ICMP

Az *Internet Control Message Protocolt* (RFC 792) egyszerű, a hálózat működéséhez szükséges üzenetek küldésére használják. Egy IP csomag kap ICMP fejlécet; ez sérti ugyan a protokollhierarchiát, de funkciója miatt mégis a

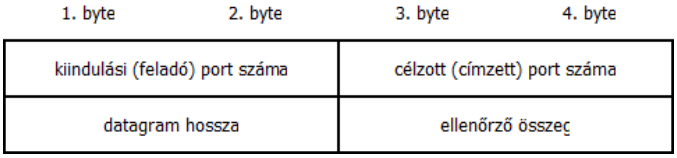
Address	HW type	HW address	Iface
akela.sch.bme.hu	ether	00:06:2B:00:DB:A3	eth0
router.sch.bme.hu	ether	00:E0:18:DD:26:58	eth0

4. ábra. ARP tábla

hálózati réteghez soroljuk. 255 különböző típusú ICMP üzenet létezik, legfontosabbak az echo üzenetek, ezekkel lehet ellenőrizni egy távoli állomás működési állapotát. Ilyen csomagokat használ a ping program is. (Emellett routerek is gyakran küldenek ICMP üzenetet, ha cél elérhetetlen, de más alkalmazások is léteznek.)

UDP protokoll

A User Datagram Protocol (RFC 768) a szállítási rétegbe tartozik. Összeköttetésmentes protokoll, rövid fejlécében csak a cél-, illetve forrásport szerepel, ennyi a többletinformáció az IP-hez képest. Így ez sem biztosíthat garanciát arra, hogy a csomag kézbesítésre kerül. Olyan alkalmazásoknál használják, ahol fontos a gyors átvitel és nem okoz problémát, ha egy-egy csomag elvész. Például: játékprogramok, hangátvitel stb.



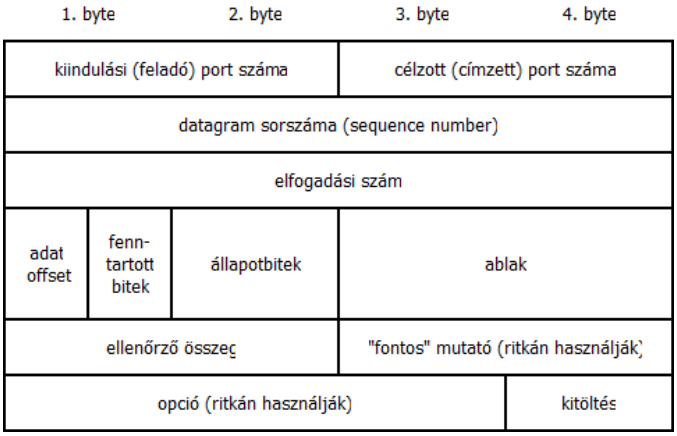
5. ábra. UDP fejléc (az IP fejléc felett)

TCP protokoll

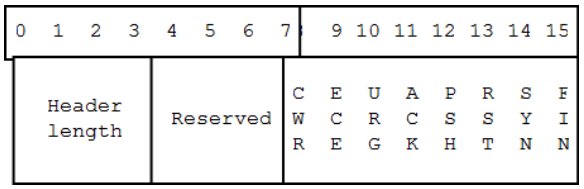
A Transmission Control Protocol (RFC 793) szintén a szállítási réteghez tartozik. Összeköttetés alapú protokoll, amely csatornát alakít ki a küldő és a fogadó gép között. A TCP-t használó alkalmazás általában sok adatcsomagot akar elküldeni a cél állomásra, és emellett azt is tudni akarja, hogy azok épségben megérkeztek. Ha egy csomag átvitel közben megsérül, azt újra elküldik. Gyakran választják ezt a protokollt, ha az üzenet nem fér bele egyetlen adatcsomagba, vagy ha a kliens és a szerver között egyfajta beszélgetésre van szükség.

A TCP protokoll alapváltozatán felül sok bővítés, kiegészítés is napvilágot látott, hogy a protokoll jobban alkalmazkodjon az Internet megváltozott struktúrájához. A kiegészítések szolgálják a jobb minőségi paramétereket

(QoS, torlódásmentesség), a nagyobb sebességek (kisebb overhead) elérését stb.



6. ábra. TCP fejléc



7. ábra. A TCP fejléc 13. és 14. byte-jának pontos felépítése

A fenntartott bitek közül többet az RFC3168-ban közölt Explicit Congestion Notification eljárás használ, így 13., 14. byte pontos felépítése a 7. ábrán látható. A legfontosabb bitek jelentése:

- ACK: az acknowledge bit az jelzi, hogy az „elfogadási szám” mező érvényes értéket tartalmaz.
- SYN: ez a bit jelzi, ha egy új kapcsolat felépítése kezdődik
- FIN: kapcsolat bontásnál használjuk

A TCP-ben minden csomagnak sorozatszáma van, ez alapján kerül később nyugtázásra. Ha a küldő fél a csomag elküldését követően adott időn belül (TCP timeout) nem kapja meg a nyugtát, újra elküldi a csomagot. A nyugtázás is a TCP fejlécben történik. Ha az állapot bitek közül az ACK 1-es, akkor az „elfogadási szám” (nyugta) mező tartalmazza a nyugtázott csomag

sorozatszámát. A nyugtázó csomag természetesen adatokat is tartalmazhat, ezt nevezik piggybackingnek.

Kapcsolatfelvételkor a kezdeményező gép olyan adatcsomagot küld a célgépnek, amelyben a SYN jelzőbit 1-es, a sorozatszám értéke pedig A. A célgép erre egy SYN=1, ACK=1 csomaggal válaszol, amelynek nyugta mezejébe elhelyezi a kapott A értéket, a sorszám mezőbe pedig új B értéket sorsol ki. Végül a kezdeményező küld egy ACK=1 csomagot, amelynek nyugta mezejében B szerepel. Így jön létre a kapcsolat. A kapcsolat bontása hasonlóképpen zajlik le, de itt nem a SYN, hanem a FIN bitet használják.

A TCP fejléc 24 bájt hosszú (6 db 4 bájt hosszú szó). A legfontosabb információk a küldő és a fogadó szolgáltatás egyedi azonosító száma (más néven: port száma vagy socket száma), és a datagram (azaz az adott adatcsomag) sorszáma. A port száma alapján dönthető el, hogy a kliensen melyik program küldte az üzenetet a szerver melyik szolgáltatásának. A sorozatszám pedig a nyugtázáson túl arra is szolgál, hogy a vevő megfelelő sorrendben rakhassa össze a datagrammokat. Előfordulhat ugyanis, hogy azok nem ugyanabban a sorrendben érkeznek meg, mint ahogy elküldték őket. Hiszen a köztes hálózati elemek, csomópontok a csomagokat eltérő útvonalon, összevissza sorrendben is továbbíthatják.

A fejléc ellenőrző összeg mezőt is tartalmaz. A vevő ez alapján döntheti el, hogy a csomag épségben érkezett-e meg. Ezen kívül a TCP fejléc több kiegészítő információt is tartalmaz, amelyek az üzenet tulajdonságaira, vagy a párbeszéd vezérlésére alkalmasak. A fejléc végén a kitöltés a felhasznált opcióktól függ, célja, hogy a fejléc 4 byte-tal osztható legyen.

TELNET

A TELNET protokoll egyszerű módszer távoli gépre való bejelentkezéshez. Sem titkosítást, sem integritásvédelmet nem biztosít, az így kialakított kapcsolat a TCP kapcsolat jellemzőivel bír. Általában jelszó alapú felhasználó azonosítás történik, ilyenkor a felhasználói név és a jelszó is védtelenül utazik a hálózaton. Ezek lehallgathatóak, de a kapcsolat akár aktívan is támadható (hijacking). Manapság az SSH protokoll különböző verzióit használják a TELNET helyett. Az SSH többek között a TELNET funkcióit is ellátja, de erős titkosítással. Tipikus esetben valamelyik nyilvános kulcsú algoritmus segítségével kulcscsere történik, majd az így kicserélt titkos kulcsok segítségével szimmetrikus kulcsú algoritmussal zajlik a kapcsolat titkosítása, hitelesítése. TELNET protokoll engedélyezése egy távolról is elérhető szerveren ma már biztonsági résnek minősül. A TELNETet ennek ellenére még ma is használják. Például routerek és switch-ek főként ennek segítségével menedzselhetőek, természetesen számos más biztonsági intézkedéssel megtámogatva. (Például, csak kitüntetett IP címekről és csak belső hálózatról, vagy közvetlen kapcsolatról használható ez a megoldás.)

HTTP

A Hypertext Transfer Protocolt hypertext dokumentumok gyors és hatékony átvitelére, megjelenítésére terveztek. A http a world wide web átviteli protokollja. Állapotmentes protokoll, vagyis az ügyfélprogram kérdéseit egymástól függetlenül kezeli, és minden dokumentum elküldése után lezárhatja a kapcsolatot. Ez az állapotmentesség fontos feltétel ahhoz, hogy a kiszolgáló mindenki számára egyformán gyors legyen.

A HTTP kapcsolat 4 lépésben jön létre:

1. A TCP kapcsolat megnyitása: Ekkor a kliensprogram meghívja a kiszolgálót az Interneten keresztül az adott IP cím és port azonosító alapján (alapértelmezésben a 80-as porton keresi a kiszolgálót).

2. Kérés elküldése: A kliensprogram üzenetet küld a kiszolgálónak, amelyben valamilyen szolgáltatást kér. A kérés HTTP fejlécből és a kiszolgálónak küldött adatokból áll. A fejléc információkat tartalmaz a kiszolgáló számára arról, hogy milyen típusú a kérés és a kliens programnak milyen lehetőségei vannak. Tipikus HTTP módszerek az információkérésre a GET és a POST. Tipikus kérés: `GET /index.html`

3. Válasz: A kiszolgáló a választ visszaküldi a kliensprogramnak. Ennek része a fejléc, mely leírja a válasz állapotát (sikeres, vagy sikertelen), a küldött adatok típusát. Ezután maguk az adatok következnek.

4. A kapcsolat lezárása: A kiszolgáló a válasz elküldése után lezárja a kapcsolatot, így a kiszolgáló erőforrásai rögtön felszabadulnak a következő kérések teljesítéséhez. (itt is van kivétel: HTTP keepalive, ezesetben a kapcsolat nem kerül azonnal lezárásra)

A HTTP kapcsolat során csak egyetlen dokumentum átadása történik meg, illetve csak egyetlen kérés-feldolgozás zajlik le. Az állapotmentesség miatt a kapcsolatok semmit nem tudnak egymásról, mert a szerver minden kérést külön-külön kezel. Ha egy dokumentum képeket is tartalmaz, akkor a kliens program annyiszor építi fel a kapcsolatot, ahány hivatkozást talál (egyet magának a dokumentumnak és minden képnek is egyet-egyet).

1.2. Berendezések

Hub

A hub egy többcsatlakozós jelismétlő (Multiport repeater). Ez az eszköz a fizikai rétegben helyezkedik el, több fizikai közegen futó hálózatot (vagy gépet) kapcsol össze. Az egyes részhálózatok a hub ún. portjaira kapcsolódnak. (A port gyakorlatilag az ajzat, ahova a kábelt kell csatlakoztatni.) Mivel a fizikai rétegben dolgozik, az adatkapcsolati rétegi címeket nem érti, így minden

bejövő adatot kiküld minden portra. Előnye, hogy olcsó, de ennek ellenére manapság csak otthoni hálózatokban használják. Nem menedzselhetők, a berendezés saját IP címmel, konfigurálható paraméterekkel, stb. nem rendelkezik. Sok hubból álló hálózatban nagyon nagy a nem célirányú forgalom, de elvi akadálya is van nagy, hubbal kapcsolt hálózatnak.

Kapcsoló (switch)

A switch több csatlakozással rendelkező híd (multiport bridge). Az adatkapcsolati rétegben dolgozik, feladata azonos a hubéval, de sokkal intelligensebb annál, ugyanis a kapott adatcsomagokat csakis arra a portjára továbbítja, amelyen a célállomás található. Ennek érdekében táblázatban tárolja, hogy mely célállomások (vagyis mely MAC-ek) mely porton keresztül érhetőek el. Természetesen az ún. broadcast adatcsomagokat minden csatlakozó felé továbbítja. Egy switch-nek számos egyéb funkciója is lehet, legfontosabbak a következők:

- VLAN, Virtual LAN-ok kezelése: Egy nagyobb switch-ben, vagy switchekből álló hálózatban szükség lehet bizonyos hálózati csoportok logikai szeparálására.
- STP, Spanning Tree Protocol: Az összekapcsolt switch-ekből feszítőfát épít, így garantálható a hurokkmentes üzem. Természetesen hurkok esetén hibatűrés is vihető a rendszerbe.
- Hozzáférés korlátozás: Definiálni lehet, hogy egy adott porton milyen ethernet hardver (MAC) címmel lehet csatlakozni.
- Menedzsment lehetőség: konfigurálás, monitorozás, naplózás, forgalom korlátozása, riasztások stb.
- Hálózat lehallgatásának támogatása: Switch esetében egy állomás a broadcast adatcsomagokon túl nem képes megfigyelni a többi állomás által egymás között forgalmazott adatforgalmat. Hibafelderítés stb. céljára definiálható, hogy egy csatlakozás nem neki szóló adatokat is megkapjon.

Útvonalválasztó (router)

A router olyan hálózati rétegi eszköz, amely több interfésszel is rendelkezik. Az feladata, hogy az egyes interfészein beérkező csomagokat a megfelelő interfészre továbbítsa. Ezen útvonalválasztó berendezések a konkrét útvonalválasztáson túl számos feladatot elláthatnak:

- Kódoló/dekódoló szerepe lehet virtuális magánhálózati technológiák alkalmazása esetén (pl. IPSEC)

- Biztonsági szűréseket végezhet az áthaladó adatfolyamokon, állapotmentes vagy állapottal rendelkező módon. A leggyakoribb esetben egyszerű portszűrőként viselkedik, ACL (Access Control List) táblázatok alapján.

- Authentikációs és authorizációs eszközként is funkcionálhat, illetve együttműködhet hasonló eszközökkel, hogy csak azonosított felhasználók érjenek el bizonyos hálózati elemeket.

A statikus útvonalválasztás elsősorban kis hálózatok esetében alkalmazható. A nagyobbak, bonyolultabbak esetében az útvonalválasztást sokkal kifinomultabb algoritmusok és őket támogató protokollok (pl. BGP, RIP stb.) végzik. Ezen intelligensebb megoldások mérhetik például a vonalak terheltségét, kapacitását, késleltetését, és ezt is figyelembe vehetik útvonalválasztáskor.

Gyakran alkalmazzák az útvonalválasztót a NAT (Network Address Translation) feladat elvégzésére is. Ezen belül például arra, hogy olyan belső állomás is hozzáférhessen a külső hálózathoz, amelynek nincsen az Internet felől elérhető IP címe. Ilyen esetekben a belső hálózaton olyan speciális (az RFC-ben szabályozott) belső IP tartományokat használnak (`192.168.X.X`, `172.16.X.X`, `172.17.X.X`, `10.X.X.X`), amelyeket az útvonalválasztók egyáltalán nem továbbítanak. A router ilyen esetben a következőt teszi a belső hálózatról érkező csomagokkal: a kliensről jövő kéréseket nem továbbítja, hanem új kérést generál, amelyben saját magát jelöli meg küldőként, és egy tetszőleges (általában magas, tipikusan 4000 feletti) portot választ ki forrásportnak. Eltárolja, hogy a választ melyik belső kliens melyik portjára kell küldeni, és ha az megérkezik, akkor továbbítja a kliensnek. Kívülről NAT-tal elrejtett állomásokat nem lehet közvetlenül elérni, így szerver szolgáltatásokat közvetlenül nem lehet rajtuk futatni. Ezt a technikát nevezik masqueradingnek.

Tűzfal (firewall)

A tűzfal biztonságtechnikai eszköz, amely több rétegben is működhet. A legtöbb tűzfal képes hálózati és szállítási rétegben is dolgozni. Gyakori továbbá az adatkapcsolati réteg támogatása, de a bonyolultabbak alkalmazás rétegben is működnek. A tűzfal feladata a forgalomszűrés. Általában két, vagy több hálózat határán kerül telepítésre, és ezek közt szabályozza a kommunikációt. Meghatározza, melyik hálózatból melyik hálózatba milyen feltételek mellett (pl.: milyen protokollokkal) lehet kapcsolódni. A legtöbb tűzfal emellett alkalmas például NAT funkciók ellátására is. A szűrést többnyire a szűrést leíró ún. tűzfal szabályokkal lehet végezni:

- az adatkapcsolati rétegben forrás és cél MAC cím alapján,

- a hálózati rétegben forrás és cél IP, valamint a bejövő és kimenő interfész alapján.
- a szállítási rétegben forrás és cél port alapján, valamint TCP flagek (jelzőbitek) alapján (például, ezekkel lehet megkülönböztetni a már meglévő kapcsolatokat és a kapcsolatfelvételi kezdeményezéseket)
- az alkalmazási rétegben protokoll-specifikus adatok alapján (A döntés ebben a rétegben gyakorta nem explicit lista alapján, hanem bonyolultabb, pl. heurisztikus algoritmusok alapján történik. Szűrhetünk például HTTP tartalomra, FTP forgalomra, de akár csúnya szavak használatára is beszélgetési szolgáltatások esetén stb.)

A továbbítás annál gyorsabb, minél alacsonyabb rétegben fut a tűzfal, mert alacsonyabb rétegben kevesebb időt kell a felsőbb rétegek kicsomagolására, illetve a csomagok tördelésére és összeállítására fordítani. Az alkalmazásszintű szűrés nehéz feladat, és szinte lehetetlen nagy forgalom (több optikai kábelen érkező adatok) esetén. Az alacsonyabb rétegekben való szűrés könnyen megvalósítható.

PC alapú hálózati eszköz

Kapcsoló, útvonalválasztó és tűzfal szerepét elvileg PC is el tudja látni, ez kis rendszerekben jó megoldás lehet. Ennek a megoldásnak azonban több hátránya van:

- sok hálózati csatolókártya szükséges (router esetén ez elfogadható),
- a gép könnyebben támadható, (nem célirányos operációs rendszert futtat, és ez gyakran tartalmaz biztonsági hibát)
- kapcsolási sebessége sokkal rosszabb egy céleszköznél.

Mindezek ellenére sok helyen alkalmaznak PC alapú routert vagy tűzfalat (switch-et csak 2-3 port esetén reális megvalósítani), azonban ezek csak kis forgalom esetén működnek. Nagy sebességű útvonalválasztásra gyakran nem elégséges egy PC. Az is gyakori probléma, hogy az adott hálózati közeg elérése PC-ről nem biztosított (pl. szinkron soros port, ATM vezérlőkártya, stb.). A biztonsági kérdésekre PC esetén különös figyelmet kell fordítani.

1.3. Támadások, módszerek

Figyelem, a most következő eljárások a legtöbb hálózati szabályzat szerint támadásnak minősülnek!

IP spoofing

Az IP spoofing IP cím hamisítását jelenti. Ilyenkor egy IP hálózaton kommunikáló berendezés támadó, megtévesztő céllal megváloztatja az IP címét, esetleg egy másik berendezés IP címét veszi fel. A legtöbb operációs rendszer alatt könnyen meg lehet változtatni az IP címet, sőt, lehetőség van egy interfészhez több IP cím kiosztására. Így elérhető például.

- Másik számítógép megszemélyesítése, és így jogosultságok szerzése IP alapján kontrollált erőforráshoz. Tipikus példák: adatbázis-hozzáférés megkaparintása, viszonyrablás (már meglévő TELNET kapcsolat eltérítése).
- Tűzfal szabályok kijátszása: Például, egy tűzfalnak külső oldalról küldünk olyan üzenetet, amelyben a forrás IP a belső címtartományból való. Egy rosszul beállított tűzfalat ez könnyen átejthet.
- Man-in-the-middle támadás felépítése. (Két kommunikáló fél megszemélyesítése egymás felé.)
- Távoli gép kommunikációjának zavarása. Két azonos IP című gép esetén előre nem látható hibák keletkezhetnek az átvitelben. Gyakori a kezelhetetlenül magas csomagvesztés, illetve a teljes forgalmazási képtelenség.
- Hálózati támadást indító forrás számítógép elrejtése.

Az IP spoofing egy lokális hálózaton belül könnyebben megvalósítható, míg az Interneten nem vehető át tetszőleges gép IP száma, hiszen így az útvonalválasztás nem feltétlenül továbbítja az adott gép felé az adatcsomagokat. Korábban előfordult, hogy a hamis feladóval elküldött IP adatcsomagok gond nélkül átmentek a hálózaton, de ma már sok helyen elvégzik a nem az adott hálózatba tartozó címekről érkező adatcsomagok szűrését (ingress filtering).

ARP spoofing, poisoning

Az ARP spoofing az IP spoofinghoz hasonló módszer, de itt MAC (ethernet hardver) címeket hamisít meg a támadó. Ennek a segítségével is hasonló támadások hajthatók végre, de eggyel alacsonyabb rétegben. Mivel adatkapcsolati rétegbeli címeket használ, csak olyan eszköz ellen hajthatók végre sikeresen az ilyen típusú támadások, amelyek egy hálózaton vannak a támadóval, hiszen adatkapcsolat szinten csak ezekkel lehet kommunikálni.

Egy hálózat alatt kicsit pontosabban egy collision domaint kell érteni, azaz azoknak a gépeknek a halmazát, amelyek együtt versenyeznek a csatornáért. Ez BNC Ethernet esetében az egy kábelen levő, illetve HUBbal összekapcsolt gépeket jelenti, UTP esetén pedig a HUB-bal összekapcsolt node-okat. Az ARP kérésekkel könnyen vissza lehet élni:

- A támadó válaszolhat tetszőleges ARP kérésre, így megakadályozhatja bizonyos hostokkal az összes kommunikációt.
- ARP poisoning: a támadó szándékosan félrevezeti a célpontot, például a saját MAC címét adja meg a célpont ARP kérésre, így végrehajthatja az IP spoofingnál ismertetett megszemélyesítéses támadást. A célpont ARP táblájában lévő hamis ARP bejegyzéseket nevezzük ARP poisionnak.
- Mivel a switch-ek figyelik, hogy melyik porton milyen MAC címmel forgalmazó hostok vannak, könnyen támadhatóak DOS támadással: sok véletlen MAC címről küld csomagokat a támadó, így a switch erre a célra elkülönített memóriája betelik, és ez lassú, vagy szakadozó működéshez vezethet.

Portátfésülés (portscan)

A portscan behatolás előkészítési, biztonsági ellenőrzési technika. A támadó a kiszemelt célszámítógép több portjához próbál kapcsolódni, így meg tudja állapítani, hogy a célgép mely portjai vannak nyitva, zárva, esetleg tűzfallal védve. A portok ellenőrzését általában TCP alatt végzik, mivel itt a kapcsolódás módszeréből adódik, hogy kideríthetőek a nyitott portok. Más protokollok, így például UDP esetén is vannak módszerek a nyitott portok megállapítására, de ezek sokkal kevésbé megbízhatóak. A portátfésülés menete a következő: a támadó egyesével SYN=1 csomagokat küld a célpont portjaira. Ha erre TCP válasz érkezik SYN=1, ACK=1-el, akkor a port nyitva van (az nmap szóhasználatában OPEN). Ha a válasz ACK=1, RST=1, akkor a port zárva van (CLOSED), ha pedig ICMP válasz érkezik, pl: host unreachable, akkor abból tűzfallal lezárt portra lehet következtetni (FILTERED). Amennyiben a kapcsolódási kérésre nem érkezik ACK=1, RST=1 válasz sem, úgy az adott kapcsolódási kérést valószínűleg szintén valamely közbülső elem szűri ki, tehát ebben az esetben is tűzfalra következtethetünk (az nmap szóhasználatában FILTERED). A portok lekérdezési sorrendjét a jobb portscannerek véletlenszerűen választják meg, illetve időbeli eltolást alkalmaznak, hogy megnehezítsék a portátfésülés észlelését, továbbá más technikákkal (jelzőbitek állítása) próbálják megnehezíteni a portátfésülés detekcióját. Ezt a módszert nevezik stealth scannek. A különböző operációs rendszerek detektálhatóak (OS detection) a TCP/IP implementációjuk alapján: a legtöbb rendszerben egymástól eltérő módon valósították meg a nem specifikált részleteket, ezek alapján szinte biztosan meg lehet mondani, hogy a cél milyen operációs rendszert futtat. Ilyen részletek például: a TOS mező (szolgáltatás típusa) értéke, szabálytalan csomagok kezelése, ICMP csomagok hossza, TTL értéke bizonyos esetekben, TCP timeout változása stb.

Sniffing

A sniffing a hálózat lehallgatását jelenti. Ilyenkor a támadó a hálózati interfészt (NIC – Network Interface Card) ún. promiscious, válogatás nélküli módba kapcsolja. Így minden ethernet csomagot elkap, nem csak azokat, amelyek kártyája hardver címéhez (MAC) tartoznak. A lehallgatás segítségével a helyi hálózat forgalmát szerezheti meg a támadó, így könnyen hozzájuthat minden a hálózatra nyíltan kiküldött (nem titkosított) adathoz. Mivel így is csak azt hallgathatja le, ami fizikailag is eljut a NIC-hez, ezért ezt a módszert csak akkor alkalmazhatja, ha a helyi hálózat minden csomagot elküld az adott gépnek (pl. hub esetén), vagy ha valamilyen ARP táblát illetve switch-et befolyásoló módszerrel ráveszi a köztes elemeket, hogy felé is továbbítsák az adatcsomagokat. Természetesen az ilyen ARP táblát befolyásoló aktív támadási módszerek esetében az is elképzelhető, hogy az eredeti címzett nem kapja meg az adatokat.

A promiscious módban futó NIC általában detektálható! Erre több módszer is létezik, leghatékonyabb az idő alapú és az ARP spoofing megoldás. Az előbbinél a detektor a helyi hálózaton levő számítógépek válasz sebességéből következtet a lehallgató (sniffer) jelenlétére, mert az ilyen módon működő számítógép a szokásosnál sokkal lassabban válaszol. A másik módszer azon alapul, hogy a lehallgató bármilyen MAC címre szóló csomagot feldolgoz. A detektor küld a feltételezett sniffernek egy olyan csomagot, amely az ő IP címére szól, de nem az ő MAC címére. A sniffer adatkapcsolati rétege fel fogja adni a csomagot a felsőbb rétegeknek, és ha az IP egyezőségét vizsgálja, de ethernet cím egyezőségét nem, válaszolni fog a kérésre. A detektor a válasz alapján ismeri fel a lehallgatókat. Természetesen a lehallgatás észrevétele ellen is alkalmazhatóak módszerek.

IDS

Az *Intrusion Detection System* behatolásérzékelő rendszer. Figyelhet egy célszámítógépet, vagy akár egy egész hálózatot (Network IDS, NIDS). Feladata a behatolási kísérletek észlelése, naplózása, és esetleges ellenintézkedések tétele. Mivel a behatolás fogalma, és felismerhetősége nem egyértelmű, sok lehet a téves riasztás. A hálózati alapú IDS-ek többsége lehallgatja a helyi hálózat forgalmát, és mintaillesztési módszerekkel keres támadásra jeleket. Ilyen lehet például biztonsági rés kihasználására használt gyanús HTTP kérések, portscan, sorozatos jelszópróbálgatás, nem szokott állomásról történő szolgáltatáshasználat stb. Az ellenintézkedés lehet azonnali tiltás a tűzfalon, levél a rendszergazdának, riasztás az érintett felhasználónak, stb. Az ellentámadás nem szerepel a lehetőségek között, de gyakorta alkalmazzák a támadóról való információgyűjtés végett. Ennek jogi megítélése (személyiségi jogok, adatvédelem, stb.) bonyolult kérdés.

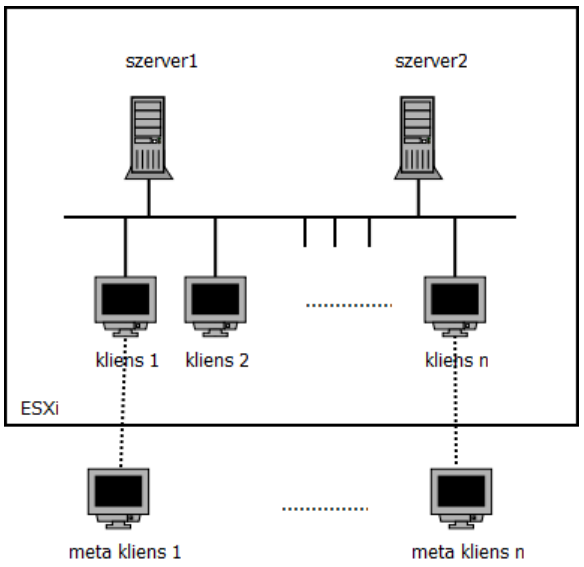
Figyelem! A helyi hálózatban átvitt adatok egy részét még akkor sem

biztos, hogy jogunk van megismerni, ha saját cégünket szereljük fel különböző célszoftverekkel. A hálózaton áthaladó e-maileket például a levéltitok védelme illeti meg.

2. Mérési eszközök

A mérés egyszerű és rugalmas lebonyolítása érdekében a 8. ábrán látható mérési elrendezést használjuk. A mérést kizárólag a helyi hálózatban végezzük, Internetkapcsolat nincs.

A mérésben CD-ről indítható Linux alapú számítógépek vesznek részt. A számítógépeket négy csoportba osztjuk:



8. ábra. Mérési elrendezés

1. `Szerver1`, amely a következő funkciókat látja el:
   - DHCP szerver, ez a számítógép osztja ki a mérést végző többi számítógép (kliensek) számára az IP címeket
   - Minta web kiszolgáló (Apache szerver PHP kiegészítéssel)
   - Folyamatos hálózati forgalmat generál a vizsgálatok céljára
2. `Szerver2`
   - A berendezésen IDS szoftver (Snort fut), eredményét webről lehet elérni

- Bizonyos folyamatos hálózati forgalmat generál a vizsgálatok céljára

3. Kliens számítógépek (ezek a mérőállomások)
   - Alkalmasak X-Windows futtatására megfelelő hardver feltételek esetén
   - Telepítve van rajtuk számos, a méréshez szükséges alkalmazás
4. Meta kliensek: ezek az ESXi környezet elérését teszik lehetővé

A mérés során alkalmazott ún. Sniffix operációs rendszer a GNU Debian Linux alapú Knoppix operációs rendszer egy speciális módosítása. A CD-ről való futtathatóság érdekében a hagyományos Linux fájlrendszert meg kellett változtatni, hiszen szoftverek a CD-re nem írhatnak. Ezen okok miatt:

- A gyökérkönyvtár (`/`) egy memórialemez (ún. initrd), mintegy 3 MB nagyságú lefoglalt memóriaterület, ahova írni és olvasni lehet, ám a terület nagy része már a rendszer által használt a bekapcsolás időpontjában is
- A `/ramdrive` alkönyvtár a memóriából lefoglalt terület, ez a felhasználók által írható-olvasható, a mérés során használt ideiglenes fájlokat ide kell írni.
- A `/cdrom` alkönytáron látható a CD nyers tartalma
- A `/KNOPPIX` alkönyvtárban találhatóak a CD-n elhelyezett tömörített fájlok

A további alkönyvtárak általában a CD tömörített, csak olvasható alkönyvtáraira mutatnak.

A rendszer felhasználói:

Rendszergazda: `root`, jelszava: `proba`

Próba felhasználó: `crysys`, jelszava: `proba`

A használt IP hálózat felépítése

A kialakított minta hálózat a `10.105.2.0`-tól `10.105.2.255`-ig terjedő alhálózatot használja (azaz `10.105.2.0` hálózat `255.255.255.0` hálózati maszkkal) A `szerver1` gép a `10.105.2.254`-es belső hálózati címen helyezkedik el. A `szerver2` fix ip száma `10.105.2.253`. A `szerver1`-en futó DHCP szerver osztja ki a kliens számítógépek IP számait, ezek a `10.105.2.10-30` tartományba esnek. Az ARP támadásos tesztek miatt a `szerver2` gép további IP számokon is megtalálható, ezek: `10.105.2.210-225`.

A mérés kezdete:

A számítógép bekapcsolása után a mérést végzők a meta kliensek segítségével VNC protokoll használatával becsatlakoznak a kliensekre a mérésvezető által megadott IP címre és portra. Ezután az egész mérést a VNC kliensen keresztül elért Backtrack Live disztibúcióban végzik.

Egyes programoknak, így pl. a hálózati lehallgatást végző programoknak rendszergazdai jogok szükségesek, ezért nem célszerű a rendszergazdai jogok helyett más felhasználó nevében bejelentkezni.

X-Windows használata esetén a szöveges módba a `crtl-alt-f1` gombbal lehet váltani, vissza az `alt-f5`-tel. A linux virtuális szöveges terminálok közötti váltás az `alt-f1` – `alt-f4` gombokkal lehetséges. A kliens gépen SSH kiszolgáló fut, így elméleti lehetőség van más mérési gépek elérésére, de ehhez engedélyt kell kérni a mérésvezetőtől.

A gépeken számos, a munkát segítő program került elhelyezésre, ezek a mérések során használhatóak. Rövid felsorolás:

- A mérés elvégzéséhez szükséges programok: tcpdump, tshark, dsniff, nc, telnet (A programok használatát segítő tanácsok a 4. fejezetben olvashatóak.)
- Programozási nyelvek: perl, C, tcl, ...
- Editorok: emacs, vi, joe, nano, ...
- Szűrők és átalakítók: more, less, grep, sed, awk, ...
- Szöveges web böngésző: links, lynx
- Web letöltő program: wget
- Fájl menedzser: midnight commander (mc)

A mérés helyi hálózata nincs csatlakoztatva az internet felé. A szöveges módban billentyűzetbeállítást váltani a `loadkeys us`, `loadkeys hu stb`. parancsokkal lehet.

3. Feladatok

3.1. Forgalom lehallgatása

Hallgassa le a hálózati teljes forgalmat a tcpdump és a tshark programmal! Nézze meg a kijelzés alapvető különbségeit az alapértelmezett kimeneteken!

3.2. Alapvető szűrés

Fogalmazzon meg néhány egyszerű szűrőszabályt és vizsgálja meg működésüket mindkét programban. (pl. ARP üzenetek szűrése, ICMP szűrése, esetleg bizonyos TCP port szűrése)

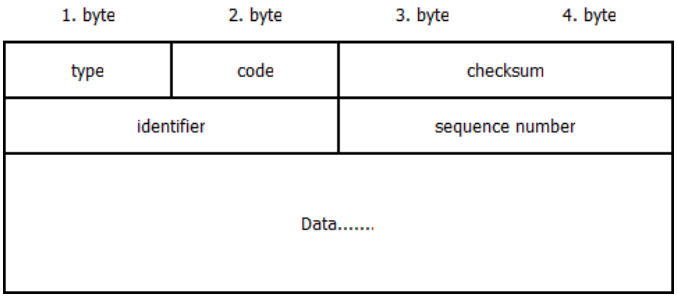
3.3. Adattartalom vizsgálata

Mindkét programmal írassa ki az adatcsomagok tartalmát is a fejléc tartalmán túl. Figyelje meg a kijelzés különbségeit.

A hátralévő feladatok tetszőlegesen megoldhatók tcpdump, tshark vagy wireshark program segítségével (ha nincs másként jelezve).

3.4. ICMP fejléc vizsgálata

Az ICMP fejléc adattartalma ICMP echo és echo reply üzenetek esetén a 9. ábrán látható. A "type" mező értéke 0 echo reply (ping válaszcsomag) esetén, míg 8 echo request (ping) csomag esetén.



9. ábra. ICMP fejléc echo és echo reply üzenetek esetén

Hallgasson le a hálózaton két különböző irányú ping adatcsomagot (ICMP echo request és reply csomagokat) és azonosítsa az ICMP fejléc típus értékét. Nézze meg, hogy használja-e a rendszer az "identifier" mező értékét a csomag hexadecimális formátumának elemzésével.

3.5. TCP fejléc vizsgálata

Hallgasson le valamilyen TCP adatfolyamot, és mentse le a teljes adatcsomagot hexadecimális formában! Keresse ki a TCP fejléc egyes mezőit! Mi a feladó port száma? Számolja át decimális értékre a sorszám mező tartalmát!

Azonosítsa a csomagban a feladó IP számának helyét és nézze meg, hogy a rendszer milyen formátumban tárolja a 32 bites IP számot (kisebb-nagyobb helyiértékű részek).

3.6. ECN bit vizsgálata

Az RFC 3168-ban bevezett új funkcionalitás a TCP esetében az explicit congestion negotiation, mely segít a hálózati torlódások okozta gondok elhárításában. Sajnos, egyes korábbi implementációk, pl. tűzfalak az ECN használatát jelző TCP fejlécbittel (ECE) ellátott adatcsomagokat tévesen

hibásnak nyilvánítják és eldobják. A szerver gépek egyikén és a kliens gépen az ECN használata engedélyezve van. Derítse ki melyik számítógépen van az ECN engedélyezve. A következő módon járjon el:

Csatlakozzon a telnet program segítségével a szerverek ssh (22-es) TCP portjára. A csatlakozási folyamatot a tcpdump (vagy tshark) programmal hallgassa le! A SYN-t tartalmazó, illetve az erre válaszul érkezo ACK adatcsomag fejlécben azonosítsa az ECE jelzőbitet tartalmazó byte-ot, majd abban értékelje ki a bit értékét!

Ismételje meg a kisérletet úgy, hogy az ip fejléc megfelelő bitjére elhelyezett szűrőszabállyal kiszűri a bekapcsolt ECE bittel rendelkező adatcsomagokat!

3.7. MSS, MTU meghatározás

Az MTU a hálózati csatoló IP rétegének azon értéke, amelyet egy IP adatcsomag nem haladhat túl. Hagyományos ethernet hálózaton az érték átlalában 1500 byte. (lásd ifconfig parancs). A TCP a kapcsolat felépítése során az MTU értéke alapján megállapítja, hogy a TCP fejléc nélkül mekkora lehet a csomagonként átvihető maximális hasznos tartalom (payload) mennyisége (=MTU-40 byte). A kapott érték a TCP maximal segment size, azaz MSS. Az MSS értéket a TCP a másik fél által küldött MSS adat alapján teszi meg. Amennyiben a két gép közti internetes hálózaton valahol nem tudnak átmenni ezek az adatcsomagok (kisebb az adott alhálózat MTU értéke, pl. ADSL esetén 1496 byte), úgy a kapcsolatfelvétel „rövid" adatcsomagjai után a kapcsolat hosszú várakozás után megszakad az első nagy adatcsomag küldésekor. (hiszen azt a címzett nem kapja meg, mert a köztes elemek a túlméretes adatcsomagot kidobják).

Az MTU egyeztetés számos helyen okozhat problémát egy internetes rendszer létrehozásakor. Amennyiben a fenti példa szerint egy köztes hálózat alacsonyabb MTU értéket használ, úgy célszerű a rajta keresztül átmenő összes TCP kapcsolatfelvételi kérés fejlécében módosítani az MSS értékét, így a célszámítógépek helyesen egy alacsonyabb MSS értékkel dolgozhatnak.

Vizsgálja meg az MSS beállítását TCP kapcsolatfelvétel esetén! Küldjön át bármilyen nagyobb adatot az webszervernek (`10.105.2.254` című hoszt, `80`-as port) a nc program segítségével és lehallgatással vizsgálja meg, hogy hogyan alakul ki az MSS értéke, illetve mekkorák az átvitt nagyobb adatcsomagok! Nézze meg, mekkora a szerver MTU értéke (számolja ki a kapott MSS érték alapján)! Csökkentse a helyi számítógépe MTU értékét az ifconfig parancs segítségével, majd végezze el újra a lehallgatásos kísérletet!

3.8. Telnet kapcsolat megfigyelése

Lépjen be a telnet program segítségével a szerverre (név: `crysys` jelszó: `proba`). Vizsgálja meg a belépés folyamatát lehallgatással, a tcpdump prog-

rammal az adatok élő, ascii kijelzésével. Nézze meg, hogy a billentyűleütések milyen adatforgalmat generálnak!

3.9. Jelszó rögzítése

Rögzítse a telnet prokollon átmenő jelszavakat a dsniff program segítségével. A rendszer működőképességét vizsgálja meg úgy, hogy saját maga próbál belépni a szerverre a telnet program segítségével (megjegyzés: a dsniff program csak a lezárt telnet sessiont írja ki a logba). A `szerver2` időnként (2-3 percenként) felhasználót emulálva telnet segítségével belép a szerver1-re. A dsniff szoftver lehallgatásos módszereivel vizsgálja meg milyen névvel és jelszóval lép be!

3.10. IP Spoofing

A mérési környezetben levő DHCP szerver a kliens számítógépeknek `10.105.2.10`-től kezdve osztja ki az IP számokat. Állapítsa meg saját IP számát az ifconfig parancs segítségével!

A `szerver2` berendezést úgy állítottuk be, hogy számos IP számon egyszerre megtalálható legyen, így a `10.105.2.200-` kezdődő címeken. Saját IP számának utolsó jegyéhez 190-et adva válasszon ki egy egyedi számot ebből a tartományból! Az `ifconfig eth0:0 <cim> netmask 255.255.255.0` parancs segítségével adja hozzá saját IP címeihez a kijelölt IP számot. A helyi hálózatben ezután ugyanazon az IP számon két gép fog működni, az Öné és a `szerver2` berendezés. A `szerver1` berendezés folyamat ping (ICMP echo request) kéréseket intéz ezekhez a címekhez. Vizsgálja meg, hogy a kérések a változtatás után az Ön gépéhez, vagy a `szerver2`-höz érkeznek meg. Vizsgálja meg, ha a lehallgatásokat elvégezheti a két szerveren is. Ssh program segítésgével tud belépni a szerverekre (pl. `ssh -l crysys 10.105.2.254` ; jelszó: `proba`). A lehallgatásos méréshez a szerveren a sudo parancs segítségét használja, ennek segítségével tud a program rendszergazdai jogokkal futni. (pl.: `sudo tcpdump -n -e -i eth0 icmp`, jelszó: `proba`)

3.11. ARP táblák vizsgálata

Az előző mérés kiegészítéseként vizsgálja meg, hogy a különböző berendezések, főként a `szerver1` berendezés ARP táblája milyen bejegyzést tartalmaz az Ön által meghamisított IP cím viszonylatában (parancs: `arp -avn`) Miért nincs bejegyzés ehhez az IP címhez bizonyos gépeken?

3.12. CGI paraméter átadás vizsgálata

A `szerver2` számítógép időnként (percenként többször is) letölti a `szerver1`-en tárolt titkos alkalmazást, és annak CGI paraméterként átad két változóban egy felhasználói azonosítót és egy jelszót. A jelszó helyes megadása

esetén a `szerver1` egy titkos weboldalt mutat, egy titkos kóddal a `szerver2`-nek, azonosítás nélkül hibaüzenetet ad. A hálózat lehallgatásával derítse fel, hogy mi az alkalmazás pontos URL címe. Vizsgálja meg, mit ad vissza a szerver azonosítás nélkül. A lehallgatott adatokat alapján próbálja meg a CGI GET metódust használva átadni az azonosítási paramétereket a szervernek (pl. `http://szerver/cgiprogram.php?valtozo1=ertek1&valtozo2=ertek2&`), és érje el, hogy Ön is megkapja a titkos kódot (adatot) a szervertől!

3.13. Web kérés generálása kézzel

Az 1. szervert úgy konfiguráltuk, hogy a `http://10.105.2.254/` címről (megtekintése pl. `lynx` programmal) elérhető egy rövid html lap, amely egy php-ban írt kiszolgáló programon keresztül webes, e-mailben közvetített visszajelzésre teszi alkalmassá az oldalt néző felhasználókat.

Próbálja ki a levélküldőt egy próbaüzenettel! Az üzeneteket a szerver előre definiált felhasználója, a `webapp` felhasználó kapja meg. A `webapp` felhasználó mailboxát a vizsgálat célából elérheti a `http://10.105.2.254/mail` címen, nézze meg, rendben megérkezett-e üzenete!

Mentse le valamelyik lehallgató szoftverrel (pl. tcpdump) a gépéről a szerver felé átvitt adatokat, vizsgálja meg a benne átadott paramétereket!

A PHP program, amely a levél küldését végzi, a paramétereket a kliens számítógéptől kapja meg. Az adat egy része a levélküldő html lapon (látszólag) módosíthatatlan „rejtett” paraméterként került elhelyezésre, ezeket adja tovább a webböngésző. A tervező rossz munkát végzett. A célszemély e-mail címe a rendszerben nem konstans, hanem a html-ben átadott paraméter. Ennek megfelelően egy támadó felhasználó kis ügyességgel a levél küldő alkalmazást harmadik személyeknek történő levélküldésre (pl. illegális reklámra) használhatja. A lehallgatott párbeszédben azonosítsa a címzett mező értékét, és módosítsa azt a `fakerec` névre. Figyeljen oda a HTTP fejléc Content-length paraméterének pontos tartására is. Az így módosított kérést a nc program segítségével küldje el a webszervernek! A `fakerec` felhasználó mailboxának megtekintésével győződjön meg arról, hogy sikeres támadást hajtott-e végre!

3.14. Portscanner (portletapogató) használata

Vizsgálja meg a mérési környezet számítógépeit (első sorban a két szervert), hogy mely nyitott TCP portok találhatóak rajta! Vizsgálja meg a portscanner operációs rendszer azonosító képességét is. Méréseit a következő módon végezze:

A hálózati nyitott portok vizsgálatára az nmap nevű közismert programot tudja használni.

Az nmap egyszerűbb paraméterei:

`nmap [vizsgálat típusa] [opciók] célok`
A vizsgálat főbb típusai:

`-sT:` szabályos TCP kapcsolódási kisérlet

`-sS:` Syn (félbemaradt) kapcsolódási kisérlet (nem építi fel a teljes kapcsolatot, azaz a kapott válaszra azonnal RST-t küld, számos számítógép az ilyen kisérleteket nem naplózza

Opciók

`-P0:` ne végezzen ping (icmp echo) vizsgálatot a portfelderítés előtt. Amennyiben a célgépen az ICMP le van tiltva, úgy ezt az opciót kell alkalmazni

`-p <port, vagy port régió (pl. 5-1000,1125)>:` csak adott portokat vizsgáljon

`-O:` operációs rendszer felderítése az IP ujjlenyomat alapján (szükséges hozzá legalább egy nyitott port felderítése is)

Vizsgálatait végezze el különböző opciók használatával (pl. `-p` port kézi meghatározása az alapértelmezésen felül).

3.15. Portscanner lehallgatásos vizsgálata

Az előző mérésben alkalmazott portlehallgatást ismételje meg úgy, hogy a letapogatás folyamatát valamelyik lehallgató szoftverrel is ellenőrzi. Próbálja megfigyelni, hogy a szoftver a portokat milyen sorrendben, milyen gyorsasággal nézi át!

3.16. 2.16. IDS rendszer log elemzés

A bevezetőben bemutatásra került IDS rendszereknek számos kereskedelmi változatai mellett ingyenes implementációk is elérhetők. Jelen mérés során az ingyenesen elérhető snort hálózati IDS rendszert futtatjuk a `szerver1` gépen. A futás során a hallgatók által végzett vizsgálatokat az IDS rendszer értékeli, és azokról naplófájlban bejegyzéseket helyez el. Vizsgálja meg (nézze át), hogy az előre telepített IDS rendszer milyen bejegyzéseket végzett a mérés, alatt!

Az IDS rendszer logjai a `http://10.105.2.254/snortlog/` címen érhetőek el.

4. További információk

A mérés során felhasznált programokról részletesebb angol nyelvű leírást a `man <program neve>` paranccsal kaphatunk a mérés kliensgépein.

4.1. Az IFCONFIG program kezelése

Az ifconfig parancs segítségével lehet beállítani operációs rendszerünk IP konfigurációját. paraméter nélkül az összes interfész konfigurációját listázza ki a program, az interfész nevével paraméterezve a konkrét interfész beállításait.

Példa:

```
# ifconfig eth0
eth0 Link encap:Ethernet HWaddr 00:E0:29:32:D1:26
inet addr:152.66.249.139 Bcast:152.66.249.159 Mask:255.255.255.224
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:1343458986 errors:0 dropped:0 overruns:0 frame:3
TX packets:2153108572 errors:0 dropped:0 overruns:0 carrier:0
collisions:0
RX bytes:482769613 (460.4 MiB) TX bytes:2710438236 (2.5 GiB)
```

Fontos, hogy csak azt az interfészt használja a rendszer, amely státusza "UP" (lásd fent) a kártya bekapcsolása az `ifconfig eth0 up` kikapcsolása az `ifconfig eth0 down` paranccsal történhet. Lehallgatás alatt a kártya státuszsorában a "PROMISC" felirat is szerepel, jelezve, hogy ilyenkor a kártya minden adatcsomagot feldolgoz, nem csak azokat, melyek az ő hardver címére érkeztek. A kártya hardver címét a "HWaddr" sorból tudhatjuk meg. Fontos paraméter még az MTU értéke, ez a Maximum Transfer Unit, az adott csatolón átmenő legnagyobb IP Csomag mérete (a legnagyobb ethernet csomag mérete az ethernet fejléc hozzáadásával számítható ki ezalapján). További példák:

IP cím átírása:

```
ifconfig eth0 10.105.1.222 netmask 255.255.255.0
```

MTU átírása:

```
ifconfig eth0 mtu 1200
```

Interfész letiltása:

```
ifconfig eth0 down
```

4.2. A TCPDUMP program használata

A program a következő paramétereket fogadja el:

```
tcpdump [ -adeflnNOpqRStvxX ] [ -c count ] [ -F file ]
[ -i interface ] [ -m module ] [ -r file ]
[ -s snaplen ] [ -T type ] [ -w file ]
[ -E algo:secret ] [ expression ]
```

A tcpdump parancs alapvetően a hálózaton átmenő adatcsomagok fejlécének vizsgálatát célozza meg. A protokollokat különböző mértékben analizálja, egyes esetekben a tartalomról, magasabb szintű protokoll üzenetekről is tájékoztatást tud adni (pl. pppoe jelszó átvitel, samba üzenetek), de a támogatott protokollok listája viszonylag szűk. A tcpdump program egyszerű, gyors, könnyen kezelhető, ezért az alapvető hálózati analízisre alkalmas, mé-

lyebb elemzésre célmegoldások (pl. jelszó lehallgató programok) illetve több protokollt ismerő megoldások (pl. wireshark, tshark) használhatók.

A tcpdump használatakor mindenképpen javasolt kézzel megadni a lehallgatni kívánt interfészt (pl. `tcpdump -i eth0`), így biztosítható, hogy tévesen nem másik interfészt kívánunk használni.

Fontos opciók:

`-n:` Ekkor a tcpdump nem fejti vissza a dns neveket, csak az ip címeket írja ki. Célszerű használni, hisz a program így gyorsabb és az állomások könnyeben azonosíthatóak

`-e:` Részletes kiírás, pl. a fogadó és küldő állomás hardver címének kiírása. Fontos lehet hálózati hibák felderítésekor, hiszen segítségével felderíthetőek az ARP tábla különböző hibái (pl. ARP csalás, régi ARP bejegyzés beragadása, stb.) továbbá módot ad arra, hogy a csomag valódi forrását felderítsük. Példa: ha címfordítást alkalmazunk és a helyi hálózatban furcsa csomagot veszünk észre, akkor a hardver címet összehasonlítva az ismert állomások címeivel (pl. helyi arp tábla kilistázása az `arp -avn` parancs segítségével) megadhatja, hogy mely eszköz generálta a csomagot.

A helyi hálózatokon gyakori, hogy nagy a forgalom, ezt szűrni különféle szűrőszabályok (fent: expression) definiálásával tudjuk.

A leggyakoribb kifejezések:
src: forrás ip címe
dst: cél ip címe
src port: forrás port száma (udp/tcp)
dst port: célgép port száma
arp: csak az arp kérések
ip: csak az ip kérések
icmp: csak icmp kérések
tcp: csak tcp kérések
udp: csak udp kérések

Az egyes szabályok között logikai kapcsolatok alakíthatóak ki (or, not, and, ...)

A szűrőszabályok teljes listája a tcpdump program kézikönyvében [man tcpdump] elérhetőek. Minták szűrőszabályokra:

- `tcpdump -i eth0 icmp or scr port 22 or dst port 22` – ssh folyamok és icmp üzenetek kiírása
- `tcpdump -i eth0 src or dst 10.105.2.254 and src port 80 or dst port 80` – web-es lekérdezések az adott állomás és más állomások között

- `tcpdump -i eth0 -n -e arp or icmp` – arp és icmp üzenetek kijelzése, hardver cím megjelenítésével, alkalmazható ping-gel kombinálva hibás hálózati beállítások felderítésére

A tcpdump alkalmas a csomag tartalmának kijelzésére is, erre az `-x` (hexa kijelzés) és az `-X` (hexa és ascii) kijelzés opciók alkalmasak. A kijelzés során az ethernet keret fejléce nem kerül kiiratásra, így célszerű a `-e` opcióval együtt használni a hardver címek megörzése érdekében. (pl.: `tcpdump -n -e -x -i eth0 src or dst 10.105.1.254 and src port 25 >logfajl` – a 25-ös portról érkező adatok rögzítése a "logfajl" fájlba)

4.3. A TSHARK program használata

A wireshark és tshark program ugyanazt a funkcionalitást éri el, csak míg az wireshark egy X-windows alatt működő grafikus felület, addig a tshark szöveges módban is használható. A tshark funkcionalitásában megvalósítja mindazt, amit a tcpdump, csak több protokollt ismerve, szélesebb paraméterezhetőséggel, továbbá más szintaktikával.

Fontosabb paraméterek:

`-f <filter szabály>:` míg a tcpdump esetén nem kell jelölni, a tshark esetén -f kapcsoló után lehet magdni szűrőszabályokat

`-i <interface>:` vizsgált interfész megadása

`-x:` az adatcsomag tartalmának kiiratása a fejlécen felül, hexadecimálisan és ascii sztringként

A tshark a fogadott adatokat le is tudja menteni, illetve fájlból is működik. Ily módon a feldolgozás két ciklusra is bontható: Az elsőben a forgalmat lementjük, utána végezzük a pontos feldolgozást, a szűrőszabályok meghatározását. Nehezen reprodukálható hálózati események esetén nagyon hasznos ez a funkcionalitás. Az ehhez kapcsolódó opciók:

`-w <fájl>:` elfogott adatok mentése fájlba

`-w -:` elfogott csomagok stdout-ra (képernyőre) iratása

`-r <fájl>:` lementett adatok visszaolvasása a fájlból

A tshark program esetén részletesen tudjuk szabályozni azt, hogy az adott adatcsomagok hogy kerüljenek feldolgozásra, illetve milyen adattartalom legyen róluk kijelezve a képernyőn. A tshark a feldolgozás eredményeiről széleskörű statisztikát tud nyújtani, és számos más kényelmi funkcióval is rendelkezik, ezeket nem soroljuk fel, a `man tshark` parancs segítségével megtekinthető a program angol nyelvű részletesebb leírása.

A tshark szűrőszabályai hasonlatosak a tcpdumphoz:

- `host <ip>`: (ip-től és ip-hez menő adatcsomagok)
- `src host <ip>`: forrás ip meghatározott
- `dst host <ip>`
- `net 192.168`: a `192.168.bármi.bármi` IP címek
- `ether proto \arp` vagy simán `arp`: arp kérések
- `ip proto \tcp` vagy egyszerűen `tcp`: tcp kérések
- `port <port>`: a forrás vagy a cél port a megadott
- src port <port>
- dst port <port>
- `ip[8]`: az IP header 8. byte-ja
- `tcp[0:2]`: a TCP header első két byte-ja
- `ip[0] & 0x0f`: az IP header első byte-jának alsó bitjei

Példák szűrőszabályokra:

- `host 10.10.10.10 && ! net 192.168`: az egyik fél a `10.10.10.10`, és egyik fél sem tartozik a `192.168.X.X` címtartományhoz
- `tshark -i eth2 -f 'ether[0] & 0xF0' -w file1.pcap`: az eth2 interfész adataiból mentsük azokat, ahol az ethernet fejléc 1. byte-jának felső bitjei 0 értékűek a `file1.pcap` fájlba
- `tshark -f "icmp[0] != 8 and icmp[0] != 0"`: minden icmp adatcsomag megjelenítése, amely nem icmp echo request vagy echo reply, azaz nem ping adatcsomag.
- `tshark -i eth0 -f "host 10.105.2.254 && ip proto \icmp"`: az eth0 interfész adatainak kiejlzése, ahol icmp adatcsomagok mentek vagy jöttek a megjelölt gép viszonylatában.

4.4. A DSNIFF program használata

```
dsniff [-c][-d][-m][-n][-i interface][-s snaplen][-f services]
          [-t trigger[,...]]]  [-r|-w savefile] [expression]
```

A dsniff program egyértelműen a hálózaton átmenő jelszavak elfogására lett elkészítve. Használata nagyon egyszerű és számos szolgáltatás jelszavát képes lehallgatni. A paraméterezést szinte ki sem kell használni, máris könnyen megállapíthatóak a hálózati jelszavak.

Fontos paraméterei:

`-s snaplen` : ez a paraméter itt, és az előző programoknál (tcpdump, tshark) is azt szabályozza, hogy a protokollok feldolgozása során az adatcsomagok mekkora részét kell feldolgozni a programnak. A dsniff esetén alapértelmezésben ez 1024 byte. Hosszabb www post metódusok esetén érdemes növelni

`-w savefile`**:** az elkapott jelszavakat a rendszer egy bináris fájlba mentheti, a `-r savefile` opcióval lehet innen visszaolvasni azokat. egyszerűsíti a mentést

`-i` **interface:** interfész megadása

`expression`**:** tcpdump formátumú szűrőszabály

4.5. A TELNET program használata

A telnet távoli hozzáférést tesz lehetővé, a kapcsolatot – a közönséges telnet program – nem rejtjelezi. A telnet alklamas a telnet kiszolgálóhoz (telnetd) való kapcsolódásra, de segítségével az is megoldható, hogy más, TCP alapú protokollt használó kiszolgálót érjünk el távolról. Főbb paraméterei:

`telnet <hoszt> <port>`

Példa (kapcsolódás webszerverhez a 80-as porton):

`telnet 10.105.2.254 80`

A telnet kapcsolódás után a crtl-] billentyűkombinációval lehet parancsmódba lépni, ahol a kapcsolatot a "`quit`" parancs segítségével lehet megszakítani.

A telnet program nem támogatja a szinkronizációt más program outputjával (pl. a `cat fajl|telnet host 25` parancs segítségével e-mailt küldeni igen bizonytalan. (a kapcsolat megszűnik az EOF érkezésekor, a feldolgozás vége előtt) Erre a célra célszerű a NC, azaz netcat program használata.

4.6. A NC (netcat) program használata

A program két módban használható: Egyrészt kapcsolatfelvételre, mint a telnet program, másrészt TCP (ill. UDP) kapcsolatok fogadására is. Kapcsolatfelvételre (kliens módra) a program a telnethez hasonlatosan paraméterezhető, továbbá támogatja a "csővezetéken" át kapott információk feldolgozását is, tehát pl. a `cat fájl|nc host 25` parancs segítségével akár leveleket is küldhetünk.

Kapcsolatok fogadása (szerver) módban az nc csatlakozik valamely tcp portra, és a bejövő kéréseket fogadja, illetve akár más hoszt felé továbbíthatja is (egyszerű relay). (`nc -l -p port [host] [port]`) Pl. egyszerű adatfogadás, az adatok képernyőre írásával: `nc -l -p 1055`. Egy másik gépről ilyenkor az előző gép 1055-ös portjára telnetelve (vagy nc-vel kapcsolatot létesítve) egyszerű chat-programot kapunk, természetesen mindenféle authentikáció nélkül.

4.7. Fontosabb TCP/UDP portok

TCP 80: http (web)
- TCP 25: smtp (e-mail küldés)
- TCP 110: pop3 (e-mail letöltés)
- TCP és UDP 53: dns
- TCP 22: ssh
- TCP 23: telnet
- TCP 21: ftp
- TCP 20: ftp adat
- TCP 143: imap (levelezés)