Általános tudnivalók a méréssel kapcsolatban

A mérés feladatai bármikor végezhetők, akár a labor időpontján kívül is, a labor időpontjában emellett konzultációra van lehetőség. (Kérdések feltehetők e-mailben és szükség esetén megbeszélhetőek a labor időpontján kívül is.)

A mérés elvégzéséhez szükség lesz a Wiresharkra (vagy valamilyen más hálózatanalizátor programra), valamint a *4. Passzív forgalmi mérés* feladathoz egy .pcap fájlra, melyet a mérés weblapján található linkről lehet letölteni.

Az opcionális feladatokat (*3. VoIP forgalom szűrése és elemzése* feladatot és a *4.2 Passzív forgalmi mérés*) nem kötelező megcsinálni és lehet párban is készíteni (vagyis a jegyzőkönyv ezen része páronként egyezhet). A jegyzőkönyv értékelését ezek pozitívan befolyásolják. Ekkor kérem jelezzétek, hogy kivel dolgoztatok együtt. A többi feladatot egyénileg kell megcsinálni.

A mérési jegyzőkönyvet a labort követő egy héten belül töltsétek fel.

A jegyzőkönyvnek az adatok egyszerű leírásán tartalmaznia kell az eredmények értékelését is. Az értékelés során ajánlott ábrákat és grafikonokat is használni. A jegyzőkönyv készüljön pdf formátumban. (Nem kell nagy műgonddal megszerkesztett alkotás, de nézzen ki normálisan.)

A feladatok elvégzéséhez bármilyen alkalmazás használható. A segédletben szereplő alkalmazások ajánlások, azoktól el lehet térni, azonban a megfelelő végeredmény a hallgató saját felelőssége. Az adatok feldolgozásához szintén bármilyen eszköz használható (Matlab, Python, Excel, stb.). A használt szoftvereket a jegyzőkönyvben meg kell adni, a feladatok megoldásához használt egyszerű parancsokat a jegyzőkönyvben kell leírni, egy-két sornál hosszabb eljárásokat a jegyzőkönyvhöz mellékelt fájlokban kell beadni. (Részletes kommentezésre nincs szükség, de jelezni kell, hogy melyik program mit csinál.)

Kiegészítések a mérési feladatokhoz

1.3 Feladat

A feladathoz kiválóan alkalmas például valamilyen hírportál kezdőoldala (pl. jól használható az index.hu, de megfelel az origo.hu, nytimes.com, stb. is).

Egy adott oldalhoz letöltött objektumokat, a letöltés idejét, és egyéb dolgokat is nyomon tudunk követni a Developer tools (Fejlesztői eszközök, Ctrl + Shift + i) Network fülében.

A https titkosított kommunikációt használ. Ahhoz, hogy a Wiresharkban meg tudjuk nézni a titkosított üzenetek tartalmát, a Wiresharknak ismernie kell a megfelelő titkosítási kulcsokat. Ehhez az alábbi (vagy ehhez nagyon hasonló) lépéseket kell elvégeznünk:

1. Csukjuk be a böngészőt.

2. Nyissuk meg a környezeti változókhoz tartozó ablakot. Ez elérhető például a sysdm.cpl futtatásával (Win + R and type sysdm.cpl) az Advanced fül kiválasztása után.
3. Hozzunk létre egy környezeti változót SSLKEYLOGFILE névvel és adjuk neki értékül az általunk választott elérési utat mely megadja azt a fájlt, ahol a rendszer az SSL kulcsokat tárolni fogja. Például:C:\Users\cloud\sslkeys.pms
4. Nyissuk meg a Wireshark-ot.
5. A Wireshark Edit/Preferences menüjében válasszuk a Protocol alpontot és állítsuk be a TCP protokollnál a „Reassemble out-of-order segments" opciót.
6. Nyissuk meg a böngészőt és töltsünk be egy oldalt egy HTTPS-t használó webszerverről.
7. A Wireshark Edit/Preferences menüjében válasszuk a Protocols alpontot, és a TLS protokollnál jelöljük be a „(Pre)-Master-Secret log filename" mezőt a korábban megadott, kulcsokat tartalmazó fájlnak. (A példában C:\Users\cloud\sslkeys.pms.)

A vírusirtók esetenként problémát okozhatnak a Wiresharknak, ezért érdemes kikapcsolni őket a feladat idejére.

Nyissuk meg újra a korábban is megnyitott oldalt. Szűrjük ki Wiresharkban az oldal letöltéséhez kapcsolódó kommunikációt. Ha mindent jól csináltunk, a Wireshark alsó ablakában meg tudjuk nézni a titkosítatlan csomagokat. Vessük össze a letöltött objektumok méretét a böngésző Developer tools Network füle alatt találhatókkal. Ehhez célszerű az újra összeállított szegmensek méretét külön oszlopba kiíratni Wiresharkban. Ezt megtehetjük ha például az Edit/Preferences menüpontban az Appearance fülben a Columns alpont alatt új oszlopot állítunk be, ami a tls.reassembled.length mezőt tartalmazza.

2.1 Feladat

A "megfelelően hosszú" (videó/letöltés) nem egy pontos meghatározás. Gondoljuk meg, hogy mennyi mintát szeretnénk, amiből már elég jó statisztikát tudunk készíteni, de még komolyabb gond nélkül feldolgozható. Célszerű az Export Packet Dissections segítségével CSV formátumban menteni a capture-t, és ezt feldolgozni egyéb programokkal.

2.2 Feladat

A csomagok érkezési ideje nem azonos az érkezési időközökkel.
Hisztogram helyett használható tapasztalati eloszlásfüggvény is. A legfontosabb, hogy az alkalmazott statisztikai eszközükkel a viselkedést minél jobban jellemezzük (gondoljunk például az outlierek kezelésére).

3.1 Feladat

Bármilyen szimpatikus VoIP alkalmazás használható (Skype, Teams, Viber, Zoom, stb. – ne mérjük kétszer a Discordot). Számos VoIP program tartalmaz "csend elnyomást" (silence suppression), ezért célszerű valóban valamilyen hangot átvinni a beszélgetés időtartama alatt.

Bónusz feladatként videóhívás is vizsgálható.

4. Feladat

A passzív méréshez használt trace-ről:
A mérés 15 perces időtartamú, nagyjából 5 millió csomagot tartalmaz, melyeknek hossza legfeljebb 1500 byte.
A mérés a 157.253.0.0/16-os hálózathoz tartozó linken készült, a hosztok forgalmi statisztikáinak elemzésekor csak ezeket a hosztokat vegyük figyelembe.
A csomagok hosszához az ip.len mezőt exportáljuk. Nem minden csomagnak van ip.len mezője, ezeket a csomaghossz statisztikákban ismeretlen hosszú üzeneteknek vegyük és külön említsük meg.
A TCP és UDP üzenetek portszámai 0 és 65535 közötti értéket vehetnek fel – beleértve a 0-t is.

FONTOS!
Az eredeti .pcap fájlban számos frame hossz hibás, ezért mindenképp az ip.len mezőt nyerjük ki (ami a csomaghosszt tartalmazza).
Szintén fontos, hogy a trace-ben találhatók ICMP üzenetek ezért, ha tshark segítségével exportáljuk, érdemes használni a "-E occurrence=f" opciót, különben két ip.len mezőt fog exportálni a tshark.

A tshark alapbeállításban a standard outputra küldi szövegesen a kimenetét, ezt tudjuk fájlba kiírni.
Példa a tshark alkalmazására:
```
tshark -r mycap.pcap -E occurrence=f -T fields -e ip.len > mydat.dat
```

Az adatok elemzésénél az alábbiakra lesz szükségünk:

- idő információ
- csomaghossz
- csomag protokoll
- forrás és célport TCP és UDP esetén

Forgalom sebessége: A linken mért sávszélességet a sima görbe érdekében általában érdemes csúszó ablakkal számítani, de a feladat megoldásához elegendő, ha minden másodpercre külön számolunk egy sebességet és ebből készítünk grafikont.

Protokoll statisztikák: A trace jónéhány fajta protokollt tartalmaz. Ezek közül elég a TCP, UDP, ICMP és DNS protokollokat szerepeltetni név szerint, a többit "egyéb protokoll" vagy hasonló címszó alatt is megadhatjuk. (Természetesen aki szeretné vizsgálhatja a többi protokollt is.)

Port statisztikák: Csak a TCP és UDP protokollokhoz tartoznak portszámok, így értelemszerűen csak ezekre kell statisztikát készíteni, de a két protokollra külön-külön. Nézzük meg azt is, hogy ezek a portok milyen alkalmazáshoz tartoznak.

Forgalom megoszlása hosztok szerint: Ehhez a feladathoz érdemes lehet a SplitCap alkalmazást használni. Ebben az esetben érdemes figyelni, hogy a statisztikában csak a 157.253.0.0/16 hálózat hosztjait szerepeltessük.

A javasolt munkamenet SplitCap használata esetén (természetesen ettől szabadon el lehet térni):
1. Az eredeti .pcap fájl szétbontása hosztok szerint.
2. A kis .pcap fájlokból a releváns mezők kinyerése és szöveges fájl(ok)ba írása egy általunk generált, tsharkot használó szkript segítségével.

3. A szöveges fájl(ok) feldolgozása.

Készítsünk ábrát, ami megmutatja, hogy a hosztok mekkora része felelős a forgalom mekkora hányadáért. Néhány pontot (pl. 90%-os forgalmi arány) adjunk meg külön is.

A forgalom flow-k szerinti statisztikái: Ehhez a feladathoz javasolt a SplitCap alkalmazás használata. A javasolt munkamenet megegyezik a hosztok esetén leírtakkal. A trace pontosan 50.000 flow-t tartalmaz, ezért a szétbontott .pcap fájlok szöveges fájlba írás elég sok időbe telik (órák), ezért ezt célszerű a labor végzésétől külön megtenni.