Kódolás és IT biztonság
(VIHIBB01)
Laboratóriumi gyakorlat

Hálózati forgalom elemzése

Szerző:
Holczer Tamás

2023. szeptember 7.

Tartalomjegyzék

1. Mérés célja

A mérés célja, hogy a résztvevők megismerkedjenek a hálózati forgalom elemzésének alapjaival. Képesek legyenek a normális működés paramétereit felvenni, és az attól való eltéréseket észrevenni. A mérés további célja, hogy a detektált hálózati anomáliákat értelmezni tudják, és olyan alapvető kérdésekre tudjanak válaszolni, hogy egy támadás mikor történt, ki volt az áldozat, ki volt az elkövető, és mi történt.

2. Háttér

A hálózati forgalom elemzése nagyon sok munkafolyamatban felmerül. Az elemzés célja lehet egy hálózati hiba felderítése vagy egy támadás elemzése. Az elemzett forgalom lehet élő forgalom vagy rögzített forgalom is. A rögzítésre sok szoftver képes, pár gyakrabban használt megoldás: *tcpdump, tshark, Arkime, Wireshark*. Ezek a szoftverek elemzésre is képesek különböző szűrési és megjelenítési beállítások segítségével. A mérés során a széles körben használt, kényelmes grafikus felülettel is rendelkező Wiresharkot fogjuk használni.

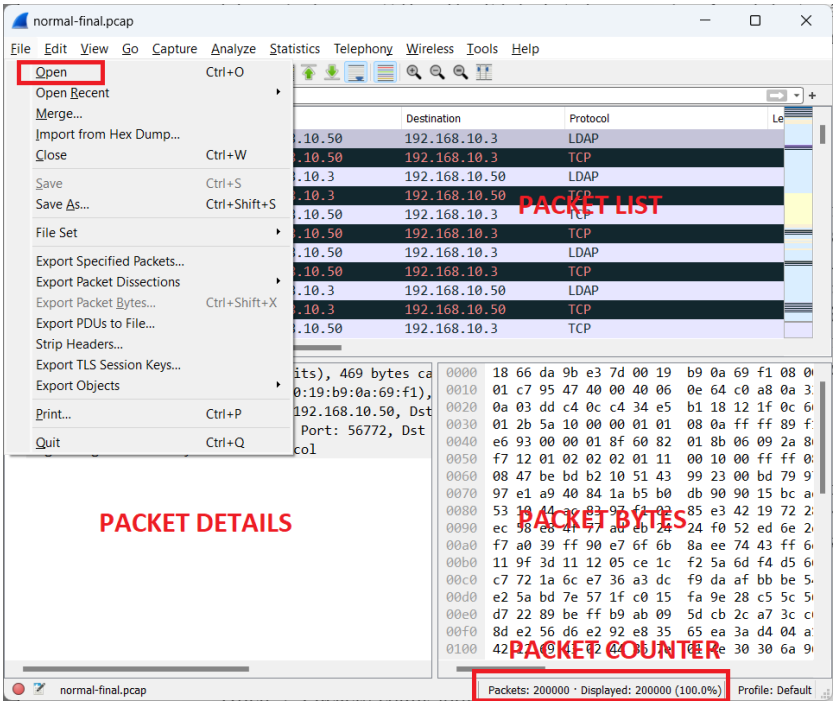
3. Wireshark

A Wiresharkhoz több jó leírás is létezik. Ebben a dokumentumban csak a mérés szempontjából lényeges funkcionalitást ismertetjük. Bármilyen itt nem tárgyalt kérdés esetén érdemes a hivatalos dokumentációhoz fordulni: `https://www.wireshark.org/docs/wsug_html_chunked/`.

A következő alfejezetekben a legfontosabb funkciókat ismertetjük csak röviden.

3.1. Fájlok kezelése

A kezdőképernyőn a csomagok listája, az adott csomag részletezése és bináris tartalma látható alapból. A fájlokat a szokásos módon lehet megnyitni vagy elmenteni. A jobb alsó sarokban az összes csomag, illetve a megjelenített csomagok száma és aránya látható.



1. ábra. Wireshark kezdő képernyő

A `View` menü is többségében a szokásos beállításokat tartalmazza. Egy beállítást érdemes kiemelni: a `Time Display Format` segítségével lehet beállítani, hogy a csomagok relatív (a rögzítés kezdete óta eltelt) vagy abszolút idejét mutassa a csomag lista Time oszlopában a program.

3.2. Statisztikák

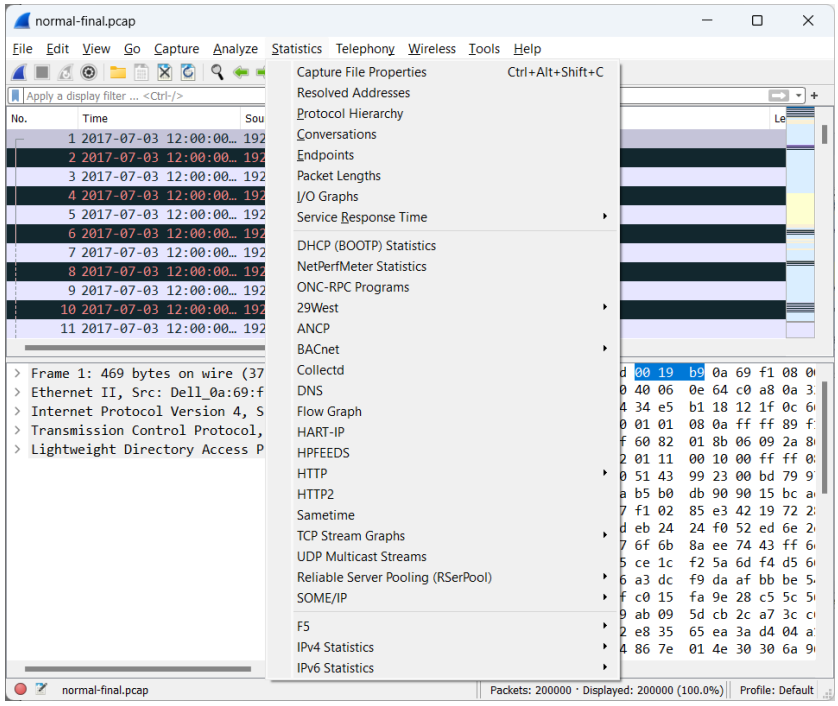
A statisztikai menü mérésben használt részei:

Capture properties: A fájl fontosabb paraméterei

Conversations: Melyik hoszt melyik másik hoszttal társalgott a protokoll hierarchia különböző szintjein. Az elemzés vonatkozhat az összes csomagra vagy csak a mutatott (kiszűrt) csomagokra beállítástól függően.

I/O Graphs: A forgalom időbeli lefutását lehet vizsgálni. Az összes csomagot vagy filterekkel kiszűrt forgalmakat is lehet ábrázolni.

Service response times: Különböző protokollok kiszolgálási idejét lehet ezzel elemezni.



2. ábra. Wireshark statisztikai elemzés

3.3. Szűrés és analízis

A Wireshark legnagyobb erőssége az interaktív analízis. Az analízis folyamán különböző display filtereket lehet megadni, és így leszűkíteni a mutatott csomagokat a minket érdeklőkre. Filtereket a forgalom rögzítése közben is lehet használni (capture filter), de ezek halmaza jóval szűkebb mint a megjelenítésnél használható filterek halmaza.

Pár példa filter, ami hasznos lehet:

ip.addr==192.168.1.1 adott című hoszt forgalma (a hoszt küldő vagy fogadó is lehet)

ip.addr==192.168.1.0/24 adott alhálózat forgalma (az alhálózatbeli hoszt küldő vagy fogadó is lehet)

ip.src==192.168.1.1 az adott című hosztról küldött IPv4 csomagok (létezik dst is)

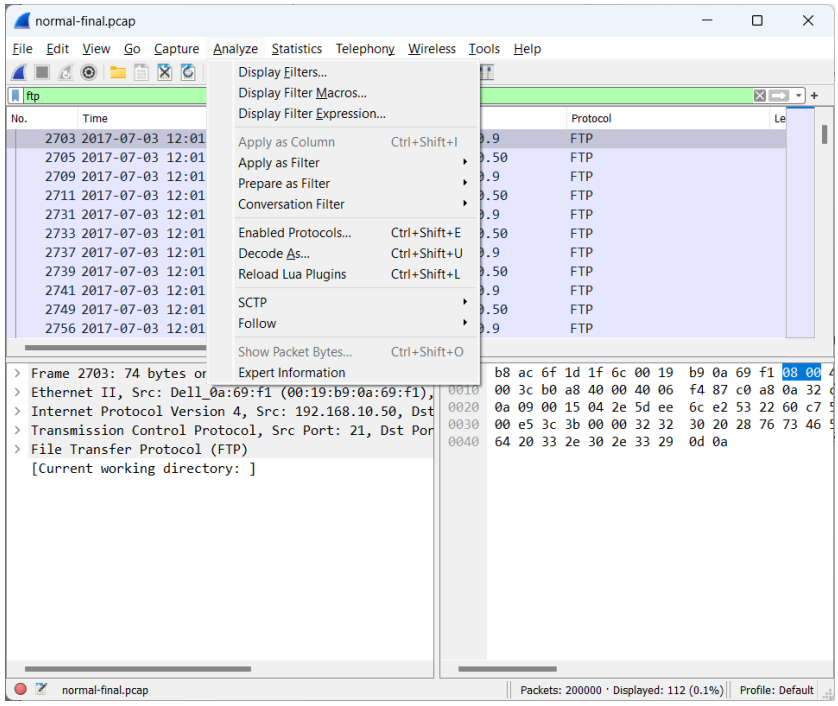
tcp.port==80 csak az adott portról vagy portra küldött csomagok (létezik src és dst is)

A filtereket logikai műveletekkel egymáshoz is lehet kapcsolni (`not and or`) illetve nem csak egyenlőséget lehet vizsgálni, hanem más szokásos operátorok is működnek. A protokollokon belül mélyebbre is le lehet menni, és az ott található értékeket vizsgálni, például az `ftp.response.code==230` a 230-as ftp válaszkódot tartalmazó üzeneteket válogatja ki.

Ha sikerült egy érdekes kommunikáció legalább egy csomagját megtalálni, akkor az `Apply as Filter` segítségével a többi csomag könnyen kiválasztható, míg a `Follow` segítségével a párbeszéd adattartama jeleníthető meg könnyen olvasható formátumban.

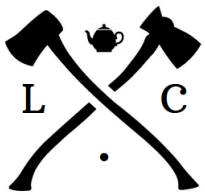
A Wireshark elég jó a protokollok kitalálásában, de bizonyos esetekben össze lehet zavarni. Ilyenkor segíthet a `Decode As` lehetőség, amivel meg lehet mondani, hogy egy adott porton folyó kommunikációt milyen protokollként értelmezzen.

A protokollok értelmezéséhez nagyon sok beállítás elérhető az `Edit/ Preference/ Protocols` alatt. Itt lehet például egy TLS mesterkulcsot megadni, aminek a segítségével bele lehet nézni az adott kulcshoz tartozó forgalmakba is (a Wireshark automatikusan dekódolja az üzeneteket).



3. ábra. Wireshark elemzés

4. Háttértörténet



A Lines&Code cégnél nagyon figyelnek a biztonságra. A hálózati forgalom rögzítése egy bevett gyakorlat, így észre lehet venni a hálózaton keresztül jövő támadásokat, illetve meg lehet nézni a hatásukat. A gyakorlat folyamán a cég komplex irodai hálózatában rögzített forgalmat kell vizsgálni. Az első feladat a normális működés rögzítése az első napi fájl alapján. Ezek után sajnos többféle támadás érte a hálózatot, így a gyakorlat további részében ezeket a támadásokat kell felderíteni a következő napokon rögzített fájlok segítségével.

A gyakorlaton használt forgalmi mentések természetesen nem egy valós cégtől származnak, és napok helyett csak perceket tartalmaznak a könnyebb kezelhetőség érdekében. A fájlok létrehozásához az alábbi forrásokat és eszközöket

használtuk (nagy köszönet a szerzőknek):

1. University of New Brunswick, Canadian Institute for Cybersecurity kutatói által készített virtuális iroda forgalma[1]
2. Nagy Roland által készített demonstrációs malware forgalma
3. *tcprewrite* ha a csomagok tartalmát meg kellett változtatni (például IP címek cseréje)
4. *editcap* ha a fájl metaadatait meg kellett változtatni (például rögzítés időpontja)

5. Feladatok

A feladatok folyamán a Lines&Code-ot ért támadást kell felderíteni Wireshark segítségével. A hivatkozott pcap fájlok a felhasználó *final_pcaps* könyvtárában találhatóak.

5.1. feladat (vezetett)

A feladat folyamán megvizsgáljuk a hálózat normális működését (különös tekintettel az FTP forgalomra, mert annak még lesz szerepe később):

- Nyissa meg a *normal-final.pcap* fájlt *Wiresharkban.* A megnyitás egy kis időt vesz igénybe, mert ez egy közel 180 MB-os fájl. Ezt lehetetlen manuálisan elemezni, de szerencsére a Wireshark képes vele megbirkózni.
- Vizsgálja meg a fájl tulajdonságait (formátum, hash, rögzítés ideje).
- Vizsgálja meg a párbeszédeket. Melyik két hoszt közötti IPv4 forgalom generálta a legtöbb forgalmat. Milyen jellegű forgalom volt ez? Mit lehet tudni erről a szerverről egy gyors keresés alapján?
- Vizsgálja meg az összes átküldött csomagot az idő függvényében.
- Vizsgálja meg az FTP forgalmat az idő függvényében. Melyik IP-t használhatja vajon az FTP szerver? Mi az *iscxtap* user jelszava?

[1] `https://www.unb.ca/cic/datasets/ids-2017.html`

- Opcionális: Vizsgálja meg az SMB2 jellemző válaszidejét.

A feladat folyamán meg lehet vizsgálni egyéb protokollokat alaposabban, de nem szükséges a többi feladathoz. Természetesen valós körülmények között nem érdemes az FTP-nél leragadni.

Beadandó: az **iscxtap** FTP user jelszava, ami egy szám.

5.2. feladat

Ebben a feladatban egy jelszófeltörés nyomait kell felderíteni:

- Nyissa meg a *attack-final.pcap* fájlt *Wiresharkban.*
- Vizsgálja meg az összes forgalom időbeliségét és hasonlítsa össze az előző feladatban kapott eredményekkel.
- Vizsgálja meg az FTP forgalom időbeliségét. Mintha megnövekedett volna a forgalom, de pontosan mikor is?
- Vizsgálja meg az FTP forgalmat a gyanús idősávban. Milyen IP címről érkeznek a támadások? Ki az áldozat?
- Sikerült bejutnia? Ezt a legkönnyebben az `ftp.response.code==230` filterrel lehet megvizsgálni.
- Hány sikertelen FTP próbálkozás található a fájlban? Módosítsa az előző filtert megfelelően és rögzítse a találatok számát.

Beadandó: Az áldozat IP címe A.B.C.D formátumban.
Beadandó: A sikertelen FTP kísérletek száma a fájlban.

5.3. feladat

Ugyan az előző fájlban nem látszik, de további elemzés alapján úgy tűnik, hogy a támadó később mégis bejutott a szerverre. Ott több mindent is csinált, de az egyik részlet teljesen érthetetlen. Egy távoli szerverrel kommunikált az áldozat (nem az ftp próbálkozást indító géppel), de ránézésre a Wireshark se érti mi történik:

- Nyissa meg a *cnc-final.pcap* fájlt *Wiresharkban.*

- Úgy tűnik, hogy a támadó az előző feladatból mintha mégis célt ért volna, mert furcsa kommunikációra bukkantunk. Milyen külső IP-vel és porttal kommunikált az áldozat (segítségül: a port számjegyeinek összege 16)? A feladat megoldása során érdemes azokra a csomagokra szűrni, aminek az áldozat a feladója, és a cél nem ugyanabban az alhálózatban van. Ha túl sok a találat, akkor a kizárható protokollokat (pl ntp) is érdemes kizárni a szűrőben. Ha megvan a port, akkor érdemes arra szűrni, hogy a kommunikáció mindkét iránya látható legyen.

- Ugyan a port egy ismert alternatív http port, de ez biztos nem http. Mi lehet ez? Esetleg http titkosítva? Nézzük meg a Decode megfelelő beállításával.

- Igen, ez valóban https (http forgalom TLS-el titkosítva). Jó lenne megnézni a tartalmát, de ehhez szükség lenne a titkos kulcsokra. Szerencsére pont úgy volt beállítva a gép, hogy ezt rögzítette. Állítsa be a secret fájl használatát TLS dekódoláshoz.

- Milyen parancsot küldött a szerver az áldozatnak elsőre?

- Kinek a nevében futott az áldozaton a processz? A TLS párbeszéd követésével könnyen megtalálható.

- Opcionális: Ez a fájl jól láthatóan manipulálva lett (a gyakorlat könnyebb végrehajtása érdekében). Vajon mennyi idővel lett eltolva az eredeti fájl (figyeljen az időzónákra is)?

Beadandó: A támadó IP-je és portja A.B.C.D:X formátumban (az IP és PORT közötti kettőspont kötelező).

Beadandó: Kinek a nevében futott a processz az áldozaton?