Kutatási beszámoló

Ransomware és egyéb kártékony kódok elleni védekezés kutatási projekt háttérinfrastruktúrájának megteremtése

Dr. Bencsáth Boldizsár

BME Hálózati Rendszerek és Szolgáltatások Tanszék

2018

Összefoglaló

A projekt célja a ransomware és más kártékony kódok elleni védekezés hatékonyságának növelése. A hatékonyság növeléséhez alapvetően szükséges, hogy friss, a vírusírtók adatbázisában nem szereplő ransomware mintákat keressünk. Ennek érdekében pedig hatékony algoritmusokra van szükség az ismert ransomware mintákhoz hasonló új kártevők felfedezéséhez, a projekt ezzel foglalkozik.

Munkamódszer

A friss ransomware mintákat több forrásból próbáljuk beszerezni, két forráshoz tudtunk támaszkodni:

- Egy nagy malware adatbázishoz, amely több tíz millió malware mintát tartalmaz
- Egy folyamatosan érkező feed-hez, ami napi kb. 40-300 ezer malware mintát tartalmaz

Mindkét adatforrással kapcsolatos probléma, hogy nem klasszifikált, azaz az egyes minták nincsenek besorolva, így nem lehet tudni, mi milyen okból került fel a forrásra, ártalmas-e, vagy sem, ransomware-e vagy sem.
A malware classification egy ismert probléma, számos kutatás kezeli ilyen-olyan szinten, de nincsen általánosan elfogadott legjobb megoldás és nem is várható.

Munkánk során a kapcsolódó hallgató dolgozott a klasszifikációs módszerek tesztelési módszertanának kidolgozásán, míg én az infrastruktúra környezet kialakításán és az alkalmazott módszertan egyeztetésén és validációján dolgoztam.

Motiváció

Ahhoz, hogy vizsgálódni tudjunk a témában, mindenekelőtt szükségünk van friss ransomware-ekre. Azért fontos, hogy frissek és aktívak legyenek, mert így a malware gyártók jelen technikáival ismerkedhetünk meg.

E feladat megoldásához először információt gyűjtött a hallgató arról, hogy mások, akik hasonló témában dolgoztak, milyen módszerrel gyűjtöttek mintákat. Számos cikk elolvasása után végül arra jutottam, hogy legtöbben publikusan elérhető malware adatbázisokat használtak vagy mintákat kaptak egy anti-vírus üzemeltetőtől. A kigyűjtött adatbázisokat áttekintve megállapítottuk, hogy számos közülük már sajnos nem üzemel. A továbbra is aktív státuszban levők között voltak ingyenesek és fizetősök is, amik lehetővé tették fájlok letöltését SHA hash-ük alapján.

Azonban mindegyiküknek volt egy hiányossága, ami alkalmatlanná tette őket számunkra: nem lehetett család vagy osztály szerint listázni, keresni bennük. Az egyetlen ilyen szolgáltatást a

VirusTotal nyújtja, ahol YARA hunting szabályok alkalmazásával kereshetünk a folyamatosan érkező fájlok között, majd ezeket le is tölthetjük.

Első opcióként egy speciális módszerrel próbálkoztunk, az Interneten talált és néhány minta alapján saját magam által írt hunting szabályokat fogalmaztunk meg WannaCry, BadRabbit, Petya és ismeretlen kategóriájú zsarolóvírusok keresésére az általunk hozzáférhető, speciális, de széles körben használt adatbázison, a Virustotalon. Ez az adatbázis keresési lehetőség önmagában kuriózum, mert az adatbázishoz való hozzáférés jelentős anyagi költséggel járt. A hunting szabályokon alapuló keresés sikertelen volt, főleg azért, mert nem volt elég tapasztalat az ilyen keresések beállításával kapcsolatban.

Ezek alapján új célt adtunk a kutatásnak: készítsünk malware adatbázist, ahol családok szerint csoportosítva vannak a minták. Ehhez rendelkezésre állt számomra az partnerünk kezelésében levő malware adatbázisa, mely 300 000 000 mintát tartalmaz, napi ~40 000 újonnan érkező fájllal.

Kutatási folyamat

A kutatást a meglévő ismert LSH algoritmusok vizsgálatukkal kezdtük.

Az ssdeep az egyetlen elterjedt LSH algoritmus. Ahol elérhető LSH hash ott ez elérhető. Az SSDEEP egy spamsum-nak keresztelt algoritmusra épül, melyet Andrew Tridgell 2002-ben spam-detektálás segítésére hozott létre. Ehhez szükség volt egy adatbázisra, mely tartalmazta ismert spam emailek ssdeep hash értékeit. Egy beérkező emailre ki kellett számolni a hash-t, végül ennek értékét az összehasonlító algoritmus segítségével összehasonlítani az adatbázisban található hashekkel. Ha a távolság (score) értéke elért egy küszöböt, akkor vélhetően a beérkező email is spam volt.

A 64 részre osztás megfelelő módszer lehet spam emailek összehasonlítására, hiszen ezek gyártói egy email sablont kisebb változtatásokkal többször is felhasználnak. Binárisok esetében azonban más a helyzet. Esetükben egyetlen fordító flag megváltoztatásával drasztikus különbség érhető el a lefordított állományban. Tegyük fel például, hogy a mind a 64 részbe valamilyen módon beillesztünk egy-egy NOP (No OPeration) utasítást. Ez nem változtatja meg a program működését, azonban a 64 részből képzett minden block hash bájt értéke megváltozik és a két fájlra a kimeneti hash-ek értéke teljesen különböző lesz. Ez csupán egyetlen példa, amivel ki lehet játszani az ssdeep-et, ezen gondolatmenet alapján rengeteget találhatunk még.

Azonban az imént említett támadás főként akkor jelent problémát, ha a támadók azt a célt is kitűzik maguk elé, hogy átverjék az ssdeep alapú összehasonlítást és szándékosan ennek megfelelően módosítják programjaikat.

TLSH:

A TLSH a TrendMicro Locality Sensitive Hash rövidítése. Ez egy újabb, 2013-ban publikált eljárás. Lényege, hogy statisztikát készít a fájlban előforduló bájt n-esekről. Ehhez egy 5 bájtos csúszó ablakkal halad végig a bemenet bájtjain. A csúszó ablakban található bájt 3-asokat

vizsgál, ezeket leképezi egy bájtra majd eggyel növeli annak a vödörnek az értékét, ami a képzett bájt előfordulásait tartalmazza. Végül ebből az eloszlásból képzi a hash-t, mely rögzítetten 64 bájt hosszú.

Az összehasonlítás leginkább Hamming távolság számítására hasonlít a két hash között. A kimeneti érték nullától egészen ezres nagyságrendig felmehet, bár az ezret ritkán haladja meg. A nulla tökéletes hasonlóságot jelent, ellentétben az ssdeep-pel.

A publikáló cikkben szerepel az algoritmus vizsgálata és értékelése is. Kivételesen binárisokat, sőt, malware-eket is használtak a módszer eredményességének megállapítására, azonban ez csupán 169 malware-t tartalmazott. Véleményem szerint ilyen kicsi adathalmaz esetén csupán felületes eredményt kapunk az algoritmus malware-eken értelmezett pontosságáról.

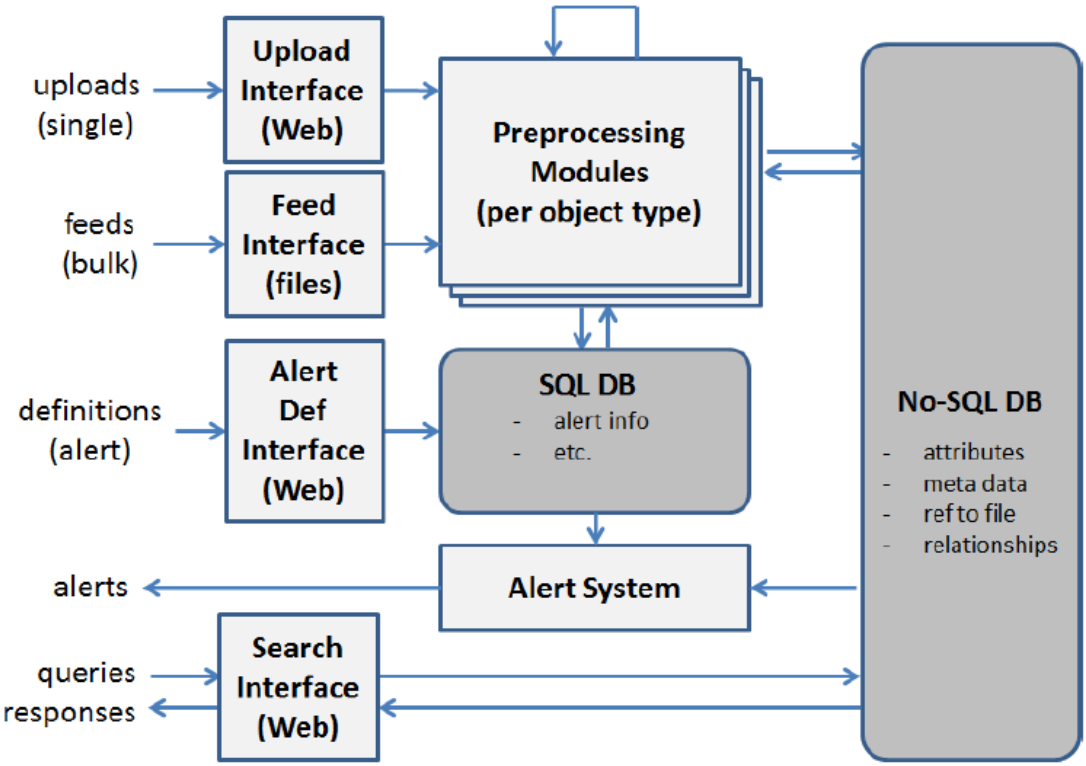
SDHASH:

Az sdhash-t 2009-ben implementálták. Alapvető ötlete, hogy statisztikailag valószínűtlen tulajdonságokat keresünk a bemenet bájtjaiban, ezekből képződik a hash érték. Egy valószínűtlen tulajdonság jelenléte egyedien jellemez egy fájlt vagy fájl csoportot.

A valószínűtlen tulajdonságok megállapításához a bementet 64 bájtos egységenként tekinti, ezekre számol entrópiát. Végül az entrópia eloszlásából állapít meg valószínűtlen tulajdonságokat.

Sajnos azt sdhash-t bemutató cikkben sem végezték el az algoritmus tesztelését malware-eken.

Algoritmusok tesztelése



Mint azt korábban említettük az algoritmusok tesztelését és összehosnlítását két különböző rendszerben végeztük.

Az egyik rendszer alapja egy napi malware feed, itt több különböző tömörítettt fájlban több tízezer friss malware kerül bedolgozásra.

A tesztelésre létre lett hozva egy ismert ransomware malware corpus, amely alapnak számít a hasz összehasonlító algoritmusoknak az új, napi mintákhoz való összehasonlításhoz, amelyet apróbb-nagyobb sciprtekkel és némileg változtatott open source toolokkal végeztünk el.

A másik, nagyobb vizsgálat során a meglévő óriási malware corpust használtuk, ami egy nagy, nagyságrendileg 200 TB tiszta adatméretű hadoop alapú malware adattár. Az adattár tömör architektúráját a fenti ábrán mutatjuk be. A HBase formátumban tárolt no-sql adatbázisban tárolt malware mintákon levő hash futtatás különleges kihívás: Speciális módon kell futtatni lekérdezéseket, jelen esetben ún mapreduce jobok által, és az eredményeket magában a hadoop (cloudera) klaszterben kell tárolni automatikusan. Egy teljes adatbázisra való lefutás hetekbe telik, ezt indítottuk el, és az eredmények feldolgozása sem triviális.

Algoritmus teszt-eredmények

Mindhárom csoportosító algoritmusnál a TLSH bizonyult a legjobbnak, nyomában az SSDEEP-pel és sokkal mögöttük lemaradva érkezett az SDHASH. Végül rájöttem az SDHASH gyenge teljesítményének okára. Az adatbázisban elérhető fájlok nem binárisok voltak, hanem azoknak object dump leképezéseik. Ez nagyon befolyásolhatta az SDHASH működését, hiszen egy valószínűtlen tulajdonságnak tekinthető, hogy minden fájlban hexadecimális karakterek vannak és kettesével elválasztja őket egy szóköz.

A napi feed futtatási eredményei azt mutatják, hogy az algoritmusok összehasonlítása fontos és hasznos, és segítségével pontosabban tudunk találni valódi friss ransomware mintákat. A teljes adatbázis átvizsgálása még folyamatban van, így annak eredménye csak később értékelhető.

Budapest, 2018. 03. 15. Dr. Bencsáth Boldizsár