Предлог пројекта
Детектовање малвера помоћу машинског учења

Идеја и мотивација:

Моја идеја је да направим модел који ће коришћењем машинског учења препознавати извршне фајлове као безопасне или као малвер.

Старији приступ који су антивируси користили је да поседују ручно направљену базу малвера у којој су малвери описани неким својим карактеристикама, а антивируси су само проверавали да ли се фајл који скенирају налази у бази. Овакав начин рада захтева стално ажурирање базе података и није могуће детектовати малициозне фајлове све док се не појаве у бази. Карактеристике које се користиле за описивање малвера у бази формирају јединствену сигнатуру (fingerprint) вируса која је осетљива на веома мале промене у извршном фајлу, а за сваку промену се мора правити ново правило детекције.

Антивируси који користе машинско учење су много флексибилнији па могу сами закључе да су слични фајлови малициозни без тога да су сваки од њих видели на тренингу, а могу и да донесу закључке о потпуно новим фајловима те се много лакше прилагођавају растућем броју малвера.

Референца: *Kaspersky-Lab-Whitepaper-Machine-Learning.pdf* [1]

Циљ пројекта и начин коришћења:

Циљ пројекта је направити брз и поуздан скенер за малициозним фајловима који би се могао користити за заштиту мејлова на Linux мејл серверу и сличним системима који пружају услуге размене података. Иако би радио на Linux серверу, програм би требало да детектује и Windows и Linux вирусе ради заштите клијената сервера.

Пројекат би могао бити примењен и у друге сврхе, али прилагођавање наведеном радном окружењу би био главни циљ.

Опис референтног рада:

Референце:

Santos и сар. - 2013 - OPEM A Static-Dynamic Approach for Machine-learning-based Malware Detection [2]

Malware Detection Using 1-Dimensional Convolutional Neural Networks [3]

Пројекат се састоји из прављења хибридног сета података о извршном фајлу који садржи карактеристике прикупљене и статичком и динамичком анализом који се затим користи за тренирање и тестирање модела.

Статичка анализа:

Статичком анализом се прикупљају подаци о фајлу без његовог покретања. Ови подаци се састоје од карактеристика добијених из фрагмената кода, хешева фрагмената кода и других карактеристика самог фајла.

Референтни рад[3] објашњава методу TF-IDF која се бави само карактеристикама добијеним из фрагмената кода на следећи начин (укратко):

- Формирају се све могуће секвенце од по n команди (n-gram) из програма и затим се рачуна колико пута се дата секвенца појављује у програму у односу на број секвенци (term frequency – TF).

- За сваку секвенцу рачуна се IDF (inverse document frequency) параметар који обезбеђује да се секвенцама које су учестале у свим фајловима не додели превелика важност.

- Крајња карактеристика је производ TF * IDF. Број карактеристика је број секвенци. Карактеристике које имају веома велику учесталост у свим фајловима највероватније представљају стандардне процедуре при раду програма и нису од великог значаја па се могу избацити.

Динамичка анализа:

Динамичком анализом се добијају подаци о програму на основу његових акција током извршавања. Такви програми се покрећу у изолованом sandbox окружењу, али тако да за малвер буде што теже препознати да ради у sandbox-у јер неки малвери у том случају престају са радом да би онемогућили њихово откривање и истраживање. Због тога окружење у ком малвер ради треба бити што сличније нормалном.

У референтном раду[2] карактеристике динамичке анализе су представљене вектором логичких променљивих чија вредност зависи од тога да ли је дошло до одговарајуће акције током извршавања програма.

Спајање података из статичке и динамичке анализе даје сет карактеристика спремних за коришћење у моделу машинског учења.

База података:

MalwareBazaar је сајт са базом података од преко 500 000 примерака малвера. Користећи њихову колекцију сакупио бих довољно малвера за тренирање и тестирање, а да би сет података био балансиран, додао бих сигурне фајлове са својих рачунара и других извора.
Сајт који је коришћен у референтном раду више није доступан.

Метрике:

Као и у раду [3], и ја бих користио тачност у препознавању малициозних фајлова, тачност у препознавању безбедних фајлова и укупну тачност.

Жељене модификације и хипотеза:

У раду[2] је експериментално показано да хибридни сет карактеристика доноси већу тачност од искључиво статичког или динамичког сета, али је примећено да резултати на основу само динамичке анализе нису велики у односу на резултате статичке анализе ни на једној метрици.

Разлог томе је вероватно доста мањи број карактеристика добијених динамичком анализом у односу на карактеристике статичке анализе (63 према 1000 где су тих 1000 издвојени из 144598 карактеристика корелационом таблицом).

Оно што бих ја хтео да урадим у свом пројекту јесте:

- Побољшање динамичке анализе већим бројем карактеристика
- Самостално писање правила за SELinux и SECCOMP током тренирања.
- У раду [2] су коришћени разни класификатори, али нису испробане неуралне мреже док су у раду [3] коришћене неуралне мреже, али без динамичких карактеристика. У свом пројекту бих пробао да објединим та два.
- Проверити резултате када се употребљују и аргументи команди у креирању секвенци уместо секвенци самих команди
- Додати тежине за динамичку анализу уместо само 0/1 параметра за десила се/није се десила акција

Главни циљ би био постићи доста високе вредности за све 3 коришћене метрике.

Метода:

Прављење sandbox-а и динамичка анализа:

Прављење sandbox окружења би било остварено коритећи следеће функционалности:

- Namespace-ови за одвајање малвера од остатка фајл система и других ресурса
- btrfs snapshot за креирање копије фајл система у којој ће малвер радити тако да има приступ фајловима, али не поремети структуру фајл система за остале процесе. Snapshot-ovi се брзо праве и бришу тако да би на почетку тестирања сваког фајла чист snapshot био направљен и након тестирања обрисан.
- cgroups за заштиту система од DoS напада од стране малвера са идејом сличном fork бомби (заузети све ресурсе система)

- SELinux за праћење специфичних акција као што су приступи одређеним фајловима, портовима, коришћење привилегија (capability) и слично и њихово бележење
- SECCOMP за праћење одређених системских позива од значаја и њихово бележење
- За Windows програме био би доступан и Wine log Win32 API позива датог програма
- ptrace(2) за потпуно праћење процеса

Избегао бих виртуелизацију целог система због перформанси и додатне комплексности.

За неке честе малициозне акције већ би правила за SECCOMP и SELinux била уграђена са одговарајућим тежинама. Ни SECCOMP ни SELinux не би блокирали акције за шта се иначе користе већ би само слали сигнал userspace процесу када се дата акција деси. Изолованост sandbox-а спречава малвер да направи штету, не SELinux и SECCOMP, они су само оптимизациона техника да се провере за траженим акцијама одиграју унутар језгра система и userspace процес не буде позван за сваку акцију као што је случај са ptrace-ом.

Током тренирања скенер би формирао полису безбедних акција на основу audit log-а акција програма маркираних као безбедни, а осталим акцијама из малициозних програма доделио одговарајућу тежину на сличан начин као TF-IDF. Тако би формирао сет карактеристика базираним на акцијама које ће и касније пратити.

Током тестирања SELinux/SECCOMP би пратили акције покренутог програма и акције које нису у полиси пријављивали userspace програму који врши скенирање. Ако скенер није трениран да анализира ту карактеристику, додао би полису за SELinux и SECCOMP да убудуће игноришу ту карактеристику да се не би трошило време на њу.

Ово захтева да тренинг сет буде што већи јер скенер тада формира листу акција које посматра. Ако је тренинг сет довољно велики, то би решило и проблем малог броја динамичких карактеристика.

Идеја за коришћење SECCOMP-а и SELinux-а уместо само једног од та 2 је то што би SECCOMP упозоравао на системске позиве који не би требало да буду позвани док би за уопштене системске позиве као што је write(2) SELinux био подешен да прати шта се тачно дешава док би SECCOMP дозволио све write(2)-ове. Зато за ситуације које и SELinux и SECCOMP детектују полиса би требало да буде таква да избегне дуплу пријаву акције.

Статичка анализа:

Као метод прикупљања статичких карактеристика користио бих TF-IDF објашњен у раду [3]. Скуп секвенци који се посматра би такође био формиран током тренирања и остале секвенце које се појаве током тестирања би биле одбачене тако да је битан репрезентативан тренинг сет.

Модел машинског учења:

У случају коришћења TF-IDF метода карактеристике би биле убачене у потпуно повезану неуралну мрежу као у раду. У случају великог броја карактеристика неке од њих би биле одбачене користећи корелациону анализу.

Ако се коришћењем TF-IDF-а не постигну задовољавајући резултати ни на један начин, може се користити и метода 1D конволуционе неуралне мреже споменуте у раду [3]. Тада би се за динамичку анализу мрежи проследили стрингови секвенци системских позива.