ӘОК: 004.855.5

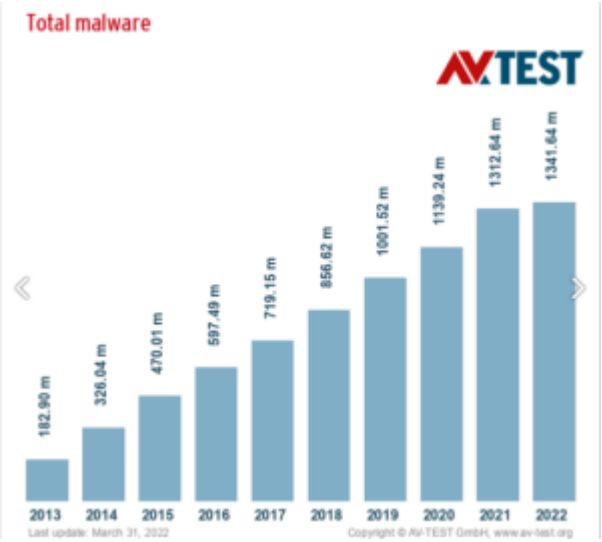
ДИНАМИКАЛЫҚ ТАЛДАУ НЕГІЗІНДЕГІ КОМПЬЮТЕРЛІК ВИРУСТЫ ЗЕРТТЕУ ӘДІСТЕРІ

Мұхамәдиев Мадияр Ғалелұлы
mdrenu@mail.ru
Л.Н.Гумилев атындағы атындағы Еуразия ұлттық университенің
Ақпараттық технологиялар факультетінің 1-курс магистранты, Нұр-Сұлтан, Қазақстан
Ғылыми жетекшісі – Қонырханова А.А.

Қазіргі таңда зиянды бағдарламаларды анықтау және алдын алу әдістері интернетке қосылған компьютерлік жүйелер үшін қажет болуда. Зиянды бағдарламаны компьютерлерге, ұялы телефондарға және басқа да электрондық құрылғыларға зиян келтірмес бұрын, алдын ала анықтау қажеттілігі көптеген жылдар бойы зерттеушілер мен зиянды бағдарламалардан қорғайтын мамандардың назарында.

Соңғы жылдардағы интернет пен заманауи технологиялардың қарқынды дамуы, Интернет қауіпсіздігінің бұрын-соңды болмаған және ауыр жағдайға тап болуына себеп болды. Компьютерлерге, ұялы телефондарға және басқа құрылғыларға зиянды бағдарлама шабуылдарының қаупі күн сайын артып келеді. Зиянды бағдарлама деп аталатын зиянды бағдарлама – шабуыл жасау ниеті бар бағдарлама. Зиянды бағдарламалардың саны 1-суретте көрсетілгендей бірнеше жыл бойы жоғары жылдамдықты өсу үрдісін сақтап қалды. Мұндай үлкен көлемдегі зиянды бағдарламалар файлдарын қолмен өңдеу мүмкін емес екені түсінікті. Сондықтан, вирусқа қарсы бағдарламалық құрал өнімдері негізінен зиянды бағдарламаларды анықтау үшін қолтаңбаға негізделген әдістерді қабылдады. Қолтаңбаға негізделген әдістерде ұсталған зиянды бағдарламадан байттардың бірегей тізбегі шығарылады және ұқсас зиянды

файлдарды анықтау үшін пайдаланылады. Дегенмен, шабуылдаушылар түрлі вирусқа қарсы бағдарламалық құрал арқылы, вирустын анықталмас үшін оның қолтаңбаларын оңай өзгерте алады. Зиянды бағдарламалық құралды анықтаудағы негізгі қиындық зиянды бағдарламаның таралу барысында мутацияға ұшырау мүмкіндігі болып табылады. Осы негізде зиянды бағдарламаны полиморфты метаморфтық деп жіктеуге болады. Полиморфты зиянды бағдарлама өзін хост бағдарламасына тіркегеннен кейін зиянды пайдалы жүктемені шифрлайды. Олар жасалған сайын пайдалы жүктеме жаңа кілтпен шифрланады және пайдалы жүктеменің шифрын шешу үшін шифрды шешу тәртібі қажет.



1-сурет. 2013-2022 жыл аралығындағы AVTEST
2-бойынша зиянды бағдарламалардың саны
3-

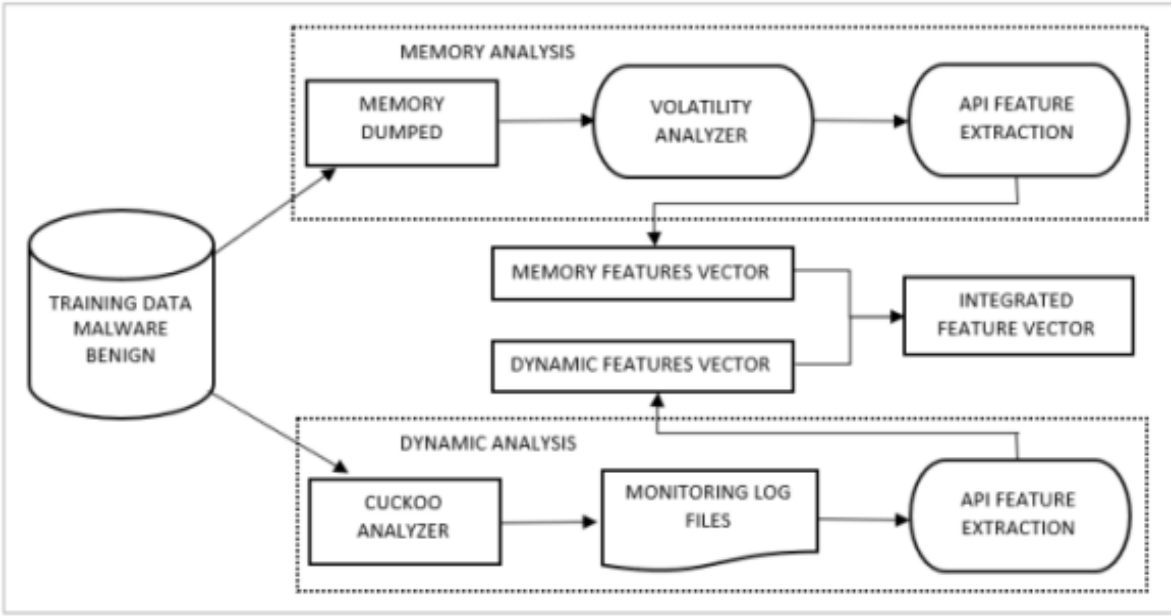
Екінші жағынан, метаморфтық зиянды бағдарлама кодты жасыру үшін қалаусыз кодты енгізу, кодты ауыстыру, регистрді қайта тағайындау және т.б. сияқты әдістерді пайдаланады. Демек, метаморфтық зиянды бағдарлама ата-аналарынан синтаксистік түрде ерекшеленетін зиянды бағдарлама көшірмелерін жасау үшін өз кодын өзгертеді.

Зиянды бағдарламаларды талдаудың дәстүрлі әдістері негізінен статикалық талдау және динамикалық талдау болып табылады. Статикалық талдау PE файл тақырыбын, бағдарламадағы ашық мәтін жолдарын және бағдарламаның бөлшектеу код ақпаратын қоса, белгісіз үлгілік бағдарламаны іске қоспай, үлгілік бағдарламаның бастапқы кодын статикалық сканерлеуді және талдауды білдіреді. Осы талдаулар арқылы біз бағдарламаның әрекеті мен зияндылығын анықтаймыз. Бұл әдістің артықшылығы - ол бағдарламаның бірнеше орындалатын жолдарын талдай алады, бірақ ол қолмен қатысуды және жоғары кәсіби дағдыларды қажет етеді. Мозердің зерттеулері көрсеткендей, статикалық талдау жаңа кодты жасыру әдістерін қолдану себебінен мұндай зиянды кодты дұрыс жіктей алмайды. Үлгі бағдарламаны іске қоспай, статикалық талдау үлгінің орындалу уақыты мен жүйе арасындағы интерактивті ақпаратты әрең ала алады, нәтижесінде автоматтандырылған талдаудан алуға болатын ақпарат аз болады.

Динамикалық талдау немесе оны мінез-құлық талдауы деп те атайды, басқарылатын қауіпсіздік ортасында үлгілік бағдарламаны іске қосуды , процестің орындалу уақыты кезінде бағдарламаның орындалуы мен жұмыс күйін бақылауды және қадағалауды, бағдарламаның орындалу барысын түсіруді, жүйелік ресурстарды пайдалануды қажет етеді. Осы негізде біз негізгі ортадан оқшауланған сэндбоксын құра аламыз. Бұл құралда біз клиенттік жүйені іске қосамыз және бағдарламаны орындаймыз, ал бақылау бағдарламасы бағдарламаның іске қосылуын автоматты түрде бақылайды және оның API шақыру ақпаратын жазады. Байер және т.б. үлгіні іске қосу және үлгіні орындау ағыны ақпаратын талдау үшін TTAnalyze деп аталатын сэндбоксты жобалау үшін модельдеу әдісін пайдаланды; Виллемс және т.б. виртуализация технологиясымен CWSandbox деп аталатын құрылғыны іске асырды.

Әдістеме және құрылым

Бұл бөлімде талдаудың бір тәсілдің архитектурасы егжей-тегжейлі талқыланады. Деректер жиынтығы зиянды файлдар жиынтығына қосымша екі көзден жиналған зиянды файлдардан тұрады. Содан кейін талдау ортасы реттеледі, ол үлгілерді орындау және қажетті талдау есептерін жасау үшін қолайлы атмосфераны қамтамасыз ету үшін барлық қажетті компоненттерден тұрады. Содан кейін объектілерді шығару жүзеге асырылады және алдын-ала модельдеу қолданылады. Сонымен, мәліметтер жиынтығы машинаны оқытудың бірнеше модельдерін жіктеудің дәлдігін үйрету және тексеру үшін қолданылады. Жалпы құрылым 2-суретте көрсетілген және келесіде егжей-тегжейлі талқыланады.



4-Сурет. Талдау архитектурасының схемасы

Деректер жиынтығы

Деректер жиынтығы 1200 тасымалданатын орындалатын зиянды бағдарламалардан (PE) және 400 жақсы үлгілерден тұрады. Зиянды файлдар екі көзден жиналды. Бірінші дереккөз-VirusTotal репозиторийі, ол жерден 2017 жылдан 2019 жылға дейін түсірілген 900 зиянды файл жүктелді. 2015 жылдан 2017 жылға дейін түсірілген зиянды бағдарламалардың қосымша 300 үлгісі Das Mahlwerk-тен жүктелді. Зиянды бағдарламалар туралы мәліметтер жиынтығы зиянды бағдарламалардың келесі отбасыларының тең санын қамтыды: жарнамалық бағдарламалар, ransomware, Keylogger, жүктеуші және артқы жағы. Екі түрлі кезеңдегі зиянды бағдарламаларды екі көзден жинаудың мақсаты-толық және сенімді мәліметтер жиынтығы, өйткені зиянды бағдарламалар тактикасы мен мінез-құлқын өзгертеді. Екінші жағынан, жақсы файлдар Windows 7 операциялық жүйесінің файлдарынан жиналды (Win 7-нің 32 биттік нұсқасы)

Ортаны орнату

Талдау ортасы Ubuntu-мен жұмыс істейтін компьютерді негізгі операциялық жүйе ретінде және Windows 7-ді бір гигабайт жедел жадта жұмыс істейтін қонақ жүйесі ретінде қамтыды. Хост жүйесі Cuckoo Sandbox, VirtualBox және Volatility tool сияқты басқа да қажетті бағдарламаларды іске қосуға жауапты болды. Cuckoo құмсалғыш алдымен конфигурацияланды, содан кейін екі мақсатта қолданылды: тексерілген файлдың әрекетін талдау және әр файлдың соңында жадыны қалпына келтіру арқылы жад кескінін алу

Орындау Процесі

Орындау процесі екі кезеңнен тұрды. (1) динамикалық талдау. Cuckoo құмсалғыш қонақтар жүйесінде іске қосылған кезде әр файлдың әрекетін бақылау үшін пайдаланылды, содан кейін процесс аяқталғаннан кейін Java server object notation (JSON) форматында мінез-құлық туралы есеп жасады. (2) жадты талдау. Осы кезде Volatility құралы жадтың әр кескінін

талдау және JSON форматында жадты талдау есебін жасау үшін қолданылды. Орындау процесінің соңында екі есеп жасалды: әр тексерілген үлгі үшін мінез-құлық және жад есептері. Осылайша, жасалған есептердің жалпы саны бүкіл деректер жиынтығы үшін 3200 болды.

Белгілерді алу

Содан кейін API қоңырау функциясы мінез-құлық және жад есептерінен алынды. Cuckoo құмсалғышымен жасалған мінез-құлық туралы есепте басқа ақпаратқа (журнал файлы) қосымша орындалған API қоңыраулары болды. Осылайша, API қоңыраулары әр мінез-құлық туралы есептерден тікелей алынды.

Алайда, жадта API функциясының қоңыраулары импорт мекен-жайы кестесінде (IAT) болды. Осылайша, Volatility құралынан Imps can пәрмені жад кескінінен API функциясының қоңырауларын алу үшін қолданылды. Imps can командасы IAT кестесіндегі API қоңырауларын, сондай-ақ басқа да байланысты ақпаратты іздеуде жад кескінін сканерлейді Жад кескіндерінен алынған нысандар саны 4705 болса, динамикалық талдау нәтижесінде алынған нысандар саны 4280 нысанды құрады

Алдын ала модельдеу

Осы кезде API функциясының сапасын жақсарту үшін функцияларды әзірлеу әдістері қолданылды. Жадтан да, динамикалық талдаудан да алынған функциялар біріктірілді.

Функциялардың жалпы саны 8985 функцияны құрады. Содан кейін қайталанатын функциялар тізімнен алынып тасталды. Осылайша, екі анализдегі ерекше белгілердің жалпы саны 6270 белгіні құрады. Қарапайым есептеулерді қолдана отырып, мінез-құлық белгілерінің саны 4280 құрады, ал екі анализде де анықталмаған белгілердің жалпы саны 8985 болды. Бұл дегеніміз, жадта кем дегенде 2715 тәуелсіз функция табылған, бірақ динамикалық талдау кезінде зиянды файлдарды орындау кезінде емес. Тәуелсіз белгілер сәтті анықтау мен жіктеу процесінің жақсы белгісі болып табылады. Осыдан кейін тізім алфавиттік ретпен реттелді. Барлық жеке нысандарды қамтитын жаңа тізім жаһандық тізім (немесе жаһандық вектор) ретінде қарастырылды. Сол сияқты, әр зиянды және пайдалы файл үшін бірдей процедура қолданылды. Алайда, жадты талдаудан да, мінез-құлықты талдаудан да алынған объектілерден тұратын әр файлдың тізімі жергілікті тізім (немесе жергілікті вектор) ретінде қарастырылды.

Осылайша, бір ғаламдық тізім және 1600 жергілікті тізім алынды. Келесі қадамда әр жергілікті тізім екілік векторға айналды, онда 1 Жергілікті тізімдегі объектінің Ғаламдық тізімге енгізілгенін және егер ол болмаса 0 болды. Ұқсас үлгілер (даналар) жіктеу процесінің дәлдігін арттыру мақсатында алынып тасталды.

Сонымен қатар, Python сценарийі мінез-құлық есептерінен де, жад есептерінен де (объектілерді шығару) API қоңырауларын шығаруға, сонымен қатар жергілікті векторлар мен ғаламдық векторларды (алдын-ала модельдеу) құруға арналған. Сценарий төрт негізгі қадамды орындайды

1-ші қадам

Деректер жиынындағы әр файл үшін келесі әрекеттерді орындаңыз:

– Динамикалық талдау жасау.
– Журнал файлынан API қоңырауларын шығарыңыз.
– Ғаламдық динамикалық тізімге API қоңырауларын қосу

2-ші қадам

Әрбір файлды деректерді теру орындаңыз:

Жад талдауын жасаңыз.

– Жад кескінінен API қоңырауларын шығарыңыз.
– Ғаламдық жад тізіміне API қоңырауларын қосу

3-ші қадам

Көмегімен векторлардың Ғаламдық тізімін жасаңыз:

– Ғаламдық динамикалық тізімді Ғаламдық жад тізімімен біріктіру арқылы ғаламдық тізімді жасаңыз.
– Жаһандық тізімнен ұқсас API қоңырауларын жойыңыз.
– Ғаламдық тізімді сұрыптау

4-ші қадам

Деректер жиынындағы әр файл үшін келесі әрекеттерді орындаңыз:

– 1-қадамнан және 2-қадамнан API қоңырауларына қосылу арқылы жергілікті тізімді жасаңыз.

– Жергілікті тізімдегі ұқсас API қоңырауларын жойыңыз.

– Жергілікті тізімді сұрыптау.

– Жергілікті екілік қорап векторы

Біріктірілген вектордың тиімділігі жіктеу процесінде тексеріледі

Қорытынды

Бұл зерттеу соңғы пайдаланушының компьютеріндегі зиянды файлдарды жіктеу кезінде жүзеге асырылуы мүмкін тиімді және сенімді тәсілді жасады. Бұл зерттеудің мақсаты зиянды бағдарламаларды анықтау дәлдігін арттыру және жалған позитивтердің жиілігін төмендету болып табылады.

Тәжірибелер негізінде динамикалық талдаудан алынған нысандарға жадтан көбірек байланысты нысандарды қосу анықтау дәлдігін арттыратыны дәлелденді. Бұл жұмыста тек API қоңырау функциялары жад кескіндерінен және динамикалық талдаудан алынып, жіктеу процесінде қолданылды. API қоңыраулары жоғары жіктеу нәтижесіне қол жеткізгенімен. Алайда, болашақ жұмыста ұсынылған тәсілді жақсарту үшін тізілім және желі функциялары сияқты басқа да функцияларды қарастыру қажет. Ұсынылған тәсілді деректер жиынтығындағы зиянды бағдарламалар санын көбейту арқылы жақсартуға болады. Сонымен қатар, зиянды бағдарламалардан, криминалистикадан және виртуализациядан қорғау әдістерінен қорғау үшін оқшауланған ортаның мүмкіндіктерін кеңейту бойынша қосымша жұмыстар қажет.

Қолданылған әдебиеттер тізімі

1. Ж. Р. Сюэ, Ж. В. Фанг және П. Чжан, "Автономды жүргізудегі оқиғалар туралы пайымдау арқылы көріністі түсінуге шолу", халықаралық Автоматтандыру және есептеу журналы, 15-Том, № 3, 1-18 беттер, 2018 ж.

2. AV-TEST. AV TEST The Independent IT Security Institute. 2022. Интернетте қол жетімді: https://www.av-test.org/

3. Сивейл Р.; Омар к.; Ариффин К. Зиянды бағдарламаларды талдау әдістеріне шолу: статикалық. динамикалық. Гибридті 2018, 8, 1662-1671.

4. Virus Total. 2022. Интернетте қол жетімді: https://www.virustotal.com/#/home

5. Э. Рафф, Р. Зак, Р. Кокс, Дж. Сильвестер, П. Ячи, Р. Уорд және басқалар, "Зиянды бағдарламаларды жіктеуге арналған байт N-граммның ерекшеліктерін зерттеу", Компьютерлік вирусология және бұзу әдістері журналы, 14-Том, № 1, 1-20 беттер, 2018

6. Р. Сочер, А. Перелигин, Дж. Ву, Дж. Чуанг, С. Д. Маннинг, А. Нг және т.б., «Сезім ағашының банкі бойынша семантикалық композициялық үшін рекурсивті терең үлгілер», 2013 жылы эмпирикалық әдістер бойынша конференция материалдарында. 1631-1642 б., 2013 ж