МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

КАФЕДРА КОМП'ЮТЕРНИХ НАУК

ВИПУСКНА РОБОТА

на тему:

«Програмний комплекс захисту мережевого периметру інформаційно-комунікаційної системи підприємства»

Завідувач випускаючої кафедри — **Довбиш А.С.**

Керівник роботи — **Кальченко В.В.**

Студента групи КБ – 71 — **Гура Д.Ю.**

СУМИ 2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую_____

Зав. кафедрою Довбиш А.С.

"_____" _____2021 р.

ЗАВДАННЯ

до випускної роботи

Студента четвертого курсу, групи КБ-71 спеціальності "Кібербезпека" денної форми навчання Гури Дениса Юрійовича.

Тема: "**Програмний комплекс захисту мережевого периметру інформаційно-комунікаційної системи підприємства**"

Затверджена наказом по СумДУ

№_____ від _____ 2021 р.

Зміст пояснювальної записки: 1) Дослідження алгоритмів роботи та відмінностей IDS / IPS систем; 2) аналітичний огляд існуючих систем захисту мережевого периметру; 3) розробка скрипту для розгортання та налаштування системи Snort на комп'ютері користувача;

Дата видачі завдання "_____"_____2021 р.

Керівник випускної роботи _____ Кальченко В.В.

Завдання прийняв до виконання _____ Гура Д.Ю.

РЕФЕРАТ

Записка: 71 стор., 17 рис., 2 табл., 2 додатки, 15 джерел.

Об’єкт дослідження — системи виявлення та запобігання вторгненням.

Мета роботи — розробка скрипту для швидкого, автоматичного розгортання системи запобігання вторгненням на комп’ютері користувача, а також здійснення базових налаштувань системи.

Методи дослідження — метод аналітичного порівняння існуючих систем.

Результати — розроблено скрипт для розгортання та базових налаштувань системи запобігання вторгненням Snort. При цьому майже всі дії виконуються автоматично та не потребують втручання з боку людини. Розроблений скрипт реалізовано за допомогою команд командного рядка.

СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕННЯ, IDS / IPS СИСТЕМИ,
ЗАХИСТ МЕРЕЖЕВОГО ПЕРИМЕТРУ ПІДПРИЄМСТВА,
СКРИПТ, КОМП’ЮТЕРНА МЕРЕЖА, ЗАГРОЗА, АНОМАЛІЯ,
СИГНАТУРА, ІНФОРМАЦІЙНА БЕЗПЕКА.

ЗМІСТ

ВСТУП

Невід’ємною частиною успішності чи не кожного сучасного підприємства є комерційна таємниця. Збереження секрету фінансового успіху здавна було однією з пріоритетних цілей будь якого виробництва чи мануфактури. У зв’язку з массовою цифровізацією та комп’ютеризацією пріоритет захисту комерційної таємниці тільки зростає, адже мережа підприємства містить в собі велику кількість конфіденційної інформації: договори, контракти, персональні дані працівників, секретні рецептури і тд. Основною метою даної роботи є розгляд існуючих систем захисту мережевого периметру комп’ютерних мереж та способи їх розгортання в мережі підприємства.

Для впровадження належного рівня мережевого захисту використовують системи виявлення та запобігання вторгненням IDS / IPS(IDS - intrusion detection system & IPS - intrusion prevention system) – це комплекс програмних або апаратних засобів, які виявляють факти і запобігають спроби несанкціонованого доступу в корпоративну систему. Серед заходів, які приймаються для досягнення ключових цілей IDS / IPS, можна виділити інформування фахівців з інформаційної безпеки про факти спроб хакерських атак і впровадження шкідливих програм, обрив з'єднання зі зловмисниками і переналаштування мережевого екрану для блокування доступу до корпоративних даних. Різниця між IDS та IPS полягає в тому, що IDS - це система виявлення вторгнень в той час, як IPS окрім відслідковування активності в реальному часі ще й швидко реалізує дії щодо запобіганню атак.[14]

Перші два типи систем IDS / IPS з'явилися в 1986 році як результат наукової роботи, і їх базові принципи досі використовуються всюди - в системах запобігання та виявлення. Системи виявлення загроз і системи запобігання загрозам почали з'являтися після написання наукової статті Дороті Деннінг, і називалася ця стаття «Модель виявлення загроз», і завдяки ній Стенфордський Дослідницький Інститут розробив систему під назвою Intrusion Detection Expert

System / (IDES). Вільно це можна перевести як експертна система виявлення загроз. Вона використовувала статистичне виявлення аномалій, сигнатури і хостові/користувацькі профілі для детектування підозрілої поведінки у систем. Таким чином, вона могла визначити якщо такі протоколи як FTP або HTTP були використані некоректно і навіть могла визначати атаки з відмовою обслуговування (DoS).

На початку 2000-х системи виявлення IDS були досить популярними. Фаєрволи обробляли трафік відносно швидко, так як в них не було глибокої інспекції пакетів, тобто ви не знали, що це за трафік приходить в мережу - фаєрволи реагували тільки на встановлені в правилах, протоколи та мережеві адреси. В той же час почали з'явилися нові атаки, такі як SQL-ін'єкції і інші, і вони моментально стали розповсюджуватися. Саме на цьому етапі IDS системи і стали в нагоді. Інша ситуація склалася з IPS системами: деякі організації не використовували IPS системи через те що вони потенційно могли заблокувати звичайний трафік. Таким чином, IDS системи просто повідомляли про таку аномалії і нічого не блокували, щоб системний адміністратор міг зреагувати і перевірити - чи правда це щось небезпечне або ж це просто аномалія. З цієї причини в той час ринок для систем запобігання загрозам був настільки малий, що існувало всього кілька IPS вендорів.[22] В цей час сигнатури писалися для виявлення експлойтів, але не вразливостей - тобто для кожної уразливості було 100 різних способів експлойта. Але така система здійснювала негативний вплив на продуктивності. Вже з 2005 IPS системи починають набирати популярність. Особливою популярністю користувалися гібридні системи, вони ефективно суміщали в собі IDS та IPS, а продуктивність таких систем становила до 5 Гбіт/с. Такі системи могли моніторити сегментовані мережі, DMZ, серверні ферми з веб-додатками і площу всередині периметра. Сучасні гібридні системи мають швидкість до 40 Гбіт/с.

Як було зазначено вище системи попередження проникнення отримали стрімкий розвиток, часто він стимулювався потребами ринку або проблемами і небезпеками, які з’являлися. Наприклад, коли з'явився стандарт безпеки PCI DSS ринок користувачів почав вимагати від організацій підтримку оплати картами, встановлення IDS, або міжмережевих екранів з можливістю фільтрації веб-додатків. В 2010-х був переломний момент для вендорів в сфері ІБ(інформаційної безпеки) - так як вони стали випускати системи запобігання загроз наступного покоління, які включали в себе такі як контроль користувачів і додатків. Таким чином, традиційний IPS дивиться в мережевий трафік на предмет відомих атак і здійснює операції над цим трафіком, в залежності від моделі розгортання, а IPS наступного покоління робить те ж саме, але крім того він покриває набагато більше протоколів (аж до 7 рівня) для захисту від більшої кількості атак.[1]

Сьогодні більшість організацій використовують NGFW і список їх функцій тільки зростає. Так як ці міжмережеві екрани відрізняються різними особливостями, організаціям доведеться вибирати в залежності від точності поставленого завдання і їх вимог. Таким чином з кожним роком IDS / IPS системи все більше розвиваються та вдосконалюються, для їх роботи застосовуються нові технології та розробки, але принцип роботи та головна функція залишається незмінною: аналіз трафіка та попередження атак на мережу офісу, виробництва чи держустанов.

1. ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

1.1. Виникнення і розвиток IDS / IPS систем

На початку розвитку комп’ютерів та комп’ютерних мереж єдиним інструментом для виявлення вторгнень були самі системні адміністратори, які вручну аналізували, через консоль, дії користувачів. До таких дій належать: підозріла активність принтера, що використовується нечасто, локальний вхід користувача у систему в той час, як він у відпустці. Незважаючи на ефективність дана система втратила актуальність через орієнтацію на певні попередньо визначені ситуації і неможливість масштабувати її в великій мережі.

Наступним кроком розвитку систем детектування загроз стала поява журналів реєстрації подій, їх переглядали адміністратори мережі, таким чином вони знаходили підозрілі дій. В середині другої половини двадцятого століття журнали реєстрації подій являли собою перфоровані карти, значним недоліком такої системи реєстрації було те, що за тиждень роботи кількість карт була такою, що в одному місці вона займала висоту метр чи два. В результаті цього будь який пошук інформації відбирав багато часу. Таким чином можливість виявити атаку на комп’ютерну мережу в режимі реального часу було неможливим. Перфоровані карти і записані на них журнали реєстрації подуй використовувалися виключно для того, щоб підтвердити спробу чи факт атаки на систему.

Ситуація полегшилася коли почалося зниження цін на дискову пам'ять, це дозволило вести журнали в електронному вигляді. Такі зміни спровокували появу перших програм для аналізу отриманих даних. Але були і певні складнощі: аналіз отриманих даних був неефективним через повільність їх обробки та необхідність задіяння великої кількості апаратних ресурсів. Запускали програми для виявлення вторгнень в той час коли кількість користувачів була мінімальна або вночі. Навіть за такої системи виявлення вторгнень найчастіше відбувалося вже після проведеної атак і тільки констатувало її факт.[13]

В кінці другої половини двадцятого століття з’явилися перші системи які виявляли вторгнення в режимі реального часу. Вони аналізували події відразу після їх додавання в журнал реєстрації. Такий прогрес дозволив оперативно інформувати адміністраторів про атаки та дало можливість оперативно приймати контр міри.

Першим хто створив ефективну систему виявлення вторгнень (СВВ) був Джеймс Андерсон. Наступною була Дороті Деннінг, в 1986 році вона з допомогою Пітера Неймана представила власну СВВ, її робота лягла в основу більшості сучасних систем.[6] Розроблена нею модель використовувала методи статистики для виявлення загроз, вона називалася IDES (Intrusion detection expert system - експертна система виявлення вторгнень). Система здійснювала аналіз трафіку мережі та даних користувацьких додатків.

Для виявлення вторгнень створена IDES використовувала два методи. Першим була експертна система, яка використовувала базу з вже відомими типами вторгнень, а другим - сам модуль, який здійснював моніторинг системи з використанням статистичних методів, даних користувачів та системи, що охороняється. Наступним кроком у розвитку систем виявлення вторгнень було використання штучних нейронних мереж в якості третього компоненту. Дане нововведення було запропоноване Терезою Лунт, в результаті цього в 1993 році була розроблена NIDES (Next-generation Intrusion Detection Expert System - експертна система виявлення вторгнень нового покоління), яка засновувалася на її ідеях.[7]

В той же час відбувається розвиток інших СВВ, які використовували схожі чи відмінні методи виявлення атак на мережу. Серед таких буда система MIDAS (Multics intrusion detection and alerting system), вона базувалася на принципах, які описали Деннінг і Нейман. Того ж 1988 року з’явилася система Haystack.

Система TIM (Time-based inductive machine) розроблена в 1990, комбінувала детектування підозрілої активності та індуктивного навчання, що базувалося на послідовних патерах.

На початку 2000-х системи виявлення IDS були досить популярними. Фаєрволи обробляли трафік відносно швидко, так як в них не було глибокої інспекції пакетів, тобто ви не знали, що це за трафік приходить в мережу - фаєрволи реагували тільки на встановлені в правилах, протоколи та мережеві адреси. В той же час почали з'явилися нові атаки, такі як SQL-ін'єкції і інші, і вони моментально стали розповсюджуватися. Саме на цьому етапі IDS системи і стали в нагоді. Інша ситуація склалася з IPS системами: деякі організації не використовували IPS системи через те що вони потенційно могли заблокувати звичайний трафік.

Таким чином, IDS системи просто повідомляли про таку аномалії і нічого не блокували, щоб системний адміністратор міг зреагувати і перевірити - чи правда це щось небезпечне або ж це просто аномалія. З цієї причини в той час ринок для систем запобігання загрозам був настільки малий, що існувало всього кілька IPS вендорів. В цей час сигнатури писалися для виявлення експлойтів, але не вразливостей - тобто для кожної уразливості було 100 різних способів експлойта.[18] Але така система здійснювала негативний вплив на продуктивності. Вже з 2005 IPS системи починають набирати популярність. Особливою популярністю користувалися гібридні системи, вони ефективно суміщали в собі IDS та IPS, а продуктивність таких систем становила до 5 Гбіт/с. Такі системи могли моніторити сегментовані мережі, DMZ, серверні ферми з веб-додатками і площу всередині периметра. Сучасні гібридні системи мають швидкість до 40 Гбіт/с.

Як було зазначено вище системи попередження проникнення отримали стрімкий розвиток, часто він стимулювався потребами ринку або проблемами і небезпеками, які з’являлися. Наприклад, коли з'явився стандарт безпеки PCI DSS

ринок користувачів почав вимагати від організацій підтримку оплати картами, встановлення IDS, або міжмережевих екранів з можливістю фільтрації веб-додатків. В 2010-х був переломний момент для вендорів в сфері ІБ(інформаційної безпеки) - так як вони стали випускати системи запобігання загроз наступного покоління, які включали в себе такі як контроль користувачів і додатків. Таким чином, традиційний IPS дивиться в мережевий трафік на предмет відомих атак і здійснює операції над цим трафіком, в залежності від моделі розгортання, а IPS наступного покоління робить те ж саме, але крім того він покриває набагато більше протоколів (аж до 7 рівня) для захисту від більшої кількості атак.

Сьогодні більшість організацій використовують NGFW і список їх функцій тільки зростає. Так як ці міжмережеві екрани відрізняються різними особливостями, організаціям доведеться вибирати в залежності від точності поставленого завдання і їх вимог.

1.2. Загальні відомості про IDS

Системи виявлення вторгнень – невід’ємна частина системи безпеки комп’ютерної мережі підприємства чи установи. Безпека інформаційної мережі в сучасних реаліях є надзвичайно важливою, через те, що велика кількість бізнес процесів та інформації знаходиться в електронному вигляді.[8]

Система виявлення вторгнень – це комплекс різних програмних та апаратних ресурсів направлених на одну мету – аналіз підконтрольної системи, виявлення аномалій чи підозрілих дій в системі та своєчасне попередження про це.

В залежності від методів, що використовуються, дій які приймаються та інших факторів можна здійснити певну класифікацію IDS систем(рисунок 1.1.) [10].

Метод виявлення може бути поведінковим чи інтелектуальним. Вразі, коли робота IDS ґрунтується на поведінці системи, яка контролюється за нормальних умов, то така система є поведінковою. Якщо IDS система за основу нормальної роботи бере дані про атаки то така система є інтелектуальною.

Рисунок 1.1. - Характеристики систем виявлення вторгнень

Дії направлені на виявлену загрозу показують відповідь IDS на атаку. Якщо IDS приймає певні дії то вона є активною. Втому випадку коли система просто інформує про можливу загрозу її називають пасивною.

Також, серед факторів, які впливають на класифікацію IDS є вид інформації, що буле проаналізована. Результати аудиту, мережеві пакети, файли реєстру можуть належати до вхідних даних.[8]

Режим роботи IDS також впливає на класифікацію. Є такі системи виявлення вторгнень, які здійснюють безперервний моніторинг системи, а є ті, які роблять періодичні перевірки системи.

Серед можливих способів класифікації СВВ можна поділити за способом реагування, збору інформації та методу аналізу. [10] Способів реагування наразі налічується два: статичний і динамічний. Статичні IDS роблять «відбитки» підконтрольної мережі та її середовища і на підставі цих даних здійснюють

перевірку, пошук можливих вразливостей та помилок в програмному забезпеченні, слабких паролей і тд. Також вони знаходять сліди можливих атак. На відміну від статичних IDS, динамічні проводять моніторинг системи в режимі реального часу, вони контролюють всі системні процеси, трафік в мережі. Вони допомагають оперативно дізнатися про можливу атаку.

Рисунок 1.2. - Класифікація систем виявлення вторгнень

При здійсненні класифікації за методом збору інформації системи виявлення вторгнень діляться на мережеві та системні. Мережеві (NIDS) здійснюють аналіз пакетів в мережі підприємства і на ранніх етапах виявляють спроби проникнення чи атак. Такі IDS обробляють весь потік даних мережі. Одним з вагомих переваг мережевих СВВ є те, що вони можуть бути встановлені на якомусь одному пристрої в мережі та контролювати повністю весь трафік мережі. Системи, які контролюють тільки один пристрій, та ведуть моніторинг можливих підозрілих дій чи аномалій називають системними IDS. До таких систем виявлення належать ті, які здійснюють перевірку системних файлів і тд.[19]

Відповідно до методу аналізу існує дві групи IDS систем: СВВ, які мають вже встановлену базу з даними про сигнатури різних видів і типів атак і

відповідно до неї порівнюють інформацію. Другий тип IDS – це ті, які контролюють частоту виникнення певних подій чи пошук аномалій.

Одним з перших методів виявлення загроз був аналіз сигнатур. Його методика роботи заснована на понятті збігу певної послідовності з вже існуючим прикладом. У пакеті, що перевіряється порівнюється кожний байт з існуючою сигнатурою. В результаті перевірки, якщо буде виявлено збіг послідовності з вказаними в базі даних, буде повідомлення про можливу загрозу.

Інший метод аналізу базується на аналізі певних протоколів через те, що дані, які вони містять мають строго визначену форму і стандартизовані. Всі пакети, які передаються певними протоколами містять різні поля, які мають бути заповнені нормальними значеннями. Розробники IDS створили інструмент, який перевіряє відповідність значень цих полів зі стандартами. В разі певних невідповідностей між отриманими даними і стандартом адміністраторам мережі надходить повідомлення про підозрілий трафік.

Системи, що здійснюють аналіз сигнатур володіють певними перевагами над іншими. Вони досить продуктивні та швидкі, правила для них можна досить легко написати, самі системи легко і швидко налаштовуються. В разі виявлення нових типів загроз і небезпек велику частку в створенні нових сигнатур має інтернет спільнота. Завдяки цьому виду СВВ пошук та нейтралізація хакерів значно спрощується та прискорюється.[15] Все це можливе через те, що спробу атаки можна виявити ще на ранніх етапах, так як більшість простих атак без особливих складнощів ідентифікуються системою по сигнатурах.

Незважаючи на всі сильні сторони IDS, що працюють тільки завдяки аналізу сигнатур, не позбавлені недоліків. Одним з них є те, що надзвичайна швидкість роботи в початковий період роботи системи поступово зменшується. Цей процес відбувається через те, що поступово збільшується кількість сигнатур, які перевіряються. Кожна нова атака, підозріла дія чи аномалія провокують створення нових сигнатур.

Іншою, не менш важливою проблемою є те, що система виявлення вторгнень порівнює дані пакета виключно з існуючою базою сигнатур. Сигнатур нових атак в неї немає і тому вона може їх пропустити.

Незважаючи на даний факт слід зазначити що близько вісімдесяти відсотків атак проходять за вже вивченими сценаріями, а присутність в базі сигнатур відбитків відомих атак значно підвищує шанси виявлення загрози.

Використання аналізу протоколів також містить як позитивні моменти так і певні недоліки. Ретельна перевірка протоколів вимагається пре процесами, саме через це аналіз протоколів є досить нешвидким. Також, суттєвим недоліком є важкість написання правил перевірки трафіку за застосування такого методу. Не буде перебільшенням якщо скажемо, що правильність написання і їх працездатність повністю залежать від розробника.

З одного боку системи виявлення вторгнень, які засновані на аналізі протоколу мають меншу швидкість роботи у порівнянні з тими, що працюють на основі сигнатури, проте вони дають можливість отримати більш точні і об’ємні результати. Також завдяки таким системам можна знаходити найновіші «загрози нульового дня» та інше.

Якщо розглядати архітектуру систем виявлення вторгнень то в цьому випадку їх можна розділити на дві групи: локальну і глобальну. Локальна архітектура в своїх межах реалізує елементарні компоненти, що згодом можна об’єднати для забезпечення правильної роботи мереж [9].

Головні компоненти локальної архітектури та їх взаємозв’язки можна побачити на рисунку 3. Сенсорами називаються агенти, які роблять початковий збір необхідних даних. Вся необхідна реєстраційна інформація отримується з системних чи прикладних журналів, також можливий варіант коли шукана інформація отримується безпосередньо з мережі або можливе перехоплення

інформаційних пакетів. Дані маніпуляції можливі завдяки спеціальним службам та режиму моніторингу мережі.

Рисунок 1.3. - Основні елементи локальної архітектури систем виявлення вторгнень

Зменшення обсягу вхідних даних можливе завдяки їх попередній фільтрації на першому рівні, рівні сенсорів.

Інформація від агентів надходить до центру розподілу, який в свою чергу уніфікує її, доводить до визначеного формату.[5] В комплекс функцій центру розподілу також входить: фільтрація, занесення інформації в базу, відправка її на аналіз іншим компонентам. До одного центру може надходити інформація з декількох сенсорів.

Наступні два компоненти: статистичний і експертний здійснюють активну фазу аудиту. В тому випадку якщо під час процесу аналізу даними компонентами буде виявлено факт підозрілих дій буде сформовано повідомлення та направлене на вирішувач, він визначає чи реально існує загроза проникнення чи це було помилкове спрацювання системи.

Якісна система виявлення вторгнень під час кожного повідомлення про можливу загрозу має дати детальні пояснення, що до причин такої поведінки, а також список рекомендованих заходів. І в тому випадку коли адміністратору надається можливість робити вибір відносно певних дій, то це має бути лаконічне меню.[5]

В поняття «глобальна архітектура» закладено можливість утворення короткотермінових і різнорівневих зав'язків, які засновані на локальних СВВ (рисунок 4).

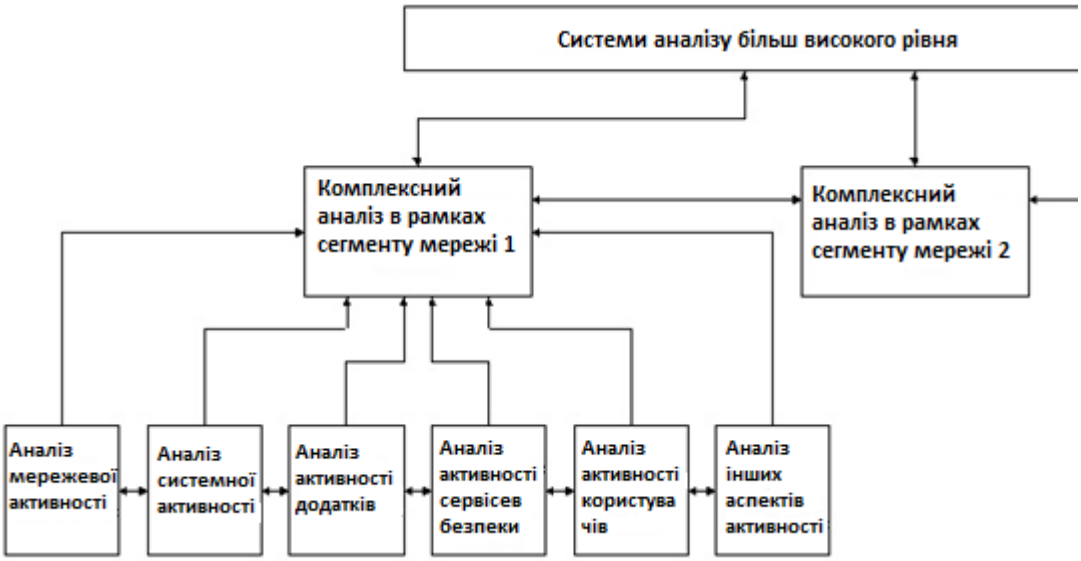

Рисунок 1.4. - Глобальна архітектура систем виявлення вторгнень

Особливістю даного типу архітектури є те, що підозрілі дії аналізуються з різної точки зору різними компонентами, які розташовані на одному рівні.[19] В тому випадку коли один з компонентів виявить небезпеку він повідомляє інші компоненти на цьому ж рівні, в результаті чого буде відповідна відповідь системи.

Використання різнорангових зав'язків дає можливість підсумувати результати аналізу та зрозуміти повністю всі процеси, які відбуваються. Також, в певних випадках локальний компонент не володіє необхідною інформацією для того, щоб повідомити про загрозу, але «спільне» рішення різних компонентів на

основі проаналізованих даних можуть дати підстави для зменшення чи підвищення рівня тривоги.

1.3. Загальні відомості про IPS

Система запобігання вторгнень виявляє (СЗВ) спроби атак, підозрілу активність та може приймати відповідні міри для їх усунення. СЗВ може мати як програмну чи апаратну реалізацію так і програмно-апаратну.

Системи IPS мають схожий набір функціоналу, що і системи виявлення вторгнень так як головна мета у цих двох типів систем спільна. Маючи спільний характер різниця між ними все ж є: система запобігання вторгнень в режимі реального часу здійснює перевірку стану мережі та приймає певні дії для нейтралізації загрози.

Порівняння принципів роботи системи виявлення вторгнень і системи запобігання вторгнень можна побачити на рисунку 5.

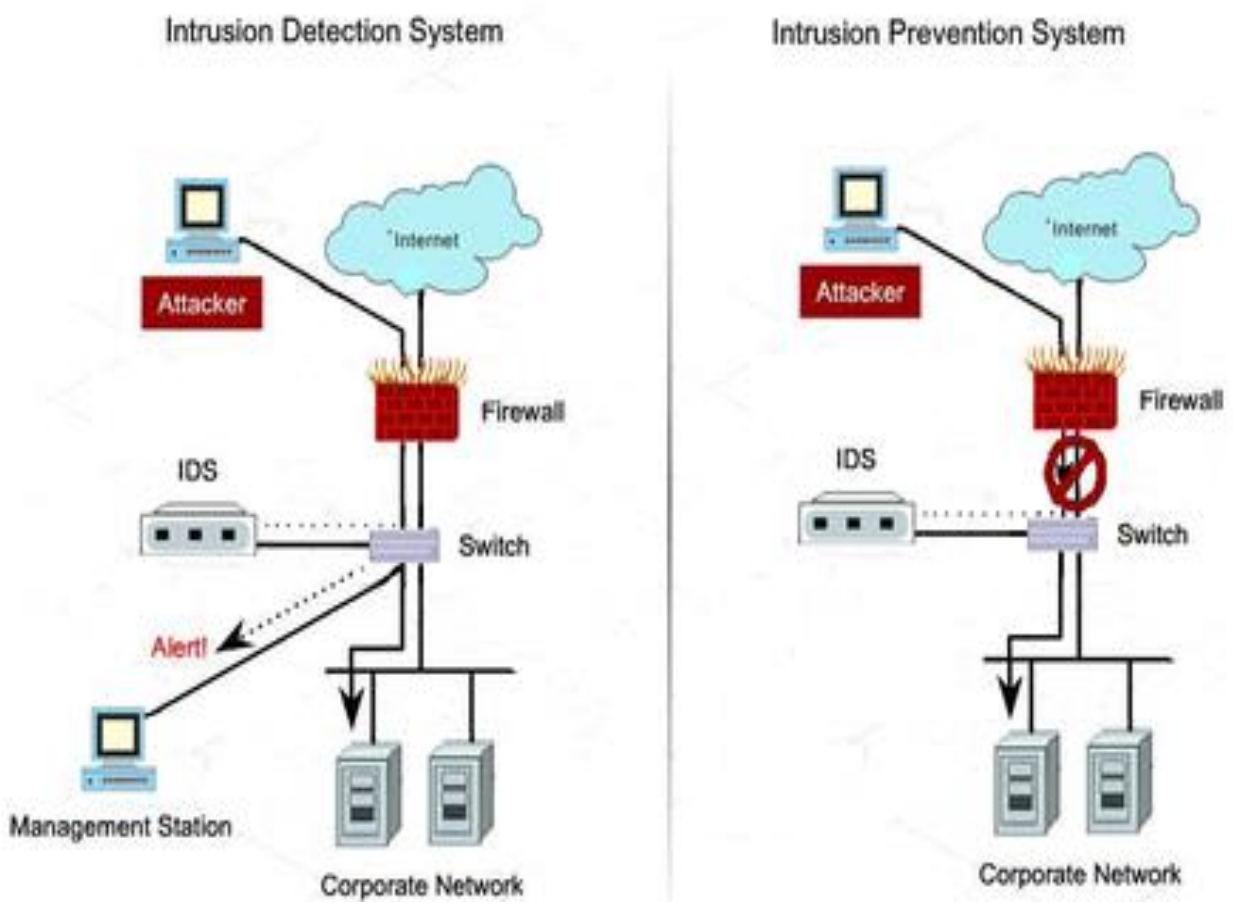

Рисунок 1.5. - Поведінка систем IDS і IPS

Як і у випадку система виявлення вторгнень, СЗВ також мають певну класифікацію:

Мережеві IPS (Network-based Intrusion Prevention, NIPS): здійснюють контроль трафіка в підконтрольній мережі і знешкоджують всю нестандартну активність чи аномалії.

IPS для бездротових мереж називається Wireless Intrusion Prevention Systems (WIPS) її головна мета – аналіз активності в бездротовій мережі. В цей список також можна включити виявлення підозрілих точок для бездротового доступу, які можуть бути направлені для атак чи хакерських дій.

Аналізатор поведінки мережі – NBA (Network Behavior Analysis) контролює трафік в мережі на предмет DoS і DDoS атак.

IPS для окремих комп'ютерів (Host-based Intrusion Prevention, HIPS) програми, що здійснюють контроль на хостовій машині і перевіряють підозрілу активність.

В разі виявлення певної загрози активуються методи для її протидії.

Ві методи задіються тільки після встановлення факту початку атаки. Такий підхід має певний недолік пов'язаний з тим, що навіть при успішній ліквідації загрози певна шкода системі може бути нанесена.

Якщо зловмисник проводить атаку через певні протоколи, наприклад TCP, то відбувається закриття протоколу. В наслідок цих дій зловмисник втрачає з’єднання з мережею, що атакується і атака завершується.

Цей метод має два головних недоліки:

1. Працює виключно з тими протоколами, які потребують попереднього з’єднання;
2. Зазначений метод активується тільки в тому випадку, коли зловмисник зміг здійснити з’єднання.

Якщо система визначить, що певні облікові записи користувачів мережі були скомпрометовані після атаки або через них вона проходила то запускається певний метод, який здійснює блокування датчиками системи. Для виконання цієї процедури всі дії мають виконуватися від імені адміністратора. Саме блокування може носити, як постійний так і тимчасовий характер.

Якщо під час атаки будо виявлено, що вона йшла з певного вузла мережі, то в якості відповідних мір він може бути заблокованим або заблокуються мережеві інтерфейси, їх блокування може бути, як на самому вузлі так і на маршрутизаторі чи комутаторі через які хост здійснює підключення до мережі.[7] Період блокування визначається адміністратором і може відключатися через встановлений час або за команди системного адміністратора.

Певні маніпуляції з вузлом, наприклад: відключення вузла від хостової мережі, перезавантаження обладнання не знімають блокування.

В тому числі заблокувати атаку можна за допомогою брандмауера.

У випадку використання такого способу система запобігання вторгненням створює та надсилає нові конфігурації міжмережевому екрану згідно яких він буде здійснювати фільтрацію трафіка від порушника. Цей процес можна автоматизувати, це досягається завдяки стандартам OPSEC. [12]

Ті міжмережеві екрани, які не працюють з протоколами OPSEC для взаємодії з системою запобігання вторгнення користуються спеціальними модуль-адаптерами:

- на які надходитимуть спеціальні команди для зміни конфігурації міжмережевого екрану.
- які здійснюватимуть редагування конфігурації міжмережевого екрану для вдосконалення його параметрів.

Проведення зміни налаштувань в комунікаційному обладнанні.

Спеціально для протоколу SNMP, IPS аналізує і змінює певні параметри з бази даних MIB (наприклад: таблиці з маршрутизацією, певні налаштування портів) з використанням агенту для цього пристрою, для того щоб здійснити блокування атаки. Також можуть бути задіяні і інші протоколи: TFTP, Telnet і ін.

Проведення активного ліквідації джерела небезпеки.

Описаний метод теоретично можна задіяти в тому випадку коли всі інші методи не будуть мати успіх. Система запобігання вторгненням ідентифікує та здійснює блокування пакетів, що надходять від порушника і в рамках відповідних дій робить контрнаступ на вузол хакера, це можливе за тієї умови, коли відомий точний адрес зловмисника і це не принесе шкоди другим, легальним, вузлам.

Описаний вище метод реалізований в кількох безкоштовних системах:

• NetBuster ефективно протидіє потраплянню на хост «Троянського коня». Він може також бути застосований як спосіб для «fool-the-one-trying-to-NetBus-you» ("обведи довкола пальця тих, хто намагається вломитися до тебе на« Троянського коня »). В такому випадку він відшукує небезпечну програму і вивчає її, а потім він її «повертає» назад, до адресанта.[17]

• Tambu UDP Scrambler оперує з портами UDP. Дане програмне забезпечення може не тільки працювати в якості фіктивного UDP-порту, а й паралізувати роботу обладнання зловмисників.

Через те, що ці методи вимагають виконання великої кількості умов, вони не набули масового розповсюдження.

Частина методів вживає певних дій ще на початку атаки. Як тільки вона почалася і ще не досягла своєї мети. Це допомагає зберегти мережу від серйозних пошкоджень.

Блокування атаки з використанням мережевих датчиків

Для того, щоб проаналізувати всі пакети, які проходять по мережі мережеві датчики встановлюють у розрив каналу зв’язку. Для виконання даних функцій вони оснащені парою мережевих адаптерів, вони працюють в змішаному режимі, тобто здійснюють прийом та передачу. Всі пакети потім записуються в пам'ять буфера, звідти вони зчитуються системним модулем для виявлення атак системи запобігання вторгненням. В тому випадку, коли буде виявлено факт атаки знайдені пакети можуть видалитися. [13]

Для аналізу пакетів застосовують сигнатурний або поведінковий метод, які ефективно справляються з поставленою задачею.

Блокування атак також здійснюється з використанням хостових датчиків.

• Дистанційні атаки здійснюються за допомогою відправки від зловмисника серією підроблених пакетів. Захист здійснюється за допомогою мережевої компоненти системи запобігання вторгнень, яка працює по прикладу мережевих датчиків, але головна відмінність від вищезазначених полягає в тому, що мережева компонента перехоплює і здійснює аналіз пакетів, які надходять з різних рівнів взаємодії, це дає запобігати атакам, які здійснюються по криптозахищеним IPsec- та SSL / TLS з'єднанням.

• Локальні атаки під час несанкціонованого запуску зловмисником певних програм або здійснення певних дій, які порушують безпеку інформаційного середовища. Відловлюючи системні повідомлення від усіх додатків, проводять їх аналіз і блокують ті повідомлення ,що несуть потенційну небезпеку. [5]

1.4. Порівняння

Отже, розглянувши обидва види систем, можна зробити певний порівняльний аналіз. Системи виявлення вторгнень здатна попереджати про певні аномалії чи підозрілі дії, це дає змогу запобігти нанесенню серйозної шкоди чи зведення її до нуля. На противагу їм стають системи запобігання вторгненням, методи роботи, якими вона користується дають можливість в разі небезпеки приймати певні дії. Слід взяти до уваги той факт, що по своїй суті IPS системи є більш розширеною версією IDS систем, тому вони користуються майже однаковими методами детектування атак.

До особливостей роботи систем запобігання вторгнення відноситься також можливість працювати, як на хостовому рівні і мережевому. Процес запобігання атак здійснюється за рахунок того, що система IPS вбудовується в корпоративну мережу таким чином, що весь трафік проходить через неї.[10] Таким чином здійснюється перевірка всіх пакетів для виявлення аномалій чи загроз.

В загальному, системи запобігання вторгненням за способом аналізу трафіка можна поділити на ті, що здійснюють це за допомогою сигнатур, та ті, що аналізують пакети на основі раніше знайдених аномаліях та вразливостях.

Описаний другий тип систем значно ефективніший у виявленні нових, раніше невідомих, загроз.

Якщо розглядати методи реакції на атаки, то слід зазначити, що їх було створено досить багато але серед них виділяються декілька основних: здійснення переривання з'єднання, що досягається застосуванням брандмауера або TCP-пакета, зміною конфігурації мережевого обладнання, відключення або обмеження прав певного користувача чи вузла мережі.

Враховуючи все вище наведене, можна сказати, що найбільш ефективною є система захисту, яка побудована на об'єднанні можливостей IDS і IPS в єдиний комплекс для захисту мережевого периметру. Таке об'єднання створить потужний міжмережевий екран, який здійснює детальний і повний аналіз пакетів у трафіку та ідентифікує атаки та їх спроби. Слід зазначити, що це лише одна зі сторін захисту. Для досягнення повноти захищеності корпоративної мережі окрім систем виявлення та запобігання вторгненням необхідно також використовувати антивірусний захист, системи блокування небажаного контенту, використання закритих каналів передачі даних та проведення регулярних навчань серед працівників на предмет попередження про безпеку користування мережею інтернет.

На основі вище описаних матеріалів було створено таблицю порівняння IPS / IDS систем:

Таблиця 1.1. - Порівняння IDS та IPS систем

Параметр	IPS	IDS
Повна назва	Система запобігання вторгненням	Система виявлення вторгнень
Режим роботи	Активний – пошук аномалій та їх ліквідація	Пасивний- аналіз трафіка, сповіщення в разі небезпеки

Механізми виявлення	Аналіз на основі сигнатур, перевірка трафіка на основі вже знайомих моделей загроз	Аналіз на основі сигнатур, правил, моделей типових загроз
Розташування	Вбудована в мережу, пропускає весь трафік через себе	Поза діапазоном передачі даних
Реакція на аномалію	Блокування і видалення загрози, інформування про небезпеку системного адміністратора	Надсилає повідомлення про потенціально можливу загрозу
Вплив на продуктивність мережі	Знижує ефективність роботи мережі через процеси постійного моніторингу системою IPS	Не пливає на роботу мережі
Переваги	Виявлення та ліквідація загроз, гнучкість налаштування політики безпеки	Аналіз трафіку мережі на предмет загроз, не впливає на ефективність роботи корпоративної мережі

Отже, дані системи та принципи їх роботи лягли в основу всіх сучасних систем для захисту мережевого периметру корпоративної мережі підприємства. Список описаних функцій постійно розширюється, додаються нові, вдосконалюються алгоритми пошуку загроз та виявлення аномалій.[14] Також, зростає масштаб корпоративних мереж, що збільшує навантаження на існуючі системи і вимагає від них все більшої кількості системних ресурсів.

І хоча розглянуті системи досить ефективні все ж вони мають певні архітектурні недоліки. Це спровокувало появу нового покоління систем

запобігання загрозам NGFW. Особливістю даного міжмережевого екрану є паралельний аналіз трафіка в реальному часі всіма доступними засобами для перевірки. Для цього аналізу дані про трафік перевіряються в пам’яті без попереднього завантаження на жорсткий диск. Також, в процесі аналізу задіяні протоколи 7 рівня OSI.

2. ВИБІР МЕТОДУ РІШЕННЯ

З моменту появи перших IDS / IPS систем пройшло вже досить багато років, тож за цей час було створено велику кількість систем, які мають різну реалізацію, підхід та доступність. На даний момент часу системи IPS / IDS активно розвиваються, щоб зменшити число помилкових спрацьовувань і збільшити ефективність рішення. Результатом можна вважати системи NGIPS (Next Generation Intrution Prevention System), це IPS-системи нового покоління, вони можуть виконувати всі необхідні функції в режимі реального часу. Виконання цих функцій не впливає на продуктивність роботи мережі підприємства. Також, такі системи мають змогу здійснювати моніторинг додатків та використовувати сторонні бази вразливостей.

Розглянемо найпопулярніші на даний момент часу IDS / IPS системи та шляхи їх розгортання в цільовій мережі.

2.1. StoneGate Intrusion Prevention System

Розробник: StoneSoft Corporation.

Web-сторінка : www.stonesoft.com.

Реалізація: програмно-апаратна, образ VMware.

ОС: 32/64-бітові Windows 2k3 / Vista / 7 / 2k8R2, Linux (CentOS, RHEL, SLES).

Ліцензія: комерційна.

В сучасному світі для побудови захисту мережевого периметру використовувати захист, який базується лише на одному фаєрволі недостатньо, так як він не зможе забезпечити гідний рівень захисту від загроз. З роками атаки стають все більш складними і багаторівневими, тому блокування портів не є ефективною мірою захисту.[1] Це провокує необхідність використання все більш складних систем запобігання вторгненню. Слід зазначити, що велика кількість

програмних чи апаратних рішень здатен повністю забезпечити охорону корпоративної мережі. Чимало таких систем можуть знайти та знешкодити лише найпростіші види атак. Принципи їхньої роботи базуються на сигнатурному аналізі, в сигнатурах описано, як може виглядати трафік під час певного виду атаки.

Все описане вище не стосується програмного забезпечення випущеного компанією StoneGate та яке має назву Intrusion Prevention System. Сама система є комплексним рішенням яке використовується для захисту корпоративної мережі і здатна протидіяти великій кількості атак. В даній IPS досить широкий набір функціоналу, програмний комплекс зданий здійснювати шифрування трафіку, фільтрувати веб-частину, забезпечувати захист від DDoS-атак та використання можливих вразливостей ПЗ чи обладнання та ін. До ключових особливостей комплексу StoneGate Intrusion Prevention System відноситься також захист від динамічних технік обходу. Цей тип загроз є досить новим і фахівці компанії вивчивши його створили описану вище систему. Даний тип атак на сьогоднішній день несу досить серйозну загрозу для безпеки корпоративної мережі.

Це рішення розробила одна з фінських компаній, яка займається розробкою товарів у сфері мережевої безпеки для корпоративного користування. Даний програмно-апаратний комплекс реалізує в собі всі необхідні функції для якісного пошуку загроз: IPS, , веб-фільтрація, підтримка зашифрованого трафіку, захист від DDoS- і 0day-атак і т.д. StoneGate IPS окрім функцій виявлення загроз може також приймати певні дії у разі їх виявлення: заблокувати вірус, spyware, певні програми (P2P, IM та інше).[8] Дане рішення також містить в собі веб-фільтрацію для роботи якої використовується постійно обновлювана база сайтів. Захисту від обходу систем безпеки AET (Advanced Evasion Techniques) приділяється особлива увага.

Для розбиття корпоративної мережі на декілька віртуальних сегментів застосовується технологія Transparent Access Control, при її використанні реальна топологія мережі не змінюється, а для кожного з сегментів можна застосувати індивідуальні політики безпеки. Вони створюються в офлайн режимі за допомогою готових шаблонів, що містять типові правила . Після створення політик перевірки трафіку адміністратор мережі їх перевіряє і завантажує на віддалені вузли IPS. Схожі між собою події StoneGate IPS обробляє за принципами, що застосовуються також і в SIM / SIEM, такі дії дозволяють значно полегшити аналіз подій і трафіка.

Особливістю рішень StoneSoft є те, що декілька пристроїв їхнього виробництва можна об’єднати в кластер та інтегрувати з іншими, такими як: StoneGate Firewall / VPN і StoneGate SSL VPN. Це створює єдину екосистему продуктів StoneSoft, яка може управлятися з єдиної консолі управління StoneGate Management Center, що складається з трьох значних компонентів: Log Server , Management Server і Management Client. Завдяки тому, що консоль написана на Java її можна використовувати Windows і Linux. Вона забезпечує доступ до моніторингу мережі в реальному часі та перегляду журналів, а також дозволяє виконувати налаштування роботи IPS та створювати нові правила.

StoneGate IPS доступна не тільки у вигляді апаратного комплексу, а й у вигляді VMware образу, який можна інсталювати на власному обладнанні чи в віртуальній інфраструктурі.

Однією з переваг використання даної системи запобігання вторгненням є можливість об’єднання декількох програмних чи апаратних рішень компанії в єдину систему.[19] Подібна інтеграція здійснюється за допомогою консолі управління, розробленою StoneGate. Це дає змогу більш ефективно побудувати систему захисту та здійснювати управління всім обладнанням, а також конфігурацію чи оновлення.

Даний метод набагато надійніший, ніж побудова периметру захисту на основі обладнання та програмного забезпечення від різних виробників, а також набагато простіше в управлінні і несе певну економію коштів.

Система запобігання вторгненням StoneGate IPS представлена, як у вигляді програмного рішення, так і у вигляді апаратного комплексу. Апаратні комплекси схожі між собою, за виключенням того, що вони розраховані на різну пропускну здатність та масштаб мережі в. якій вони працюють. Програмне рішення створене для роботи на серверному обладнанні різних виробників або роботи у віртуальному середовищі.

Для забезпечення повного захисту корпоративної мережі StoneGate IPS має широкий функціонал.

Розроблена StoneGate система запобігання вторгненням контролює трафік с середині мережі та в разі виявлення підозрілої активності здійснює блокування. Це дозволяє здійснювати боротьбу з загрозами, які можуть надходити з боку корпоративних ПК.

Також, в StoneGate IPS було реалізовано технології для захисту від атак нульового дня. Також система має потужний набір інструментів для боротьби з DDoS-атаками, які спрямовані на корпоративну мережу. Всі ці програмні комплекси забезпечують неперервну роботу мережі.

Серед інших сервісів особливо виділяється система захисту для виявлення технік, якими можуть користуватися зловмисники для того, щоб оминути системи захисту. Дана система працює на приєднанні до процедури інспекції трафіку власного механізму, що здійснює нормалізацію трафіку на всіх рівнях мережі.

Досить розвиненою є система для веб-фільтрації. Дана система дає можливість адміністраторам блокувати для відвідування певні сайти чи веб-ресурси. Вона базується на великій базі сайтів, яка постійно оновлюється, в ній

сайти розподілені за категоріями.[4] Веб-фільтрація бере активну участь у підвищенні безпеки корпоративної мережі, а також опосередковано впливає на збільшення продуктивності роботи працівників, шляхом закриття доступу до ресурсів на, які вони могли відволікатися.

Для того, щоб відповідати всім критеріям і вимогам сучасності StoneGate IPS підтримує протокол IPv6.

Також, розглянута система підтримує протоколи SSL / TLS. Наявність підтримки цих протоколів дає можливість контролювати і аналізувати шифрований трафік, а також блокувати всі атаки, які проходять по протоколу HTTPS.

Спеціальні інструменти дають змогу системним адміністраторам контролювати активність в корпоративній мережі.[6] Завдяки цьому можна продивлюватися, який трафік належить бізнес-додаткам, а який ні. Базуючись на отриманій інформації можна заблокувати цей потік пакетів тим самим зменшивши навантаження на мережу.

Дана система запобігання вторгнення має два режими для роботи: IDS і IPS. В першому з вказаних режимів система просто здійснює моніторинг корпоративної мережі на предмет аномалій чи підозрілої активності. В другому режимі – окрім вище зазначених функцій система також, в разі виявлення загроз приймає активні дії. При інтеграції StoneGate IPS з іншими програмними продуктами компанії можна побудувати складну, комплексну систему для захисту від атак корпоративної мережі.[15]

Обрана нами система має корисну технологію - Transparent Access Control. Використовуючи її, можна зробити поділ великої корпоративної мережі на окремі віртуальні сегменти, варто зазначити, що при цьому фізична топологія не змінюється. Такі дії допомагають вибудувати надійний захист, а також надати кожній з віртуальних груп різні рівні допуску та політики безпеки.

Для обробки всієї вхідної інформації про події, які фіксуються сенсорами системи запобігання вторгненням застосовуються інтелектуальні аналізатори, які потім надсилають інформацію до консолі управління.[21] Задача аналізатора полягає в аналізі трафіка, де він здійснює пошук небезпечних послідовностей та сигнатур, інформація збирається з різних сенсорів та датчиків в одному місці та обробляється. Такий метод дозволяє звести до мінімуму помилкові спрацювання системи та забезпечує всебічну, детальну подій, що в результаті полегшує виконання обов'язків адміністратору.

Досить вагомою перевагою системи запобігання вторгненням, що розглядається є можливість легкого масштабування системи. При потребі, для збільшення ефективності системи в неї можна включати нові вузли та обладнання, яке додається до існуючої системи в якості нового кластеру. Ця технологія дозволяє без нанесення шкоди чи втручання в роботу вже існуючого обладнання розширювати систему.

Користуючись консоллю управління системний адміністратор має можливість здійснювати моніторинг стану всієї системи та її окремих вузлів. Також нею передбачена можливість формувати звіти у вигляді таблиць чи графічних файлів.

Серед інших компонентів присутня також спеціальна підсистема, що дозволяє робити з інцидентами. Існує консоль системного адміністратора, де він має можливість вносити зміни у політику безпеки, налаштовувати періодичність перевірки трафіку, здійснювати управління журналами та досліджувати інциденти, які були зафіксовані.

2.2. IBM Security Network Intrusion Prevention System

Розробник: IBM.

Web-сайт: https://www.ibm.com/ru

Реалізація: програмно-апаратна, образ VMware.

Ліцензія: комерційна.

Розроблена IBM система Security Network Intrusion Prevention System призначена, щоб ліквідувати атаки на корпоративну мережу підприємства та проводити її аудит. Високий результат досягається завдяки спеціально розробленій технології для аналізу протоколів. Використання даної технології забезпечує постійний активний захист мережі підприємства від достатньо великої кількості загроз.[10]

IBM розробила унікальну технологію для аналізу протоколів. На основі своєї технології була розроблена система запобігання атак її особливістю є модульна структура. Основним є Protocol Analysis Module, він поєднує аналізатор поведінки та метод сигнатурного виявлення. При цьому він відрізняє декілька сотень протоколів рівня програм і найпоширеніші формати даних, це дає змогу виявляти небезпечний код. Аналіз трафіка задіє в собі понад три тисячі алгоритмів, понад дві сотні з яких відслідковують DoS. Функціонал вбудованого брандмауера дає можливість контролювати доступ до портів і IP.

Завдяки технології Virtual Patch вдається блокувати віруси ще на етапі їх поширення, а також захищає комп’ютери до оновлення системи захисту. За необхідності сигнатури можуть бути створені адміністраторами. Модуль для контролю за додатками дозволяє за необхідності блокувати їх. Спеціальний модуль DLP відслідковує спроби передачі інформації, що може бути конфіденційною та переміщення даних в корпоративній мережі, це дає змогу відслідкувати потенційну загрозу. Через те, що вразливості веб-додатків є одними з найпоширеніших на даний момент часу продукт IBM оснащений спеціальним модулем, який здійснює захист від найпоширеніших атак.

При виявленні атаки існує декілька варіантів реагування при виявленні атаки чи аномалії: блокування хоста, надсилання попередження, логування трафіку під час атаки, ізоляція вузла мережі. Політику безпеки можна гнучко налаштувати і для кожної окремої IP-адреси або VLAN. Спеціальний режим

дозволяє працювати системі запобігання вторгненням навіть у разі виходу з ладу одного з вузлів. За наявності декількох продуктів від IBM їх можна об’єднати в єдину систему та здійснювати управління зі спеціального центру управління.

Превентивність захисту базується на неперервному відстеженні різнотипових загроз в спеціально розробленому центрі безпеки - GTOC (gtoc.iss.net).

Основні можливості Security Network Intrusion Prevention System:

- Працює з 167 різними протоколами в тому числі з протоколами рівня додатків і форматами даних.
- Для аналізу трафіка в процесі захисту мережі від вразливостей застосовується понад 2500 різних алгоритмів.
- Нова технологія Virtual Patch здійснює захист комп'ютерів до встановлення оновлень.
- Має вбудований режим пасивного моніторингу і два режими встановлення на канал.
- Підтримка не однієї зони безпеки одним пристроєм, в тому числі зони VLAN.
- Присутні вбудовані та зовнішні bypass модулі, що використовуються для безперервної передачі потоку даних через пристрій в випадку системної помилки або відключення енергопостачання.
- Застосування технології FlowSmart.
- Велика кількість способів реагування на системні події включаючи логування пакетів атаки.
- Контроль витоків інформації у даних та в офісних документах, що передаються по пірінгових мереж, службам миттєвих повідомлень, веб пошті та іншим протоколам.

Переваги від використання рішень Security Network Intrusion Prevention System:

• Превентивний захист блокує атаки ще на початкових етапах, це не дає здійснити несанкціонований доступ до об’єктів та ресурсів мережі.

• Результатом постійного аналізу є звіти та архіви подій, які дають повну інформацію про події, які відбуваються в мережі і дозволяють точно відповідати вимогам стандартів безпеки.

2.3. McAfee Network Security Platform 7

Розробник: McAfee Inc.

Web: www.mcafee.com.

Реалізація: програмно-апаратна.

Ліцензія: комерційна.

McAfee Network Security Platform (NSP) є достатньо новою IPS системою, яка в своєму арсеналі містить нові багаторівневі технології, вони є сигнатурними і без-сигнатурними. При аналізі типових ознак і моделей загроз застосовуються інтелектуальні алгоритми, які спрощують роботу та економлять час співробітників служби безпеки. Для більш швидкого і точного усунення небезпеки велика кількість процесів автоматизована.

Розроблена на основі IntruShield IPS система McAfee Network Security Platform 7 отримала всі переваги свого попередника та нові розробки. Створена система в своєму арсеналі налічує широкий асортимент інструментів, що допомагають досліджувати пакети, які передаються в межах мережі підприємства та виділяти небезпечний трафік і оперативно реагувати. Новітня технологія Global Threat Intelligence дозволяє оцінити надійність того чи іншого IP-адресу, посилання чи протоколу на основі даних з декількох сотень вузлів, що розташовані по планеті. Дякуючи цим даним McAfee Network Security Platform може визначати підозрілий трафік, загрози нульового дня та різні види атак, все це зменшує кількість помилкових спрацювань майже до нуля.

Поміж інших система запобігання загрозам McAfee NSP привертає увагу можливістю здійснювати аналіз трафіку між віртуальними машинами, а також між віртуальною машиною та фізичним хостом. Для реалізації даної функції використовується модуль від компанії Reflex Systems. Цей модуль здійснює спостереження за вузлами віртуальної машини та передає інформацію на фізичний хост.[12]

Також McAfee розробила хостову IPS - Host Intrusion Prevention for Desktop, ця система підтримує повний захист персонального комп'ютера, здійснює аналіз з'єднань, трафіку, можливих атак і тд.

В порівнянні з іншими системами для запобігання вторгненням IPS система McAfee NSP надає захист від достатньо серйозних загроз, фільтрує мережевий трафік на предмет загроз чи аномалій, забезпечує пошук загроз «нульового дня» та атак типу DoS і DdoS.

Один модуль містить в собі новітні засоби для запобігання вторгненням та для збору даних про додатки. Отримана інформація щодо погроз автоматично зіставляється з відомостями про використання додатків да даними отриманими в результаті аналізу прикладних протоколів сьомого рівня моделі OSI.

Крім того функція аналізу прикладного трафіку також здійснює збір інформації про користувачів та пристрої. Модулі пошуку аномалій в мережевій поведінці дають змогу знайти підозрілі вузли та користувачів та навіть бот-мережі.

Дякуючи архітектурі Security Conected, дана система запобігання вторгненням має можливість здійснювати обмін даними з іншими продуктами компанії McAfee в режимі реального часу. Це дає можливість здійснювати своєчасне оновлення правил та сигнатур. Така інтеграція є набагато ефективнішою ніж використання обладнання різних виробників з тією ж метою.

McAfee Network Security Platform здатна обробляти дані зі швидкістю до 40 Гбіт / с, вона заснована на достатньо потужній апаратній платформі також вона поєднує необхідне апаратне забезпечення операторського класу та засоби для однопрохідної перевірки трафіку.[18]

McAfee NSP складається з декількох компонентів:

- Network Security Sensor (Сенсор): бере на себе функції IDS / IPS системи, здійснює аналіз трафіку в мережі, попередження та блокування загроз і атак;
- Network Security Manager (Менеджер): даний модуль відповідає за коректне налаштування і управління системою запобігання вторгнень, правильне використання політик, формування звітів та сумісну роботу з іншими програмними продуктами McAfee;
- McAfee Update Server (Сервер Оновлень): відповідає за своєчасне оновлення всіх сигнатур, правил, шаблонів та інших компонентів.

Дана IPS система має ряд певних переваг, що забезпечують надійність і ефективність роботи:

- Комбінація сигнатурних і без-сигнатурних методів виявлення загроз забезпечує максимально повний мережевий захист . Дана система здійснює повний аналіз трафіку, перевірку більше ніж півтори тисячі мережевих протоколів та додатків також блокування шкідливого коду.
- За рахунок багатогранного аналізу, кореляції та обміну даними щодо певних загроз підвищується ефективність всього комплексу безпеки. McAfee Security Connected дозволяє інтегрувати між собою рішення даної компанії в єдину мережу для більш досконалої роботи.
- Завдяки вбудованим механізмам, процесам і інструкціям швидкість виявлення вразливостей мережі, аномалій чи атак значно

зростає. Все це допомагає системним адміністраторам швидко знаходити порушення, а також зводить до мінімуму кількість хибних спрацювань системи.

- Завдяки використанню технологій машинного навчання, використанню евристичних і порогових методів відбувається попередження атак типу DoS і DdoS. Дана IPS передбачає можливість обмеження кількості з'єднань з певними серверами і вузлами.
- Архітектура NSP побудована таким чином, що пропускна здатність жодним чином не залежить від того наскільки складні налаштування. В порівнянні з іншими системами це достатньо високий результат, так як в окремих випадках використання великої кількості складних політик знижує пропускну здатність в половину.[9]
- При перевірці з'єднань SSL досягається надзвичайно висока продуктивність.

Серед великої кількості IDS / IPS систем з комерційною ліцензією є й безкоштовні рішення. Розглянемо деякі з них.

2.4. Suricata

Розробник: OISF (Open Information Security Foundation).

Web: www.openinfosecfoundation.org.

Платформа: програмна.

ОС: Linux, * BSD, Mac OS X, Solaris, Windows / Cygwin.

Ліцензія: GNU GPL.

Компанією OISF в 2010 році було презентована новітня IDS / IPS система на розробку якої витратили три роки, особливість даної системи полягала в тому, що для її роботи спеціально були створені нові методи детектування атак. Як і в більшості систем виявлення атак в Suricata відстеження підозрілих дії і спроб нападу на систему здійснюється на основі правил. Адміністратори системи мать

можливість підключати шаблонні стеки правил або можна писати власні, розробники даної IDS / IPS дали змогу користувачам використовувати правила, що використовуються іншими системами. В ранніх версіях були проблеми з сумісністю між Suricata та наборами з інших проектів.

Розробники системи розробили власний формат правил, хоча він дещо подібний до формату правил Snort. Правило складається з трьох компонентів: дія (pass, drop, reject або alert), заголовок (IP / порт джерела і призначення) і опис (що шукати). Однією з особливостей системи Suricata є можливість перегляду інформації прямо з потоку, що дає змогу ще на початку атаки чи спроби відстежити дані виявити небезпеку та прийняти відповідні міри. Така система є більш ефективною ніж у Snort.[17]

З огляду на всі вище сказані особливості Suricata можна зробити висновок, що дана система є набагато швидша та ефективніша ніж подібні їй. Серед явних мінусів проекту – невелика кількість документації, що робить процес налаштування та використання системи недосвідченими користувачами важким.

2.5. Samhain

Розробник: Samhain Labs.

Web: www.la-samhna.de/samhain.

Реалізація: програмна.

ОС: Unix, Linux, Windows / Cygwin.

Ліцензія: GNU GPL

Даний продукт має відкритий вихідний код та OpenSource-ліцензію. Задача Samhain – це захист машини на якій встановлена IDS система.[3] В своїй роботі цей програмний продукт задіє багато аналітичних методів, завдяки чому контролюються всі системні події:

- створення при першому запуску бази даних сигнатур важливих файлів і її порівняння в подальшому з «живою» системою;
- моніторинг і аналіз записів в журналах;
- контроль входу / виходу в систему;
- моніторинг підключень до відкритих мережевих портів;
- контроль файлів з встановленим SUID і прихованих процесів.

Однією з особливостей даного програмного продукту є режим невидимості, при його активації всі процеси, що відбуваються в ядрі не відслідковуються в пам'яті. Функціонал Samhain дозволяє здійснювати моніторинг одночасно декількох вузлів мережі, при чому вони можуть керуватися різними операційними системами, а всі події будуть реєструватися на один комп'ютер. Для збору і передачі даних моніторингу системи застосовують захищений канал по якому здійснюється передача інформації на сервер де вона обробляється і записується в базу даних. Однією з задач сервера є також надсилання оновлень і змін в конфігурації на клієнтських машинах.

Даний програмний продукт призначений в першу чергу для систем під управлінням операційної системи Linux тому він сумісний майже з усіма версіями, а також є можливість встановлення на Windows.

2.6. Snort

Розробник: Sourcefire, Inc.

Web: http://www.snort.org/

Реалізація: програмна

ОС: Unix, Linux, Windows.

Ліцензія: GPLv2 та комерційна

Своєю популярністю та розповсюдженістю Snort зобов'язаний відкритим вихідним кодом. Snort є однією з найстарших IDS/IPS систем в світі, він був розробленим у 1998 році. Створенням системи займався Мартін Рош, він є одним

з основоположників інформаційної безпеки та автором великої кількості книг в даному направленні. Дана IDS була розроблена, як відповідь на відсутність на той час ефективних і некомерційних систем, які попереджають про мережеві атаки.[20]

Дана IPS виявляє наступне:

- Підозрілий трафік
- Використання експлойтів (виявлення Shellcode)
- Сканування системи (порти, ОС, користувачі і т.д.)
- Атаки на різні служби (Telnet, FTP, DNS, і т.д.)
- DoS / DdoS атаки
- Web-атаки на сервера (cgi, php, frontpage, iss і т.д.)
- Ін’єкції SQL, Oracle і т.д.
- Атаки по протоколам SNMP, NetBios, ICMP
- Атаки за протоколами SMTP, IMAP, pop2, pop3
- Різні Backdoors
- Web-фільтри
- віруси

Крім усього іншого Snort має:

- Можливість написання власних правил
- Розширення функціональності, використовуючи можливість підключення модулів
- Гнучку систему оповіщення про атаки (Log файли, пристрої виведення, БД І.Д.)

Snort підтримує такі інтерфейси для прослуховування:

- Ethernet
- SLIP
- PPP

Однією з причин такої розповсюдженості Snort є її кросплатформеність, встановити дану програму можна, як на розповсюджених операційних системах так і маловідомих. Функціонал Snort можна значно збільшити завдяки розширенню – inline, воно дозволяє прив'язати firewall до дій правил. Часто така зв'язка дає змогу виявити спробу DDoS атаки ще на її початку та вжити запобіжних заходів. Хоча такий метод захисту має і певні недоліки: в тому випадку коли хакер знає, що трафік буде блокуватися, він може підробити пакети так ніби вони надходять з важливих серверів.[14]

Розглянемо архітектуру Snort

В основі системи Snort лежить двигун, який в свою чергу складається з декількох модулів:

- Сніфер пакетів, головна роль цього модуля полягає в перехваті даних, що передаються мережею і подальшій їх відправці на декодер. Дані функції реалізовані завдяки бібліотеці DAQ (Data AcQuisition). Сніфер пакетів має два режими роботи: пасивний (passive) або читання даних мережі з попередньо підготовленого файлу.
- Наступний модуль – декодер пакетів, серед функцій даного модуля є: сортування заголовків перехоплених пакетів, пошук відхилень і аномалій, здійснення аналізу TCP-прапорів, перевіркою певних протоколів та їх роботи. Декодер працює з протоколами стеку TCP / IP.
- На відміну від декодера, який працює з пакетами, що передаються на 2, 3 рівнях моделі модуль препроцесора здійснює більш детальний аналіз і формалізацію пакетів забраних на 3, 4, 7 рівнях. До найпоширеніших препроцесорів належать: frag3 – він працює з трафіком, що фрагментований, http_inspect – здійснює нормалізацію трафіку HTTP, sfPortscan – виявляє підозрілі дії з портами ,DCE / RPC2. І снує також велика кількість декодерів для інших протоколів IMAP, Telnet, SMTP,

SIP, SSL, FTP, SSH і тд. В окремих випадках системні адміністратори пишуть власні препроцесори, вони зазвичай вузько направлені.

- Двигун, що відповідає за виявлення атак є двокомпонентним. Так як Snort працює за правилами то є конструктор правил, його завдання збір всіх доступних правил, певних сигнатур атак і компіляція з зібраних даних цільний набір. Даний набір максимально оптимізовано для використання системою під час пошуку порушень чи аномалій.
- Движок виявлення атак: даний движок має дві частини. Конструктор правил здійснює збір безліч різних важливих правил у єдиний набір. Цей набір є досить оптимізованим для подальшого використання підсистемою інспекції захопленого і обробленого трафіку у пошуках певних порушень.
- За формування результату перевірки в системі Snort відповідає модуль виведення. В результаті проведення атаки система може згенерувати певне повідомлення чи записати отриману інформацію про атаку в спеціальний файл, система Snort працює з різними форматами файлів: syslog, ASCII, PCAP, Unified2.

3. ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1. Підготовчій етап

Для виконання поставленої задачі, розгортання на комп'ютері IDS / IPS системи, було обрано систему Snort. Дане програмне забезпечення має безкоштовну версію, можливість встановлення на машину з операційною системою Windows та невисокі системні вимоги. Варто відмітити, що система запобігання вторгненням доступна на операційних системах Linux та Windows. В контексті даної роботи система буде встановлюватися на комп'ютер з операційною системою.

Відмітимо характеристики системи та ОС:

Версія ОС: Windows 10 Version 19042

Характеристики комп'ютера: Intel Pentium N4200, ОЗУ 4GB

Версія Snort: 2.9.17.1

Розгортання системи Snort ускладняється великою кількістю додаткових дій та маніпуляцій, які необхідно зробити. Для пересічного користувача це може бути заважким і він може допустити певні помилки на різних етапах інсталяції. В результаті цього було вирішено написати bat-скрипт за допомогою якого буде автоматизована більша частина процесів інсталяції системи Snort. Певні етапи цього процесу неможливо повністю автоматизувати.

Першим етапом в певну папку потрібно помістити всі необхідні інсталятори, архів з правилами, файл конфігурації та сам скрипт. Для роботи використано:

- Архіватор 7-Zip версія 1900
- Npcap версія 0.9984
- Snort версія 2.9.17.1
- Архів з правилами для Snort 2.9.17.1

- Файл конфігурації Snort
- Файл зі bat-скриптом для інсталяції

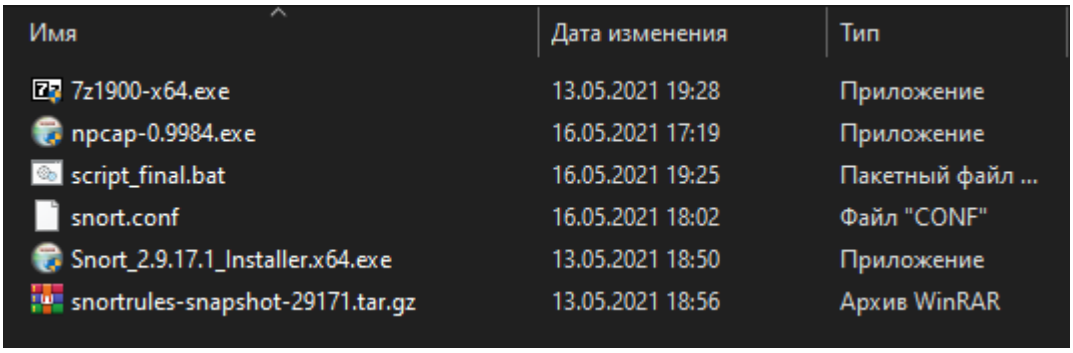

Рисунок 3.1. - Папка з інсталяторами і скриптом

Помістивши всі необхідні файли та інсталятори в одну директорію необхідно здійснити редагування файлу конфігурації для базового запуску системи запобігання вторгненням. Після успішної інсталяції системи можна буде змінити файл налаштувань відповідно до індивідуальних потреб.[14]

В нашому випадку зробимо базові налаштування системи для цього відкриваємо файл snort.conf (параметри конфігурації для запуску програми) в NotePad++ або іншому текстовому редакторі. Повністю підготовлений файл конфігурації наведено в Додатку Б.[19]

Приблизно на 103 рядку знаходимо, встановлений розробником за замовчанням рядок, покажчик шляху: c:\snort\rules, ми вказуємо шлях за яким знаходяться правила Snort. Вона збігається з розташуванням файлу на нашому комп’ютері. Можна обрати власний шлях але потрібно внести аналогічні зміни і у скрипті Там, де треба редагувати шляху, розробник ставить дві точки.

```
# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH c:\Snort\rules
var SO_RULE_PATH c:\Snort\so_rules
var PREPROC_RULE_PATH c:\Snort\preproc_rules

# If you are using reputation preprocessor set these
var WHITE_LIST_PATH c:\snort\rules
var BLACK_LIST_PATH c:\snort\rules
```

Рисунок 3.2. - Вказуємо шлях до Rules

Тепер ми маємо вказати шлях до папки в якій розташовані Log-файли, в них будуть заноситися всі логи роботи Snort, там можна відкрити їх для вивчення. Вкажемо шлях до файлів з логами.

У директорії де встановлено Snort вже створена відповідна папка, копіюємо шлях до неї та вносимо його у файл конфігурації. Перейдемо на 182 рядок там пропишемо в config logdir: c:\snort\log. Обов'язково потрібно видалити символ "#", який означає, що рядок коду закоментовано.

```
181  #
182  # config logdir: c:\Snort\log
183
```

Рисунок 3.3. - Вказуємо шлях до папки з логами

За схожим принципом знаходимо і редагуємо шлях до бібліотеки, важливо пам'ятати, що якщо планується встановлювати в директорію не за замовчанням то це потрібно також вказати при прописуванні шляхів.

```
242  # path to dynamic preprocessor libraries
243  dynamicpreprocessor directory c:\Snort\lib\snort_dynamicpreprocessor
244
245  # path to base preprocessor engine
246  dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll
247
248  # path to dynamic rules libraries
249  #dynamicdetection directory c:\Snort\lib\snort_dynamicrules
```

Рисунок 3.4. - Вказуємо шлях до бібліотек

Наступним кроком потрібно закоментувати певні рядки коду, вони наведені нижче.

Inline packet normalization. For more information, see README.normalize

Does nothing in IDS mode

preprocessor normalize_ip4

preprocessor normalize_tcp: block, rsv, pad, urp, req_urg, req_pay, req_urp, ips, ecn stream

preprocessor normalize_icmp4

preprocessor normalize_ip6

preprocessor normalize_icmp6

Back Orifice detection.

preprocessor bo

А з деяких рядків, навпаки прибрати коментар:

Portscan detection. For more information, see README.sfportscan

preprocessor sfportscan: proto {all} memcap { 10000000 } sense_level { low }

Знаходимо рядок де вказується шлях до white.list та black.list. Змінюємо його наступним чином:

```
508     whitelist $WHITE_LIST_PATH\white.list, \
509     blacklist $BLACK_LIST_PATH\black.list
```

Рисунок 3.5. - Вказані шляхи до розташування white/black- list

З 545 рядочка до 660 замінюємо символ “/” на символ “\” в кожному шляху.

3.2. Опис програмної реалізації

Реалізовувати поставлену задачу було вирішено у bat-скрипті написаного командами командного рядка.

В першій частині скрипту здійснюємо встановлення всіх необхідних програм та утиліт. Для автоматизації процесу було обрано режим «тихої» інсталяції. В цьому випадку програми встановлюються без участі користувача. Для цього порисується наступний рядок з кодом: "%~dp0\Snort_2.9.17.1_Installer.x64.exe" -y /S. Параметр -y означає, що на всі запити системи буде обрана відповідь «ОК», а параметр /S – вмикає режим тихої установки. Слід зазначити, що для різних типів інсталяторів команда для тихої установки дещо відрізняється.

Після виконання даної команди система чекає 5 секунд, після чого здійснює перевірку чи успішно пройшла інсталяція.

ping -n 6 localhost>Nul

set "prog="C:\Snort\bin\snort.exe""

if exist "%prog%" (

@echo Програма Snort успішно встановлена

@echo.

)Наступним етапом проходить розпаковка архіву з правилами. Даний архів типу *.tar.gz, де gz – це тип архіву, а tar – ступінь стикання інформації. Для роботи з архівом такого типу необхідна програма 7-Zip, процес розархівації, через своєрідний тип архіву, відбувається у два етапи, в процесі розпаковки дані тимчасова завантажуються у буфер пам’яті.

@echo Розархівація правил

"C:\Program Files\7-Zip\7z.exe" e "E:\Мои файлы\Учеба\Диплом\Преддипломная практика\Snort_installing\snortrules-

snapshot-29171.tar.gz" -so | "C:\Program Files\7-Zip\7z.exe" x -aoa -si -ttar -o"C:\Snort\"

Далі створюються blacklist і whitelist:

@echo Створення blacklist и whitelist

@echo .>C:\Snort\rules\black.list

@echo .>C:\Snort\rules\white.list

Наступним кроком копіюємо раніше підготовлений файл конфігурації у відповідну папку де розташований Snort.

@echo Перенос файлу конфігурації

xcopy "%~dp0\snort.conf" C:\Snort\

Скрипт також додає власні правила у відповідний файл. Описані правила сповіщають адміністратора системи, що користувачі відвідали певні заборонені сайти, додані у правила. До таких сайтів внесені найпопулярніші розважальні ресурси, зроблене це для того, щоб працівники не відволікалися від роботи впродовж робочого дня.

@echo Додавання власних правил

echo f| copy C:\Snort\rules\local.rules C:\Snort\rules\temp.txt

echo. alert tcp any any -> any any (content: "pikabu.ru"; msg: "Visiting site pikabu.ru detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules

type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules

del C:\Snort\rules\temp.txt

Після того, як все необхідне підготовлено можна запускати скрипт. Для більш коректної роботи його потрібно запускати від імені адміністратора.

Процес виконання скрипту займає небагато часу, всі дії супроводжуються відповідними надписами.[21]

Не вдалося зробити тиху установку для програми Npcap, це пов’язано з тим, що режим тихої інсталяції доступний лише в комерційній версії програми. Тому виконання скрипту перерветься появою діалогового вікна для інсталяції Npcap. В діалоговому вікні необхідно натиснути “I agree”, а на наступному кроці обрати всі чотири компоненти для інсталяції.

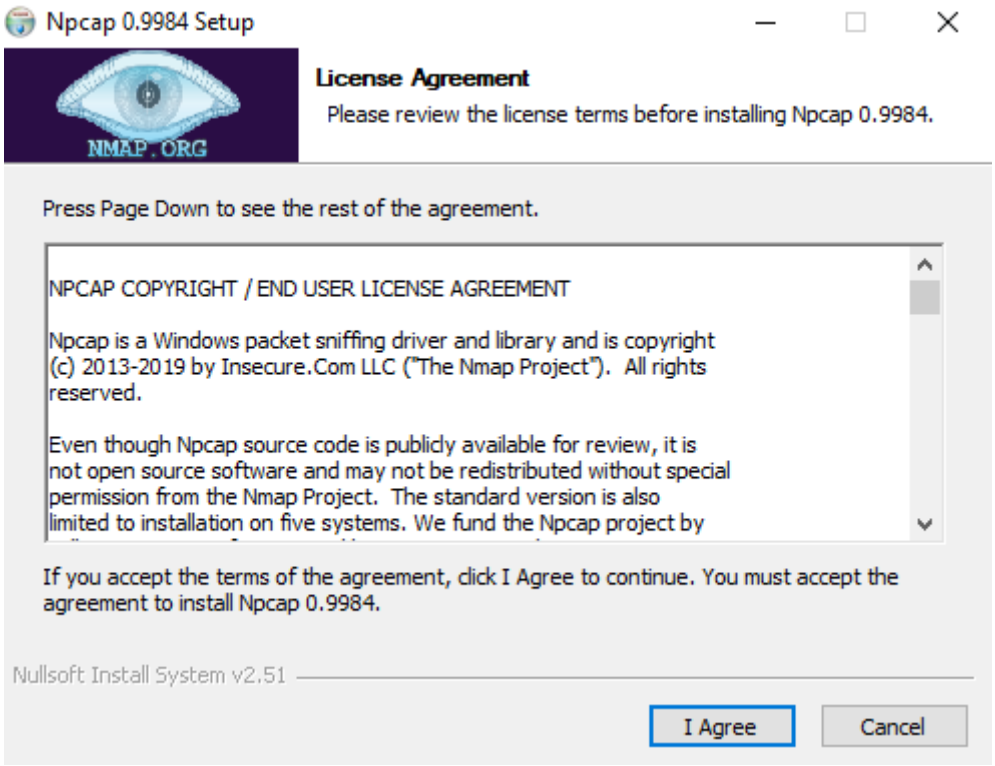

Рисунок 3.6. - Встановлення Npcap

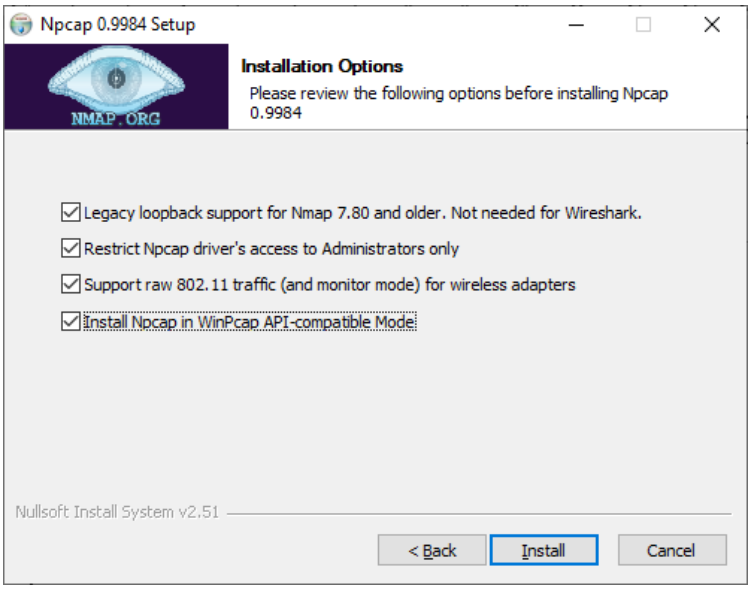

Рисунок 3.7. - Встановлення Npcap

Під час виконання скрипту на екрані всі події супроводжуються відповідними коментарями. Після завершення роботи скрипт не закривається, там виводиться список команд для першого запуску системи.

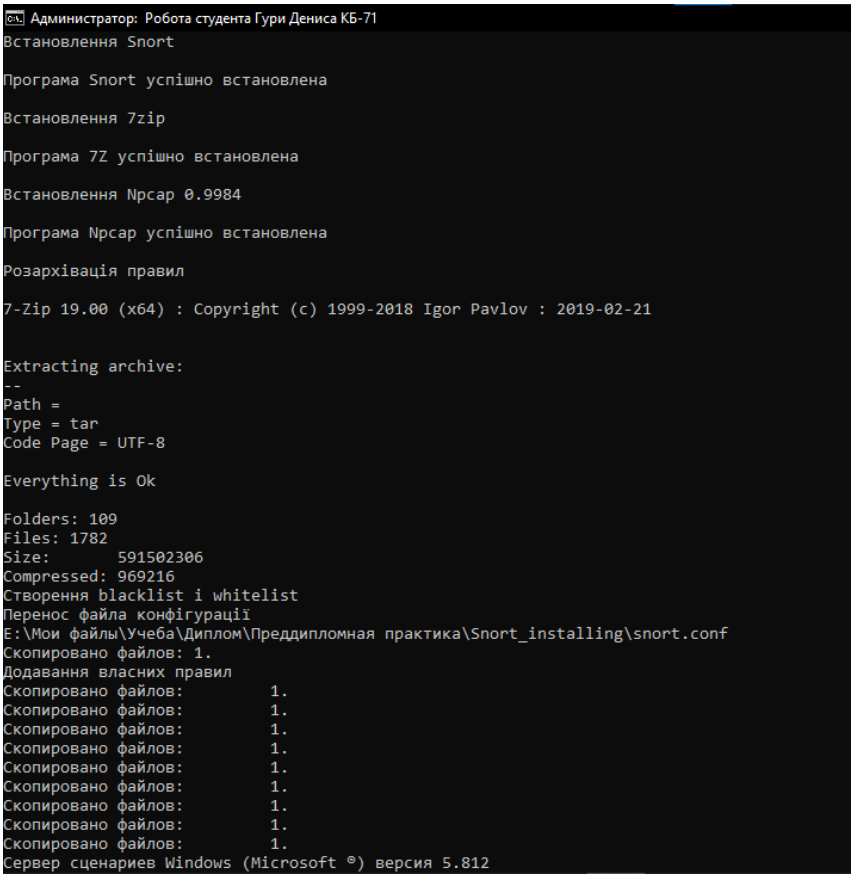

Рисунок 3.8. - Результат виконання сценарію скрипту

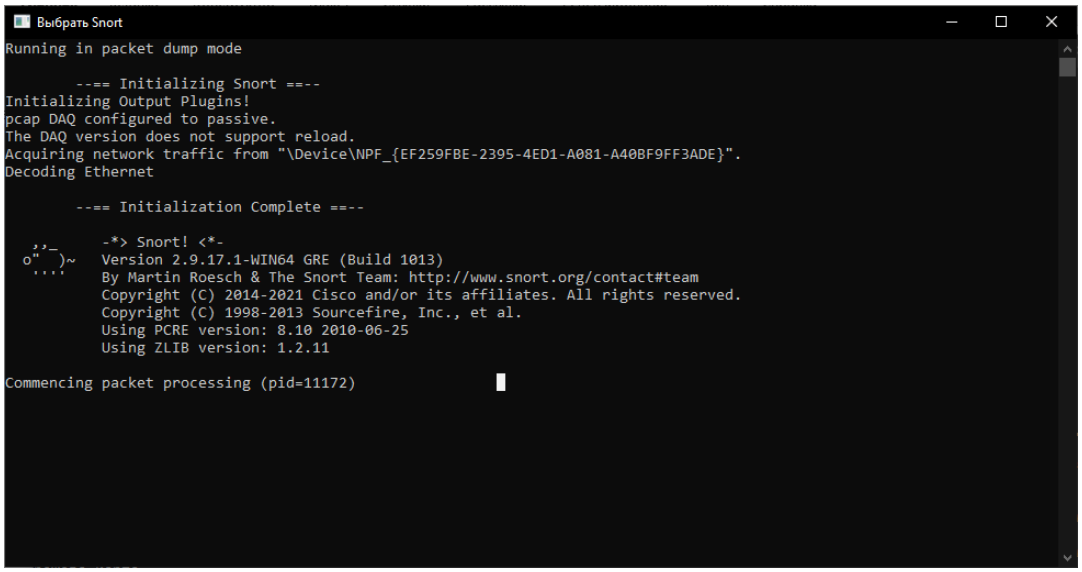

Рисунок 3.9. - Запущена система Snort

Після завершення виконання сценарію скрипту, можна відкрити командний рядок та запустити до виконання Snort. Переходимо в кореневу папку Snort\bin\ і виконуємо команду Snort -W.

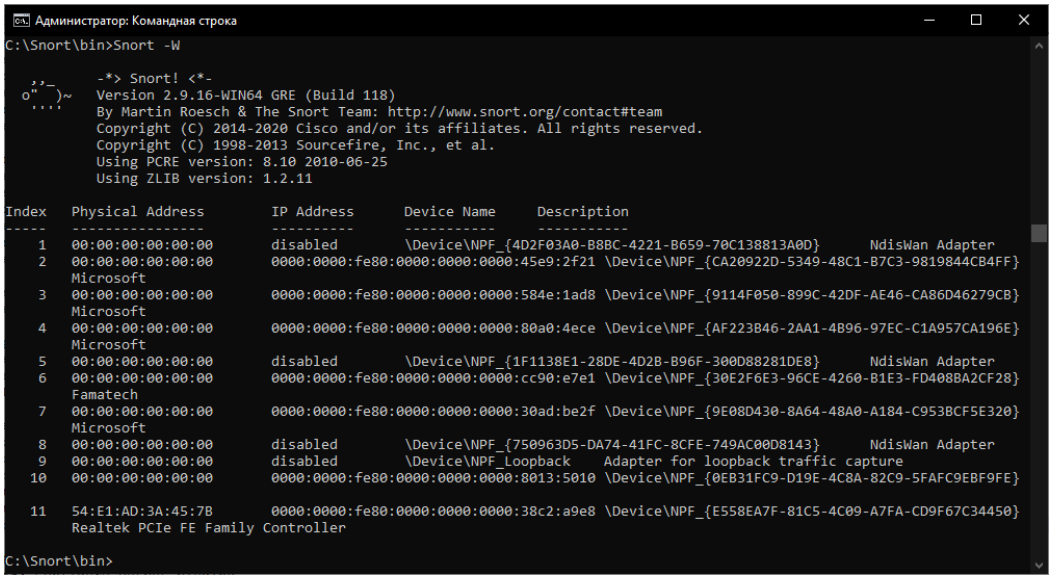

Рисунок 3.10. - Виконання команди Snort -W

Результатом виконання цієї команди є список доступних мережевих пристроїв. В цьому списку потрібно знайти мережеву карту комп’ютера, в даному випадку він знаходиться під номером 11.

Також перевіримо правильність налаштування файлу snort.conf, для цього виконаємо команду -T -c c:\snort\etc\snort.conf -l c:\snort\log -i 11 (в кінці потрібно вказати номер під яким знаходиться мережева карта).

Якщо все буде виконано правильно то в результаті виконання команди отримаємо наступне повідомлення:

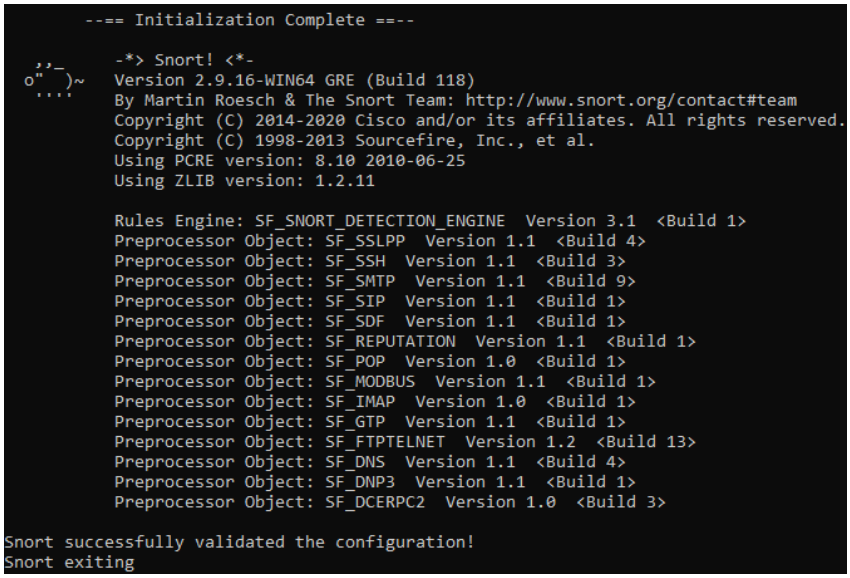

Рисунок 3.11. - Перевірка налаштувань файлу config

Після успішної перевірки файлу налаштувань можна запустити Snort. Snort -A console -c c:\snort\etc\snort.conf -l c:\snort\log -i 11

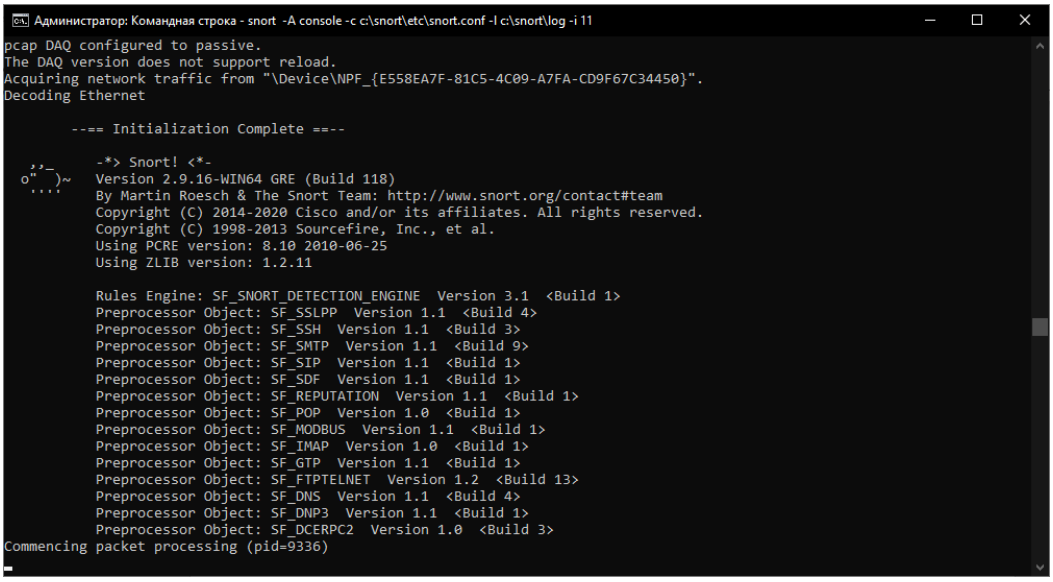

Рисунок 3.12. - Система Snort запущена

Список додаткових команд:

Таблиця 3.1. - Ключі для системи Snort

№	Ключ	Опис дії
1	-T	вказує, що потрібно протестувати поточну конфігурацію Snort
2	-c	означає, що включений режим IDS, далі йде шлях до конфігураційного файлу snort.conf
3	-l	включає режим запису на жорсткий диск із зазначенням шляху до файлу
4	-A	показує що все попередження (alerts) будуть дублюватися висновком на консоль
5	-i	вказує на порядковий номер (index) цікавить нас інтерфейсу

ВИСНОВКИ

В результаті виконання випускної роботи було виконано всі зазначені в постановці завдання.

Зроблено доскональний інформаційний розгляд поняття про IDS / IPS системи, вивчено історію появи та розповсюдження перших систем виявлення атак, принципи їх роботи, типову архітектуру та головні відмінності. В процесі дослідження питання також було розглянуто найпопулярніші на сьогоднішній день IDS / IPS системи.

Вивченню підлягли в рівній мірі системи запобігання вторгненням як з комерційною ліцензією так і безкоштовні додатки. Для кожної з розглянутих систем було виявлено переваги та недоліки, основні особливості роботи та спеціалізацію.

Засновуючись на цих даних було обрано IDS / IPS систему Snort для виконання поставлених задач випускної роботи. Обрана система може бути встановлена на операційну систему Windows та має некомерційну ліцензію.

Було проаналізовано процес встановлення та налаштування обраної системи запобігання вторгненням, визначено ключові етапи та необхідне програмне забезпечення.

На подальшому етапі було розроблено bat-скрипт, за допомогою якого відбувається інсталяція IDS / IPS системи та базові налаштування.

Розроблений скрипт дозволяє навіть користувачу з мінімальними навиками роботи за комп’ютером здійснити встановлення та налаштування системи. Також, він буде корисним адміністраторам мереж, як інструмент, який підвищить рівень їх особистої діяльності. Використання скрипту зменшить трудові та часові затрати, які наразі пов’язані з розгортанням системи Snort.

СПИСОК ЛІТЕРАТУРИ

1. МЕРИОН НЕТВОРКС. УСТАНОВКА И НАСТРОЙКА УТИЛИТЫ ДЛЯ ОБНАРУЖЕНИЯ ВТОРЖЕНИЙ В СЕТИ – SNORT [Електронний ресурс] / МЕРИОН НЕТВОРКС – Режим доступу до ресурсу: https://wiki.merionet.ru/seti/31/ustanovka-i-nastrojka-utility-dlja-obnaruzhenija-vtorzhenij-v-seti-snort/.

2. Применение IDS/IPS [Електронний ресурс] – Режим доступу до ресурсу: https://xakep.ru/2012/10/29/ids-ips/.

3. ЭВОЛЮЦИЯ СИСТЕМ IPS/IDS: ПРОШЛОЕ, НАСТОЯЩЕЕ И БУДУЩЕЕ [Електронний ресурс] – Режим доступу до ресурсу: https://wiki.merionet.ru/seti/20/evolyuciya-sistem-ips-ids-proshloe-nastoyashhee-budushhee/.

4. Dorothy D. Requirements and Model for IDES - A Real-Time Intrusion-Detection Expert System / Denning Dorothy., 1985. – 74 с.

5. Работа с bat-файлами [Електронний ресурс] – Режим доступу до ресурсу: https://datbaze.ru/article/rabota-s-bat-faylami.html.

6. Denning, Dorothy E., "An Intrusion Detection Model, " Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119—131

7. Lunt, Teresa F., "Detecting Intruders in Computer Systems, " 1993 Conference on Auditing and Computer Technology, SRI International

8. Дмитрий К. Системы обнаружения атак / Костров Дмитрий., 2002. – 154 с.

9. Активный аудит. // JetInfo. – 1999. – №8. – С. 28.

10. Интрасети: доступ в Internet, защита : учебное пособие для вузов. // ООО «ЮНИТИ-ДАНА». – 2000.

11. Container security requires more than securing your images [Електронний ресурс] – Режим доступу до ресурсу: https://developer.ibm.com/solutions/security/.

12. Чередняк Л. Семантический анализ на службе / Л. Чередняк. – 2010. – №10.

13. Прохоров А. Защита присутствия в Интернете от вирусов / А. Прохоров. // КомпьютерПресс. – 2005.

14. Snort instaling on Windows [Електронний ресурс] – Режим доступу до ресурсу: https://www.snort.org/.

15. Paxson, Vern, "Bro: A System for Detecting Network Intruders in Real-Time, " Proceedings of The 7th USENIX Security Symposium, San Antonio, TX, 1998

16. Endorf C. Intrusion Detection and Prevention / C. Endorf, G. Schultz, J. Mellander., 2003. – 500 с.

17. Pepe M. Incident Response & Computer Forensics, Third Edition / M. Pepe, K. Mandia., 2014. – 624 с. – (3).

18. Steinberg J. Cybersecurity For Dummies / Joseph Steinberg., 2019. – 368 с. – (1st edition).

19. Ozkaya E. Cybersecurity: The Beginner's Guide: A comprehensive guide to getting started in cybersecurity / Erdal Ozkaya., 2019. – 396 с. – (Packt Publishing).

20. Mitnick K. The Art of Invisibility: The World's Most Famous Hacker Teaches You How to Be Safe in the Age of Big Brother and Big Data / Kevin Mitnick., 2019. – 320 с.

21. Mitnick K. The Art of Intrusion: The Real Stories Behind the Exploits of Hackers, Intruders and Deceivers / Kevin Mitnick., 2005. – 322 с.

22. Mining E. Kali Linux Hacking: A Complete Step by Step Guide to Learn the Fundamentals of Cyber Security, Hacking, and Penetration Testing. Includes Valuable Basic Networking Concepts. / Ethem Mining., 2019. – 175 с.

ДОДАТОК А

```
chcp 1251
@echo off
cls

title Робота студента Гури Дениса КБ-71

@echo Встановлення Snort
@echo.
echo y| "%~dp0\Snort_2.9.17.1_Installer.x64.exe" -y /S
ping -n 6 localhost>Nul
:: Перевірка чи встановлений Snort
set "prog="C:\Snort\bin\snort.exe""
if exist "%prog%" (
   @echo Програма Snort успішно встановлена
   @echo.
)
ping -n 3 localhost>Nul

@echo Встановлення 7zip
@echo.
"%~dp0\7z1900-x64.exe" -y /S
ping -n 6 localhost>Nul
:: Перевірка чи встановлений 7zip
set "prog="C:\Program Files\7-Zip\7z.exe""
if exist "%prog%" (
   @echo Програма 7Z успішно встановлена
   @echo.
)
ping -n 3 localhost>Nul

@echo Встановлення Npcap 0.9984
@echo.
"%~dp0\npcap-0.9984.exe"
ping -n 6 localhost>Nul
:: Перевірка чи встановлений Npcap
set "prog=""C:\Program Files\Npcap\Uninstall.exe""
if exist "%prog%" (
   @echo Програма Npcap успішно встановлена
   @echo.
)
ping -n 3 localhost>Nul

:: Розархівація фалів з правилами у відповідну папку
@echo Розархівація правил
"C:\Program Files\7-Zip\7z.exe" e "E:\Мои файлы\Учеба\Диплом\Преддипломная
практика\Snort_installing\snortrules-snapshot-29170.tar.gz" -so | "C:\Program
Files\7-Zip\7z.exe" x -aoa -si -ttar -o"C:\Snort\"

:: Створення у папці Rules файлів blacklist і whitelist
@echo Створення blacklist і whitelist
@echo .>C:\Snort\rules\black.list
@echo .>C:\Snort\rules\white.list

:: Перенос готового файлу конфігурації до папки зі Snort
@echo Перенос файла конфігурації
xcopy "%~dp0\snort.conf" C:\Snort\

:: Додавання до файлу local.rules власних правил
@echo Додавання власних правил
echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
```

```
echo. alert tcp any any -> any any  (content: "www.xakep.ru"; msg: "Visiting site
www.xakep.ru detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "pikabu.ru"; msg: "Visiting site
pikabu.ru detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "kanobu.ru"; msg: "Visiting site
kanobu.ru detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "www.facebook.com"; msg: "Visiting
site www.facebook.com detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "www.youtube.com"; msg: "Visiting
site www.youtube.com detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "fishki.net"; msg: "Visiting site
fishki.net detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "www.adme.ru"; msg: "Visiting site
www.adme.ru detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "baskino.me"; msg: "Visiting site
baskino.me detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "www.netflix.com"; msg: "Visiting
site www.netflix.com detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "twitter.com"; msg: "Visiting site
twitter.com detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
```

```
echo. alert tcp any any -> any any  (content: "www.yaplakal.com"; msg: "Visiting
site www.yaplakal.com detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

echo f| copy  C:\Snort\rules\local.rules  C:\Snort\rules\temp.txt
echo. alert tcp any any -> any any  (content: "mirpozitiva.ru"; msg: "Visiting
site mirpozitiva.ru detected"; sid:1000008; rev:1)>C:\Snort\rules\local.rules
type C:\Snort\rules\temp.txt >>C:\Snort\rules\local.rules
del C:\Snort\rules\temp.txt

:: Створення ярлику для Snort
echo Set oWS = WScript.CreateObject("WScript.Shell") > CreateShortcut.vbs
echo sLinkFile = "%HOMEDRIVE%%HOMEPATH%\Desktop\Snort.lnk" >> CreateShortcut.vbs
echo Set oLink = oWS.CreateShortcut(sLinkFile) >> CreateShortcut.vbs
echo oLink.TargetPath = "C:\Snort\bin\snort.exe" >> CreateShortcut.vbs
echo oLink.Save >> CreateShortcut.vbs
cscript CreateShortcut.vbs
del CreateShortcut.vbs

:: Запуск Snort у новому вікні
start %HOMEDRIVE%%HOMEPATH%\Desktop\Snort.lnk

echo Для роботи зі Snort використайте наступні команди:
echo 1. Snort -W покаже список доступних мережевих пристроїв
echo 2. Snort -T -c c:\snort\etc\snort.conf -l c:\snort\log -i   в кінці потрібно
вказати номер під яким знаходиться мережева карта
echo 3. snort -A console -c c:\snort\etc\snort.conf -l c:\snort\log -i в кінці
потрібно вказати номер під яким знаходиться мережева карта

@pause
```

ДОДАТОК Б

Вміст файлу snort.conf.

```
#--------------------------------------------------
#   VRT Rule Packages Snort.conf
#
#   For more information visit us at:
#     http://www.snort.org                   Snort Website
#     http://vrt-blog.snort.org/    Sourcefire VRT Blog
#
#     Mailing list Contact:      snort-sigs@lists.sourceforge.net
#     False Positive reports:    fp@sourcefire.com
#     Snort bugs:                bugs@snort.org
#
#     Compatible with Snort Versions:
#     VERSIONS : 2.9.11.0
#
#     Snort build options:
#      OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --
enable-perfprofiling --enable-zlib --enable-active-response --enable-normalizer -
-enable-reload --enable-react --enable-flexresp3
#
#     Additional information:
#     This configuration file enables active response, to run snort in
#     test mode -T you are required to supply an interface -i <interface>
#     or test mode will fail to fully validate the configuration and
#     exit with a FATAL error
#--------------------------------------------------

######################################################
# This file contains a sample snort configuration.
# You should take the following steps to create your own custom configuration:
#
#  1) Set the network variables.
#  2) Configure the decoder
#  3) Configure the base detection engine
#  4) Configure dynamic loaded libraries
#  5) Configure preprocessors
#  6) Configure output plugins
#  7) Customize your rule set
#  8) Customize preprocessor and decoder rule set
#  9) Customize shared object rule set
######################################################

######################################################
# Step #1: Set the network variables.  For more information, see README.variables
######################################################

# Setup the network addresses you are protecting
ipvar HOME_NET any

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET any

# List of DNS servers on your network
ipvar DNS_SERVERS $HOME_NET

# List of SMTP servers on your network
ipvar SMTP_SERVERS $HOME_NET
```

```
# List of web servers on your network
ipvar HTTP_SERVERS $HOME_NET

# List of sql servers on your network
ipvar SQL_SERVERS $HOME_NET

# List of telnet servers on your network
ipvar TELNET_SERVERS $HOME_NET

# List of ssh servers on your network
ipvar SSH_SERVERS $HOME_NET

# List of ftp servers on your network
ipvar FTP_SERVERS $HOME_NET

# List of sip servers on your network
ipvar SIP_SERVERS $HOME_NET

# List of ports you run web servers on
portvar                                                                 HTTP_PORTS
[36,80,81,82,83,84,85,86,87,88,89,90,311,383,555,591,593,631,801,808,818,901,972
,1158,1220,1414,1533,1741,1830,1942,2231,2301,2381,2578,2809,2980,3029,3037,3057
,3128,3443,3702,4000,4343,4848,5000,5117,5250,5450,5600,5814,6080,6173,6988,7000
,7001,7005,7071,7144,7145,7510,7770,7777,7778,7779,8000,8001,8008,8014,8015,8020
,8028,8040,8080,8081,8082,8085,8088,8090,8118,8123,8180,8181,8182,8222,8243,8280
,8300,8333,8344,8400,8443,8500,8509,8787,8800,8888,8899,8983,9000,9002,9060,9080
,9090,9091,9111,9290,9443,9447,9710,9788,9999,10000,11371,12601,13014,15489,1998
0,29991,33300,34412,34443,34444,40007,41080,44449,50000,50002,51423,53331,55252,
55555,56712]

# List of ports you want to look for SHELLCODE on.
portvar SHELLCODE_PORTS !80

# List of ports you might see oracle attacks on
portvar ORACLE_PORTS 1024:

# List of ports you want to look for SSH connections on:
portvar SSH_PORTS 22

# List of ports you run ftp servers on
portvar FTP_PORTS [21,2100,3535]

# List of ports you run SIP servers on
portvar SIP_PORTS [5060,5061,5600]

# List of file data ports for file inspection
portvar FILE_DATA_PORTS [$HTTP_PORTS,110,143]

# List of GTP ports for GTP preprocessor
portvar GTP_PORTS [2123,2152,3386]

# other variables, these should not be modified
ipvar                                                                   AIM_SERVERS
[64.12.24.0/23,64.12.28.0/23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24,205.18
8.3.0/24,205.188.5.0/24,205.188.7.0/24,205.188.9.0/24,205.188.153.0/24,205.188.1
79.0/24,205.188.248.0/24]

# Path to your rules files (this can be a relative path)
# Note for Windows users:  You are advised to make this an absolute path,
# such as:  c:\snort\rules
var RULE_PATH c:\snort\rules
```

```
var SO_RULE_PATH c:\snort\rules
var PREPROC_RULE_PATH c:\snort\rules

# If you are using reputation preprocessor set these
var WHITE_LIST_PATH c:\snort\rules
var BLACK_LIST_PATH c:\snort\rules

##################################################
# Step #2: Configure the decoder.  For more information, see README.decode
##################################################

# Stop generic decode events:
config disable_decode_alerts

# Stop Alerts on experimental TCP options
config disable_tcpopt_experimental_alerts

# Stop Alerts on obsolete TCP options
config disable_tcpopt_obsolete_alerts

# Stop Alerts on T/TCP alerts
config disable_tcpopt_ttcp_alerts

# Stop Alerts on all other TCPOption type events:
config disable_tcpopt_alerts

# Stop Alerts on invalid ip options
config disable_ipopt_alerts

# Alert if value in length field (IP, TCP, UDP) is greater th elength of the
packet
# config enable_decode_oversized_alerts

#   Same   as   above,   but   drop   packet   if   in   Inline   mode   (requires
enable_decode_oversized_alerts)
# config enable_decode_oversized_drops

# Configure IP / TCP checksum mode
config checksum_mode: all

# Configure maximum number of flowbit references.  For more information, see
README.flowbits
# config flowbits_size: 64

# Configure ports to ignore
# config ignore_ports: tcp 21 6667:6671 1356
# config ignore_ports: udp 1:17 53

# Configure active response for non inline operation. For more information, see
REAMDE.active
# config response: eth0 attempts 2

# Configure DAQ related options for inline operation. For more information, see
README.daq
#
# config daq: <type>
# config daq_dir: <dir>
# config daq_mode: <mode>
# config daq_var: <var>
#
# <type> ::= pcap | afpacket | dump | nfq | ipq | ipfw
```

```
# <mode> ::= read-file | passive | inline
# <var> ::= arbitrary <name>=<value passed to DAQ
# <dir> ::= path as to where to look for DAQ module so's

# Configure specific UID and GID to run snort as after dropping privs. For more
information see snort -h command line options
#
# config set_gid:
# config set_uid:

# Configure default snaplen. Snort defaults to MTU of in use interface. For more
information see README
#
# config snaplen:
#

# Configure default bpf_file to use for filtering what traffic reaches snort. For
more information see snort -h command line options (-F)
#
# config bpf_file:
#

# Configure default log directory for snort to log to.  For more information see
snort -h command line options (-l)
#
config logdir: c:\Snort\log


##################################################
# Step #3: Configure the base detection engine.   For more information, see
README.decode
##################################################

# Configure PCRE match limitations
config pcre_match_limit: 3500
config pcre_match_limit_recursion: 1500

# Configure the detection engine  See the Snort Manual, Configuring Snort -
Includes - Config
config detection: search-method ac-split search-optimize max-pattern-len 20

# Configure the event queue.  For more information, see README.event_queue
config event_queue: max_queue 8 log 5 order_events content_length

##################################################
## Configure GTP if it is to be used.
## For more information, see README.GTP
##################################################

# config enable_gtp

##################################################
# Per packet and rule latency enforcement
# For more information see README.ppm
##################################################

# Per Packet latency configuration
#config ppm: max-pkt-time 250, \
#   fastpath-expensive-packets, \
#   pkt-log
```

```
# Per Rule latency configuration
#config ppm: max-rule-time 200, \
#   threshold 3, \
#   suspend-expensive-rules, \
#   suspend-timeout 20, \
#   rule-log alert

##################################################
# Configure Perf Profiling for debugging
# For more information see README.PerfProfiling
##################################################

#config profile_rules: print all, sort avg_ticks
#config profile_preprocs: print all, sort avg_ticks

##################################################
# Configure protocol aware flushing
# For more information see README.stream5
##################################################
config paf_max: 16000

##################################################
# Step #4: Configure dynamic loaded libraries.
# For more information, see Snort Manual, Configuring Snort - Dynamic Modules
##################################################

# path to dynamic preprocessor libraries
dynamicpreprocessor directory c:\Snort\local\lib\snort_dynamicpreprocessor

# path to base preprocessor engine
dynamicengine c:\Snort\lib\snort_dynamicengine\sf_engine.dll

# path to dynamic rules libraries
#dynamicdetection directory c:\Snort\lib\snort_dynamicrules

##################################################
# Step #5: Configure preprocessors
# For more information, see the Snort Manual, Configuring Snort - Preprocessors
##################################################

# GTP Control Channle Preprocessor. For more information, see README.GTP
# preprocessor gtp: ports { 2123 3386 2152 }

# Inline packet normalization. For more information, see README.normalize
# Does nothing in IDS mode
#preprocessor normalize_ip4
#preprocessor normalize_tcp: block, rsv, pad, urp, req_urg, req_pay, req_urp, ips,
ecn stream
#preprocessor normalize_icmp4
#preprocessor normalize_ip6
#preprocessor normalize_icmp6

# Target-based IP defragmentation.  For more inforation, see README.frag3
preprocessor frag3_global: max_frags 65536
preprocessor  frag3_engine:  policy  windows  detect_anomalies  overlap_limit  10
min_fragment_length 100 timeout 180

# Target-Based stateful inspection/stream reassembly.  For more inforation, see
README.stream5
preprocessor stream5_global: track_tcp yes, \
   track_udp yes, \
```

```
   track_icmp no, \
   max_tcp 262144, \
   max_udp 131072, \
   max_active_responses 2, \
   min_response_seconds 5
preprocessor stream5_tcp: policy windows, detect_anomalies, require_3whs 180, \
   overlap_limit 10, small_segments 3 bytes 150, timeout 180, \
    ports client 21 22 23 25 42 53 70 79 109 110 111 113 119 135 136 137 139 143
\
        161 445 513 514 587 593 691 1433 1521 1741 2100 3306 6070 6665 6666 6667
6668 6669 \
        7000 8181 32770 32771 32772 32773 32774 32775 32776 32777 32778 32779, \
    ports both 36 80 81 82 83 84 85 86 87 88 89 90 110 311 383 443 465 563 555
591 593 631 636 801 808 818 901 972 989 992 993 994 995 1158 1220 1414 1533 1741
1830 1942 2231 2301 2381 2578 2809 2980 3000 3001 3029 3037 3057 3128 3443 3702
4000 4343 4848 5000 5117 5250 5450 5600 5814 6080 6173 6988 7907 7000 7001 7005
7071 7144 7145 7510 7802 7770 7777 7778 7779 \
        7801 7900 7901 7902 7903 7904 7905 7906 7908 7909 7910 7911 7912 7913 7914
7915 7916 \
        7917 7918 7919 7920 8000 8001 8008 8014 8015 8020 8028 8040 8080 8081 8082
8085 8088 8090 8118 8123 8180 8181 8182 8222 8243 8280 8300 8333 8344 8400 8443
8500 8509 8787 8800 8888 8899 8983 9000 9002 9060 9080 9090 9091 9111 9290 9443
9447 9710 9788 9999 10000 11371 12601 13014 15489 19980 29991 33300 34412 34443
34444 40007 41080 44449 50000 50002 51423 53331 55252 55555 56712
preprocessor stream5_udp: timeout 180

# performance statistics.  For more information, see the Snort Manual, Configuring
Snort - Preprocessors - Performance Monitor
# preprocessor perfmonitor: time 300 file /var/snort/snort.stats pktcnt 10000

# HTTP normalization and anomaly detection.  For more information, see
README.http_inspect
preprocessor http_inspect: global iis_unicode_map unicode.map 1252 compress_depth
65535 decompress_depth 65535
preprocessor http_inspect_server: server default \
    http_methods { GET POST PUT SEARCH MKCOL COPY MOVE LOCK UNLOCK NOTIFY POLL
BCOPY BDELETE BMOVE LINK UNLINK OPTIONS HEAD DELETE TRACE TRACK CONNECT SOURCE
SUBSCRIBE UNSUBSCRIBE PROPFIND PROPPATCH BPROPFIND BPROPPATCH RPC_CONNECT
PROXY_SUCCESS BITS_POST CCM_POST SMS_POST RPC_IN_DATA RPC_OUT_DATA RPC_ECHO_DATA
} \
    chunk_length 500000 \
    server_flow_depth 0 \
    client_flow_depth 0 \
    post_depth 65495 \
    oversize_dir_length 500 \
    max_header_length 750 \
    max_headers 100 \
    max_spaces 200 \
    small_chunk_length { 10 5 } \
    ports { 36 80 81 82 83 84 85 86 87 88 89 90 311 383 555 591 593 631 801 808
818 901 972 1158 1220 1414 1533 1741 1830 1942 2231 2301 2381 2578 2809 2980 3029
3037 3057 3128 3443 3702 4000 4343 4848 5000 5117 5250 5450 5600 5814 6080 6173
6988 7000 7001 7005 7071 7144 7145 7510 7770 7777 7778 7779 8000 8001 8008 8014
8015 8020 8028 8040 8080 8081 8082 8085 8088 8090 8118 8123 8180 8181 8182 8222
8243 8280 8300 8333 8344 8400 8443 8500 8509 8787 8800 8888 8899 8983 9000 9002
9060 9080 9090 9091 9111 9290 9443 9447 9710 9788 9999 10000 11371 12601 13014
15489 19980 29991 33300 34412 34443 34444 40007 41080 44449 50000 50002 51423
53331 55252 55555 56712 } \
    non_rfc_char { 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 } \
    enable_cookie \
    extended_response_inspection \
```

```
    inspect_gzip \
    normalize_utf \
    unlimited_decompress \
    normalize_javascript \
    apache_whitespace no \
    ascii no \
    bare_byte no \
    directory no \
    double_decode no \
    iis_backslash no \
    iis_delimiter no \
    iis_unicode no \
    multi_slash no \
    utf_8 no \
    u_encode yes \
    webroot no \
   # decompress_swf { deflate lzma } \
    decompress_pdf { deflate }

# ONC-RPC normalization and anomaly detection.  For more information, see the
Snort Manual, Configuring Snort - Preprocessors - RPC Decode
preprocessor rpc_decode: 111 32770 32771 32772 32773 32774 32775 32776 32777 32778
32779 no_alert_multiple_requests no_alert_large_fragments no_alert_incomplete

# Back Orifice detection.
# preprocessor bo

# FTP / Telnet normalization and anomaly detection.  For more information, see
README.ftptelnet
preprocessor ftp_telnet: global inspection_type stateful encrypted_traffic no
check_encrypted
preprocessor ftp_telnet_protocol: telnet \
    ayt_attack_thresh 20 \
    normalize ports { 23 } \
    detect_anomalies
preprocessor ftp_telnet_protocol: ftp server default \
    def_max_param_len 100 \
    ports { 21 2100 3535 } \
    telnet_cmds yes \
    ignore_telnet_erase_cmds yes \
    ftp_cmds { ABOR ACCT ADAT ALLO APPE AUTH CCC CDUP } \
    ftp_cmds { CEL CLNT CMD CONF CWD DELE ENC EPRT } \
    ftp_cmds { EPSV ESTA ESTP FEAT HELP LANG LIST LPRT } \
    ftp_cmds { LPSV MACB MAIL MDTM MIC MKD MLSD MLST } \
    ftp_cmds { MODE NLST NOOP OPTS PASS PASV PBSZ PORT } \
    ftp_cmds { PROT PWD QUIT REIN REST RETR RMD RNFR } \
    ftp_cmds { RNTO SDUP SITE SIZE SMNT STAT STOR STOU } \
    ftp_cmds { STRU SYST TEST TYPE USER XCUP XCRC XCWD } \
    ftp_cmds { XMAS XMD5 XMKD XPWD XRCP XRMD XRSQ XSEM } \
    ftp_cmds { XSEN XSHA1 XSHA256 } \
    alt_max_param_len 0 { ABOR CCC CDUP ESTA FEAT LPSV NOOP PASV PWD QUIT REIN
STOU SYST XCUP XPWD } \
    alt_max_param_len 200 { ALLO APPE CMD HELP NLST RETR RNFR STOR STOU XMKD } \
    alt_max_param_len 256 { CWD RNTO } \
    alt_max_param_len 400 { PORT } \
    alt_max_param_len 512 { SIZE } \
    chk_str_fmt { ACCT ADAT ALLO APPE AUTH CEL CLNT CMD } \
    chk_str_fmt { CONF CWD DELE ENC EPRT EPSV ESTP HELP } \
    chk_str_fmt { LANG LIST LPRT MACB MAIL MDTM MIC MKD } \
    chk_str_fmt { MLSD MLST MODE NLST OPTS PASS PBSZ PORT } \
    chk_str_fmt { PROT REST RETR RMD RNFR RNTO SDUP SITE } \
```

```
    chk_str_fmt { SIZE SMNT STAT STOR STRU TEST TYPE USER } \
    chk_str_fmt { XCRC XCWD XMAS XMD5 XMKD XRCP XRMD XRSQ } \
    chk_str_fmt { XSEM XSEN XSHA1 XSHA256 } \
    cmd_validity ALLO < int [ char R int ] > \
    cmd_validity EPSV < [ { char 12 | char A char L char L } ] > \
    cmd_validity MACB < string > \
    cmd_validity MDTM < [ date nnnnnnnnnnnnnn[.n[n[n]]] ] string > \
    cmd_validity MODE < char ASBCZ > \
    cmd_validity PORT < host_port > \
    cmd_validity PROT < char CSEP > \
    cmd_validity STRU < char FRPO [ string ] > \
    cmd_validity TYPE < { char AE [ char NTC ] | char I | char L [ number ] } >
preprocessor ftp_telnet_protocol: ftp client default \
    max_resp_len 256 \
    bounce yes \
    ignore_telnet_erase_cmds yes \
    telnet_cmds yes


# SMTP normalization and anomaly detection.  For more information, see README.SMTP
preprocessor smtp: ports { 25 465 587 691 } \
    inspection_type stateful \
    b64_decode_depth 0 \
    qp_decode_depth 0 \
    bitenc_decode_depth 0 \
    uu_decode_depth 0 \
    log_mailfrom \
    log_rcptto \
    log_filename \
    log_email_hdrs \
    normalize_cmds \
    normalize_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM
ETRN EVFY } \
    normalize_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML
SEND SOML } \
    normalize_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP X-
ERCP X-EXCH50 } \
    normalize_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE
XQUE XSTA XTRN XUSR } \
    max_command_line_len 512 \
    max_header_line_len 1000 \
    max_response_line_len 512 \
    alt_max_command_line_len 260 { MAIL } \
    alt_max_command_line_len 300 { RCPT } \
    alt_max_command_line_len 500 { HELP HELO ETRN EHLO } \
    alt_max_command_line_len 255 { EXPN VRFY ATRN SIZE BDAT DEBUG EMAL ESAM ESND
ESOM EVFY IDENT NOOP RSET } \
    alt_max_command_line_len 246 { SEND SAML SOML AUTH TURN ETRN DATA RSET QUIT
ONEX QUEU STARTTLS TICK TIME TURNME VERB X-EXPS X-LINK2STATE XADR XAUTH XCIR
XEXCH50 XGEN XLICENSE XQUE XSTA XTRN XUSR } \
    valid_cmds { ATRN AUTH BDAT CHUNKING DATA DEBUG EHLO EMAL ESAM ESND ESOM ETRN
EVFY } \
    valid_cmds { EXPN HELO HELP IDENT MAIL NOOP ONEX QUEU QUIT RCPT RSET SAML SEND
SOML } \
    valid_cmds { STARTTLS TICK TIME TURN TURNME VERB VRFY X-ADAT X-DRCP X-ERCP X-
EXCH50 } \
    valid_cmds { X-EXPS X-LINK2STATE XADR XAUTH XCIR XEXCH50 XGEN XLICENSE XQUE
XSTA XTRN XUSR } \
    xlink2state { enabled }

# Portscan detection.  For more information, see README.sfportscan
```

```
 preprocessor sfportscan: proto  { all } memcap { 10000000 } sense_level { low }

# ARP spoof detection.  For more information, see the Snort Manual - Configuring
Snort - Preprocessors - ARP Spoof Preprocessor
# preprocessor arpspoof
# preprocessor arpspoof_detect_host: 192.168.40.1 f0:0f:00:f0:0f:00

# SSH anomaly detection.  For more information, see README.ssh
preprocessor ssh: server_ports { 22 } \
                  autodetect \
                  max_client_bytes 19600 \
                  max_encrypted_packets 20 \
                  max_server_version_len 100 \
                  enable_respoverflow enable_ssh1crc32 \
                  enable_srvoverflow enable_protomismatch

# SMB / DCE-RPC normalization and anomaly detection.  For more information, see
README.dcerpc2
preprocessor dcerpc2: memcap 102400, events [co ]
preprocessor dcerpc2_server: default, policy WinXP, \
    detect [smb [139,445], tcp 135, udp 135, rpc-over-http-server 593], \
    autodetect [tcp 1025:, udp 1025:, rpc-over-http-server 1025:], \
    smb_max_chain 3, smb_invalid_shares ["C$", "D$", "ADMIN$"]

# DNS anomaly detection.  For more information, see README.dns
preprocessor dns: ports { 53 } enable_rdata_overflow

# SSL anomaly detection and traffic bypass.  For more information, see README.ssl
preprocessor ssl: ports { 443 465 563 636 989 992 993 994 995 5061 7801 7802 7900
7901 7902 7903 7904 7905 7906 7907 7908 7909 7910 7911 7912 7913 7914 7915 7916
7917 7918 7919 7920 }, trustservers, noinspect_encrypted

# SDF sensitive data preprocessor.  For more information see README.sensitive_data
preprocessor sensitive_data: alert_threshold 25

# SIP Session Initiation Protocol preprocessor.   For more information see
README.sip
preprocessor sip: max_sessions 40000, \
   ports { 5060 5061 5600 }, \
   methods { invite \
             cancel \
             ack \
             bye \
             register \
             options \
             refer \
             subscribe \
             update \
             join \
             info \
             message \
             notify \
             benotify \
             do \
             qauth \
             sprack \
             publish \
             service \
             unsubscribe \
             prack }, \
   max_uri_len 512, \
```

```
   max_call_id_len 80, \
   max_requestName_len 20, \
   max_from_len 256, \
   max_to_len 256, \
   max_via_len 1024, \
   max_contact_len 512, \
   max_content_len 2048

# IMAP preprocessor.  For more information see README.imap
preprocessor imap: \
   ports { 143 } \
   b64_decode_depth 0 \
   qp_decode_depth 0 \
   bitenc_decode_depth 0 \
   uu_decode_depth 0

# POP preprocessor. For more information see README.pop
preprocessor pop: \
   ports { 110 } \
   b64_decode_depth 0 \
   qp_decode_depth 0 \
   bitenc_decode_depth 0 \
   uu_decode_depth 0

# Modbus preprocessor. For more information see README.modbus
preprocessor modbus: ports { 502 }

# DNP3 preprocessor. For more information see README.dnp3
preprocessor dnp3: ports { 20000 } \
   memcap 262144 \
   check_crc

# Reputation preprocessor. For more information see README.reputation
preprocessor reputation: \
   memcap 500, \
   priority whitelist, \
   nested_ip inner, \
   whitelist $WHITE_LIST_PATH\white.list, \
   blacklist $BLACK_LIST_PATH\black.list

##################################################
# Step #6: Configure output plugins
# For more information, see Snort Manual, Configuring Snort - Output Modules
##################################################

# unified2
# Recommended for most installs
# output unified2: filename merged.log, limit 128, nostamp, mpls_event_types,
vlan_event_types

# Additional configuration for specific types of installs
# output alert_unified2: filename snort.alert, limit 128, nostamp
# output log_unified2: filename snort.log, limit 128, nostamp

# syslog
# output alert_syslog: LOG_AUTH LOG_ALERT

# pcap
# output log_tcpdump: tcpdump.log

# metadata reference data.  do not modify these lines
```

```
include classification.config
include reference.config


##################################################
# Step #7: Customize your rule set
# For more information, see Snort Manual, Writing Snort Rules
#
# NOTE: All categories are enabled in this conf file
##################################################

# site specific rules
include $RULE_PATH/local.rules

include $RULE_PATH\app-detect.rules
include $RULE_PATH\attack-responses.rules
include $RULE_PATH\backdoor.rules
include $RULE_PATH\bad-traffic.rules
include $RULE_PATH\blacklist.rules
include $RULE_PATH\botnet-cnc.rules
include $RULE_PATH\browser-chrome.rules
include $RULE_PATH\browser-firefox.rules
include $RULE_PATH\browser-ie.rules
include $RULE_PATH\browser-other.rules
include $RULE_PATH\browser-plugins.rules
include $RULE_PATH\browser-webkit.rules
include $RULE_PATH\chat.rules
include $RULE_PATH\content-replace.rules
include $RULE_PATH\ddos.rules
include $RULE_PATH\dns.rules
include $RULE_PATH\dos.rules
include $RULE_PATH\experimental.rules
include $RULE_PATH\exploit-kit.rules
include $RULE_PATH\exploit.rules
include $RULE_PATH\file-executable.rules
include $RULE_PATH\file-flash.rules
include $RULE_PATH\file-identify.rules
include $RULE_PATH\file-image.rules
include $RULE_PATH\file-java.rules
include $RULE_PATH\file-multimedia.rules
include $RULE_PATH\file-office.rules
include $RULE_PATH\file-other.rules
include $RULE_PATH\file-pdf.rules
include $RULE_PATH\finger.rules
include $RULE_PATH\ftp.rules
include $RULE_PATH\icmp-info.rules
include $RULE_PATH\icmp.rules
include $RULE_PATH\imap.rules
include $RULE_PATH\indicator-compromise.rules
include $RULE_PATH\indicator-obfuscation.rules
include $RULE_PATH\indicator-scan.rules
include $RULE_PATH\indicator-shellcode.rules
include $RULE_PATH\info.rules
include $RULE_PATH\malware-backdoor.rules
include $RULE_PATH\malware-cnc.rules
include $RULE_PATH\malware-other.rules
include $RULE_PATH\malware-tools.rules
include $RULE_PATH\misc.rules
include $RULE_PATH\multimedia.rules
include $RULE_PATH\mysql.rules
include $RULE_PATH\netbios.rules
```

```
include $RULE_PATH\nntp.rules
include $RULE_PATH\oracle.rules
include $RULE_PATH\os-linux.rules
include $RULE_PATH\os-mobile.rules
include $RULE_PATH\os-other.rules
include $RULE_PATH\os-solaris.rules
include $RULE_PATH\os-windows.rules
include $RULE_PATH\other-ids.rules
include $RULE_PATH\p2p.rules
include $RULE_PATH\phishing-spam.rules
include $RULE_PATH\policy-multimedia.rules
include $RULE_PATH\policy-other.rules
include $RULE_PATH\policy.rules
include $RULE_PATH\policy-social.rules
include $RULE_PATH\policy-spam.rules
include $RULE_PATH\pop2.rules
include $RULE_PATH\pop3.rules
include $RULE_PATH\protocol-dns.rules
include $RULE_PATH\protocol-finger.rules
include $RULE_PATH\protocol-ftp.rules
include $RULE_PATH\protocol-icmp.rules
include $RULE_PATH\protocol-imap.rules
include $RULE_PATH\protocol-nntp.rules
include $RULE_PATH\protocol-other.rules
include $RULE_PATH\protocol-pop.rules
include $RULE_PATH\protocol-rpc.rules
include $RULE_PATH\protocol-scada.rules
include $RULE_PATH\protocol-services.rules
include $RULE_PATH\protocol-snmp.rules
include $RULE_PATH\protocol-telnet.rules
include $RULE_PATH\protocol-tftp.rules
include $RULE_PATH\protocol-voip.rules
include $RULE_PATH\pua-adware.rules
include $RULE_PATH\pua-other.rules
include $RULE_PATH\pua-p2p.rules
include $RULE_PATH\pua-toolbars.rules
include $RULE_PATH\rpc.rules
include $RULE_PATH\rservices.rules
include $RULE_PATH\scada.rules
include $RULE_PATH\scan.rules
include $RULE_PATH\server-apache.rules
include $RULE_PATH\server-iis.rules
include $RULE_PATH\server-mail.rules
include $RULE_PATH\server-mssql.rules
include $RULE_PATH\server-mysql.rules
include $RULE_PATH\server-oracle.rules
include $RULE_PATH\server-other.rules
include $RULE_PATH\server-samba.rules
include $RULE_PATH\server-webapp.rules
include $RULE_PATH\shellcode.rules
include $RULE_PATH\smtp.rules
include $RULE_PATH\snmp.rules
include $RULE_PATH\specific-threats.rules
include $RULE_PATH\spyware-put.rules
include $RULE_PATH\sql.rules
include $RULE_PATH\telnet.rules
include $RULE_PATH\tftp.rules
include $RULE_PATH\virus.rules
include $RULE_PATH\voip.rules
include $RULE_PATH\web-activex.rules
include $RULE_PATH\web-attacks.rules
```

```
include $RULE_PATH\web-cgi.rules
include $RULE_PATH\web-client.rules
include $RULE_PATH\web-coldfusion.rules
include $RULE_PATH\web-frontpage.rules
include $RULE_PATH\web-iis.rules
include $RULE_PATH\web-misc.rules
include $RULE_PATH\web-php.rules
include $RULE_PATH\x11.rules
include $RULE_PATH\emerging-scan.rules
##################################################
# Step #8: Customize your preprocessor and decoder alerts
# For more information, see README.decoder_preproc_rules
##################################################

# decoder and preprocessor event rules
# include $PREPROC_RULE_PATH/preprocessor.rules
# include $PREPROC_RULE_PATH/decoder.rules
# include $PREPROC_RULE_PATH/sensitive-data.rules

##################################################
# Step #9: Customize your Shared Object Snort Rules
#   For   more   information,   see   http://vrt-blog.snort.org/2009/01/using-vrt-
certified-shared-object-rules.html
##################################################

# dynamic library rules
# include $SO_RULE_PATH/browser-ie.rules
# include $SO_RULE_PATH/browser-other.rules
# include $SO_RULE_PATH/exploit-kit.rules
# include $SO_RULE_PATH/file-flash.rules
# include $SO_RULE_PATH/file-image.rules
# include $SO_RULE_PATH/file-java.rules
# include $SO_RULE_PATH/file-multimedia.rules
# include $SO_RULE_PATH/file-office.rules
# include $SO_RULE_PATH/file-other.rules
# include $SO_RULE_PATH/file-pdf.rules
# include $SO_RULE_PATH/indicator-shellcode.rules
# include $SO_RULE_PATH/malware-cnc.rules
# include $SO_RULE_PATH/malware-other.rules
# include $SO_RULE_PATH/netbios.rules
# include $SO_RULE_PATH/os-linux.rules
# include $SO_RULE_PATH/os-other.rules
# include $SO_RULE_PATH/os-windows.rules
# include $SO_RULE_PATH/policy-social.rules
# include $SO_RULE_PATH/protocol-dns.rules
# include $SO_RULE_PATH/protocol-nntp.rules
# include $SO_RULE_PATH/protocol-other.rules
# include $SO_RULE_PATH/protocol-snmp.rules
# include $SO_RULE_PATH/protocol-voip.rules
# include $SO_RULE_PATH/pua-p2p.rules
# include $SO_RULE_PATH/server-apache.rules
# include $SO_RULE_PATH/server-iis.rules
# include $SO_RULE_PATH/server-mail.rules
# include $SO_RULE_PATH/server-mysql.rules
# include $SO_RULE_PATH/server-oracle.rules
# include $SO_RULE_PATH/server-other.rules
# include $SO_RULE_PATH/server-webapp.rules

# legacy dynamic library rule files
# include $SO_RULE_PATH/bad-traffic.rules
# include $SO_RULE_PATH/browser-ie.rules
```

```
# include $SO_RULE_PATH/chat.rules
# include $SO_RULE_PATH/dos.rules
# include $SO_RULE_PATH/exploit.rules
# include $SO_RULE_PATH/file-flash.rules
# include $SO_RULE_PATH/icmp.rules
# include $SO_RULE_PATH/imap.rules
# include $SO_RULE_PATH/misc.rules
# include $SO_RULE_PATH/multimedia.rules
# include $SO_RULE_PATH/netbios.rules
# include $SO_RULE_PATH/nntp.rules
# include $SO_RULE_PATH/p2p.rules
# include $SO_RULE_PATH/smtp.rules
# include $SO_RULE_PATH/snmp.rules
# include $SO_RULE_PATH/specific-threats.rules
# include $SO_RULE_PATH/web-activex.rules
# include $SO_RULE_PATH/web-client.rules
# include $SO_RULE_PATH/web-iis.rules
# include $SO_RULE_PATH/web-misc.rules

# Event thresholding or suppression commands. See threshold.conf
include threshold.conf
```