МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ
КАФЕДРА СИСТЕМ ІНФОРМАЦІЙНОГО ТА КІБЕРНЕТИЧНОГО ЗАХИСТУ

«На правах рукопису»
УДК 004.12:681.06

«До захисту допущено»
Завідуючий кафедрою СІКЗ
_____ к.т.н. Г.В. Шуклін
«____» _____ 2022 р.

БАКАЛАВРСЬКА АТЕСТАЦІЙНА РОБОТА

зі спеціальності 125 “Кібербезпека”

на тему: «МЕТОДИКА ЗАХИСТУ КОНФІДЕНЦІЙНОЇ ІНФОРМАЦІЇ БАЗ ДАНИХ ВІД SQL-АТАК»

студент групи СЗД-41 Радоніч Нікола _____
(підпис)

Науковий керівник: к.т.н., доцент Пепа Юрій Володимирович _____
(підпис)

Нормоконтроль: Гребенніков Асаді Болдхоядович _____
(підпис)

КИЇВ – 2022

«ЗАТВЕРДЖУЮ»

Завідувач кафедри СІКЗ

_____ к.т.н. Г.В. Шуклін

(підпис)

«_____»_____ 2022 р.

ЗАВДАННЯ

на атестаційну роботу магістра

студенту: Радонічу Ніколі

1. Тема роботи: «методика захисту конфіденційної інформації баз даних від SQL-атак», затверджена наказом по університету від « ___ » _____ 2022 р. за № _____ .

2. Термін здачі студентом оформленої роботи « 01 » червня 2022 р.

3. Об'єкт дослідження: база даних з порушеною структурою.

4. Предмет дослідження: виявлення та виправлення порушень цілісності схеми бази даних.

5. Мета роботи: розробка методики виявлення та виправлення порушень цілісності схеми бази даних та блокування стороннього втручання на основі відповідного програмного забезпечення.

6. Перелік питань, які мають бути розроблені:

1. Структури баз даних, їх властивості і схеми будови;
2. Визначення основних вразливостей і видів SQL-атак на бази даних в IT-сегменті;
3. Створення системного підходу до виявлення та протидії втраті чи підміні інформації в структурах бази даних.

7. Перелік публікацій:

8. Перелік ілюстративного матеріалу: Презентація виконана на слайдах для подання за допомогою світлопроектору та комп'ютерних засобів.

9. Дата видачі завдання « 16 » лютого 2022 р.

КАЛЕНДАРНИЙ ПЛАН

Дата видачі завдання «16» лютого 2022 р.

№ з/п	Назва етапів атестаційної роботи	Строк виконання етапів роботи	Примітка
1	Огляд літератури	до 29.03.22 р.	виконано
2	Написання першого розділу роботи	до 10.04.22 р.	виконано
3	Написання другого розділу роботи	до 27.04.22 р.	виконано
4	Написання третього розділу роботи	до 08.05.22 р.	виконано
5	Оформлення атестаційної роботи	до 16.05.22 р.	виконано
6	Підготовка демонстраційних матеріалів	до 28.05.22 р.	виконано

Студент: СЗД-41 Радоніч Н. _____
(підпис)

Науковий керівник: к.т.н., доц. Пепа Ю.В. _____
(підпис)

Нормоконтроль: Гребенніков А.Б. _____
(підпис)

РЕФЕРАТ

Атестаційна робота містить: 65 сторінок, 17 рисунків, 3 таблиці.

Запропонований підхід є повним рішенням, що дозволяє не тільки виявляти порушення схеми бази даних, але і протидіяти, чого не існує наразі зараз. Даний підхід є комплексним і залежить від способу будови бази даних, її розташуванню на сервері, організації запитів і підтверджень, а також аналізу сторонніх впливів та методів протидії втраті чи підміні інформації у SQL.

Метою роботи є розробка методики виявлення та виправлення порушень цілісності схеми бази даних та блокування стороннього втручання на основі відповідного програмного забезпечення.

Завдання роботи:

1. Проаналізувати існуючі типи баз даних, їх основні аспекти, загрози на бази даних, атаки і їх різновиди та наслідки для реляційних баз даних.

2. Ознайомитися з існуючими методами виявлення та виправлення порушень бази даних, проаналізувати їх недоліки.

3. Реалізувати протидію та виявлення вторгнень на основі сучасного програмного забезпечення.

Об’єктом дослідження є база даних структура якої змінюється несанкціонованим шляхом.

Предметом дослідження є виявлення та виправлення порушень цілісності схеми бази даних.

Методи дослідження – ознайомлення та опрацювання літератури, аналіз та порівняння, системний підхід до вирішення складного завдання.

Галузь використання – кібербезпека та інформаційні технології.

Ключові слова: БАЗА ДАНИХ, СЕРЕДОВИЩЕ, SQL-ЗАПИТИ, WEB-СЕРВЕР, FTP-СЕРВЕР, СЕРВІСИ ВІДДАЛЕНОГО ДОСТУПУ, ЗАГРОЗИ, ІНФОРМАЦІЙНА БЕЗПЕКА, ІН’ЄКЦІЇ.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

БД – база даних

ПЗ – програмне забезпечення

СУБД – система управління базами даних

PHP – Hypertext Preprocessor

HTML – HyperText Markup Language

CMS – Content Management System

SQL – Structured Query Language

SQLIA – Structured Query Language Injection Attack

SPARQL – Protocol and RDF Query Language

QL – Query Language

PL – Programming Language

DoS – Denial of Service

XML – eXtensible Markup Language

SMTP – Simple Mail Transfer Protocol

ВСТУП

Один з найважливіших критеріїв надійності інформаційної системи – безпека баз даних системи. Атаки, спрямовані на неї, в більшості випадків критичні, тому що можуть частково або повністю порушити працездатність системи. Однією з найлегших та найпоширеніших атак є атака по впровадженню програмного коду. SQL-ін'єкція становить значну частину (приблизно 25%) всіх мережевих атак. Тому, існує потреба в сучасних методах захисту від цього типу атак, а також, в разі, все ж таки, несанкціонованих змін – методах виявлення та методах реагування на зміни в БД, тому робота присвячена аналізу наслідків атак та методів виявлення та реагування на дані атаки.

В даний час, у більшості організацій використовуються локальні обчислювальні мережі. Чим більше організація, тим більш розгалужена та складніша мережа. З ростом організації росте обчислювальна мережа, переходячи в більш територіально рознесену, з'являються сервера і клієнти віддаленого доступу, контролери доменів, файл-сервера, сервера баз даних, ускладнюється активне мережеве устаткування, додаються протоколи мережевої і міжмережевої взаємодії, будуються VLAN, налагоджуються канали Frame-Relay і VPN, уся ця структура зветься – корпоративна мережа.

Одним з найпоширеніших способів обмежити взаємодію корпоративної мережі з зовнішнім світом є побудова захисту безпосередньо на кожній клієнтській машині (хості) за допомогою персонального антивірусу або недорогого персонального брандмауера і файєрвола.

Сама топологія корпоративних мереж майже завжди залишає бажати кращого, частенько використовуються застарілі мережеві топології (наприклад 10base-T, TokenRing), мережеве активне устаткування (концентратори невідомих фірм, та застаріли моделі маршрутизаторів).

Порушнику досить легко пробитися в таку мережу використовуючи безліч відомих вразливостей, неточності в написанні програмного забезпечення, недбалість обслуговуючого персоналу, відсутність грамотно описаної процедури протидії порушнику.

1 МЕТОДИ ВИЯВЛЕННЯ ЗМІН В БАЗАХ ДАНИХ

1.1 Поняття бази даних

База даних (БД) – це організована структура, призначена для зберігання, зміни і обробки взаємозалежної інформації, переважно великих обсягів. БД активно використовуються для динамічних сайтів зі значними обсягами даних – це інтернет-магазини, портали, корпоративні сайти та ін. Такі сайти зазвичай розроблені за допомогою серверної мови програмування (як приклад, PHP) або на основі CMS (як приклад, WordPress) і не мають готових сторінок з даними за аналогією з HTML-сайтами. Сторінки динамічних сайтів формуються в результаті взаємодії скриптів і БД після відповідного запиту клієнта до веб-сервера.

У визначеннях присутні такі відмінні ознаки:

1. БД зберігається і обробляється в обчислювальній системі.

Таким чином, будь-які некомп’ютерні сховища інформації (архіви, бібліотеки, картотеки та ін.) БД не є.

2. Дані в БД логічно структуровані (систематизовані) з метою забезпечення можливості їх ефективного пошуку і обробки в обчислювальній системі.

Структурованість передбачає явне виділення складових частин (елементів), зв'язків між ними, а також типізацію елементів і зв'язків, при якій з типом елемента (зв'язку) співвідноситься певна семантика і допустимі операції.

3. БД включає схему або метадані, що описують логічну структуру БД в формальному вигляді (відповідно до деякої метамоделі).

Постійні дані в середовищі БД включають в себе схему і саму БД. Схема включає в себе опис змісту, структури і обмежень цілісності, що використовуються для створення і підтримки БД. БД включає в себе набір постійних даних відповідно до схеми. Система управління даними

використовує визначення даних в схемі для забезпечення доступу і управління доступом до даних в БД.

В контексті баз даних варто розглянути поняття СУБД. Система управління базами даних (СУБД) – це комплекс програмних засобів, необхідних для створення структури нової бази, її наповнення, редагування вмісту і відображення інформації. Найбільш поширеними СУБД є MySQL, PostgreSQL, Oracle, Microsoft SQL Server. СУБД служить інтерфейсом між БД та її кінцевими користувачами або програмами, дозволяючи користувачам отримувати, оновлювати та керувати способом організації та оптимізації інформації. СУБД також полегшує контроль БД, що дозволяє здійснювати різні адміністративні операції, такі як моніторинг продуктивності, налаштування, резервне копіювання та відновлення.

У реляційній БД цифрова інформація про конкретного клієнта впорядковується у рядки, стовпці та таблиці, які індексуються, щоб полегшити пошук відповідної інформації за допомогою SQL запитів. На відміну від цього, графічна БД використовує вузли та ребра для визначення зв’язків між записами даних і запитами, потребує спеціального синтаксису семантичного пошуку. Станом на сьогодні, SPARQL – єдина семантична мова запитів, яка затверджена Всесвітнім консорціумом веб-сторінок (W3C).

Зазвичай менеджер БД надає користувачам можливість контролювати доступ для читання/запису, задавати генерацію звітів та аналізувати використання. Деякі БД пропонують відповідність ACID (атомарність, послідовність, ізоляцію та довговічність), щоб гарантувати відповідність даних та завершення транзакцій.

1.2 Типи баз даних

БД еволюціонували з часу їх створення в 1960-х роках, починаючи з ієрархічних та мережевих БД, до 1980-х років з об'єктно-орієнтованими БД, а сьогодні – з БД SQL і NoSQL та хмарними БД.

З одного погляду, БД можна класифікувати за типом вмісту: бібліографічний, повний текст, числовий та зображення. При обчислювальній роботі БД іноді класифікуються відповідно до їх організаційного підходу. Існує багато різних типів БД, починаючи від найбільш поширеного підходу, реляційної БД, до розподіленої БД, хмарної БД, графічної БД або БД NoSQL.

Розглянемо наступні типи БД:

1. Реляційна БД.

Реляційна БД, винайдена Е.Ф. Коддом в IBM в 1970 р., – це таблична БД, в якій дані визначаються таким чином, щоб вони могли бути реорганізовані та доступні різними способами. Реляційні БД складаються з набору таблиць з даними, що вписуються у заздалегідь задану категорію. Кожна таблиця містить щонайменше одну категорію даних у стовпці, а кожен рядок має певний екземпляр даних для категорій, визначених у стовпцях. Реляційні БД легко розширити, а нову категорію даних можна додати після створення оригінальної БД, не вимагаючи зміни всіх існуючих програм. Реляційна БД має як свої переваги, так і недоліки.

Однією з важливих переваг реляційного підходу є його простота і доступність для розуміння кінцевим користувачем. Єдиною інформаційної конструкцією є таблиця.

Ще однією важливою перевагою та особливістю є те, що реляційна БД має строгу статичну типізацію, тобто, наприклад, після ключових слів create table завжди йде назва таблиці, що буде створена, або ж після назви таблиці в дужках ідуть ім’я колонки та її тип даних, що перераховуються через кому. Дана особливість є ключовою для подальшого процесу з даними типом БД. Інші ж БД не притримуються даним суворим форматом.

При проектуванні реляційних БД застосовуються суворі правила, що базуються на математичному апараті. Реляційна модель забезпечує повну незалежність даних. При зміні структури реляційної БД зміни, які потрібно зробити в прикладних програмах, як правило, мінімальні.

2. Розподілена БД.

Розподілена БД – це БД, в якій частини БД зберігаються в декількох фізичних місцях, і в якій обробка розповсюджується або реплікується між різними точками мережі.

Розподілені БД можуть бути однорідними або неоднорідними. Всі фізичні розташування в однорідній системі розподілених БД мають одне і те ж саме апаратне забезпечення і працюють однакові операційні системи та програми БД. Апаратне забезпечення, операційні системи або програми БД в неоднорідній розподіленій БД можуть бути різними в кожному з розташувань.

Також розподілені БД можуть бути фрагментовані або тиражовані. Фрагментована або секціонована (англ. Partitioned database) – це БД, в якій методом розподілу даних є фрагментованість (секціонування), вертикальне чи горизонтальне. Тиражована (англ. Replicated database) – це БД, в якій методом розподілу даних є тиражування (реплікація).

3. Хмарна БД.

Хмарна БД – це БД оптимізована або побудована для віртуалізованого середовища або в гібридній хмарі, у відкритій або в приватній хмарі. Хмарні БД надають такі переваги, як можливість платити за обсяг сховища і пропускну здатність для кожного використання, а також забезпечують масштабованість за потребою, а також високу доступність.

Хмарна БД також надає підприємствам можливість підтримувати бізнес-застосунки в розгортанні програмного забезпечення.

4. База даних NoSQL.

БД NoSQL корисні для великих наборів розподілених даних. Вони ефективні для проблем з великими показниками продуктивності даних, які не розроблені для реляційних БД. Вони найбільш ефективні, коли організація повинна проаналізувати великі шматки неструктурованих даних або даних, які зберігаються на декількох віртуальних серверах у хмарі. NoSQL – це нова категорія систем управління БД. Його основною характеристикою є

недотримання концепцій реляційних БД. NoSQL означає «не тільки SQL». Концепція БД NoSQL виросла з інтернет-гігантами, такими як Google, Facebook, Amazon та ін., які мають справу з гігантськими обсягами даних. Коли використовується реляційна БД для величезних обсягів даних, система починає сповільнюватися з точки зору часу відгуку. Щоб подолати це, ми, звичайно, могли б «розширити» наші системи, модернізуючи наше існуюче обладнання. Альтернативою вищевказаної проблеми був би розподіл навантаження на нашу БД на кілька хостів по мірі збільшення навантаження. Це відомо як «масштабування». БД NoSQL – це нереляційні БД, які масштабуються краще, ніж реляційні БД, і розробляються з урахуванням веб-додатків. Вони не використовують SQL для запиту даних і не слідують суворим схемам, таким як реляційні моделі. З NoSQL функції ACID (атомарність, узгодженість, ізоляція, довговічність) не завжди можуть бути гарантовані.

5. Об'єктно-орієнтована БД.

Об'єктно-орієнтована СУБД – цей тип підтримує зберігання нових типів даних. Дані, що підлягають збереженню, мають форму об'єктів. Об'єкти, що зберігаються в БД, мають атрибути (тобто стать, вік) та методи, що визначають, що робити з даними. PostgreSQL – приклад об'єктно-орієнтованої реляційної СУБД.

Переваги використання СУБД:

1) Відсутня проблема невідповідності моделі даних в веб-застосунку і БД (impedance mismatch). Всі дані зберігаються в БД в тому ж вигляді, що і в моделі застосунку;

2) Не потрібно окремо підтримувати модель даних на стороні СУБД;

3) Всі об'єкти на рівні джерела даних строго типізовані. Більше ніяких строкових імен колонок. Рефакторинг об'єктно-орієнтованої БД і працюючого з нею коду тепер автоматизований, а не одноманітний і нудний процес.

6. Графо-орієнтована БД.

Графо-орієнтована БД або графова БД, є типом БД NoSQL, яка використовує теорію графів для зберігання, відображення і запиту взаємозв'язків. Графо-орієнтована БД – це, в основному, набори вузлів і ребер, де кожен вузол являє сутність, а кожне ребро являє зв'язок між вузлами.

Графо-орієнтовані БД стають все більш популярними для аналізу взаємозв'язків. Наприклад, компанії можуть використовувати графічну БД для збору даних про клієнтів з соціальних мереж.

Графо-орієнтовані БД часто використовують SPARQL, декларативну мову програмування і протокол для аналізу графових БД. SPARQL має можливість виконувати всю аналітику, яку може виконувати SQL, а також може використовуватися для семантичного аналізу, вивчення відносин. Це робить його корисним для виконання аналітики наборів даних, які мають як структуровані, так і неструктуровані дані. SPARQL дозволяє користувачам виконувати аналітику інформації, що зберігається в реляційній БД, а також відносин один-одного (FOAF), PageRank і найкоротшого шляху.

1.3 Визначення SQL

SQL (мова структурованих запитів) – це мова високого рівня, основа якої сильно залежить від реляційної алгебри і реляційного числення. SQL складається з декларативних елементів, таких як запити, вирази, пропозиції, оператори і т.ін. SQL широко відома як потужна мова запитів. Основна відмінність між мовою запитів (QL) і мовою програмування (PL) полягає в тому, що QL не можна використовувати для складних обчислень, і вона має дуже хорошу ефективність для обробки великих наборів даних. Реалізація мови запитів заснована на реляційній алгебрі (операційна частина) і реляційному численні (декларативна частина). Реляційна алгебра відноситься до специфікації послідовності операцій для виконання певного запиту, тоді як реляційне числення відноситься до специфікації необхідного висновку без

будь-якої інформації про послідовність операцій, необхідної для обробки запиту.

Подібно до інших алгебр, деякі оператори є примітивними, а інші, будучи визначені через примітивні, є похідними від них. В реляційній алгебрі Кодда визначено таких шість примітивних операторів: вибірка, проекція, декартів добуток, об'єднання та різниця і перейменування (насправді, Кодд відмовився від включення оператора перейменування, однак, розробники ISBL навели приклади необхідності його включення). Шість операторів є фундаментальними в тому сенсі, що жоден із них не можна відкинути без втрати потужності. Багато інших операторів було визначено комбінацією цих шести. Серед найважливіших можна назвати: перетин множин, ділення та природне об'єднання. Насправді, ISBL дала підстави для заміни декартового добутку природнім об'єднанням, окремим випадком якого є декартів добуток. У табл. 1.1 наведено кілька прикладів примітивних операторів, які використовуються при розробці оператора запиту.

Таблиця 1.1

Приклад примітивних операторів

Вибірка	Вибирає підмножину рядків в таблиці
Проекція	Видаляє атрибути з таблиць, яких немає в списку. При виконанні проекції виділяється "вертикальна" вирізка відносини-операнда з природним знищенням потенційно виникаючих кортежів-дублікатів.
Декартів добуток	Дозволяє поєднання реляційного пошуку
Встановлення різниці	Визначає взаємовиключні зв’язки
Об'єднання	Відношення з тим же заголовком, що і у сумісних по типу відносин A і B, і тілом, що складається з кортежів, що належать або A, або B, або обом відносинам. Синтаксис: A UNION B

Додаткові оператори, такі як перетин, поєднання, ділення і перейменування, також є корисні.

Підводячи підсумок, можна сказати, що це реляційна модель, яка строго визначена простотою і здатністю операційної алгебри ефективно виконувати всі завдання, пов'язані з базою даних, з найкращою можливою оптимізацією. В результаті, SQL є мовою спілкування для доступу до систем БД.

Після розгляду визначень SQL і пов'язаних з ними термінів розглянемо базовий метод обробки в мережі і обговоримо архітектуру необхідну для обробки SQL. Опишемо обробку веб-застосунку в наступній послідовності:

1. Після того як клієнт (звичайний веб-користувач, підключений до Інтернету) вводить веб-адресу в застосунку веб-браузера, сервер (комп'ютер, на якому зберігаються веб-сторінки, сайти або додатки) відправляє копію форми клієнту.

2. Користувачі веб-браузера вводять дані в цю форму і відправляють їх на сервер.

3. Сервер запускає сценарій для форми і, коли він визначає, що це доречно, надає клієнтові доступ до основного застосунку або до БД.

Одним із прикладів архітектури, необхідної для обробки веб-застосунку, є архітектура, що показана на рис. 1.1:

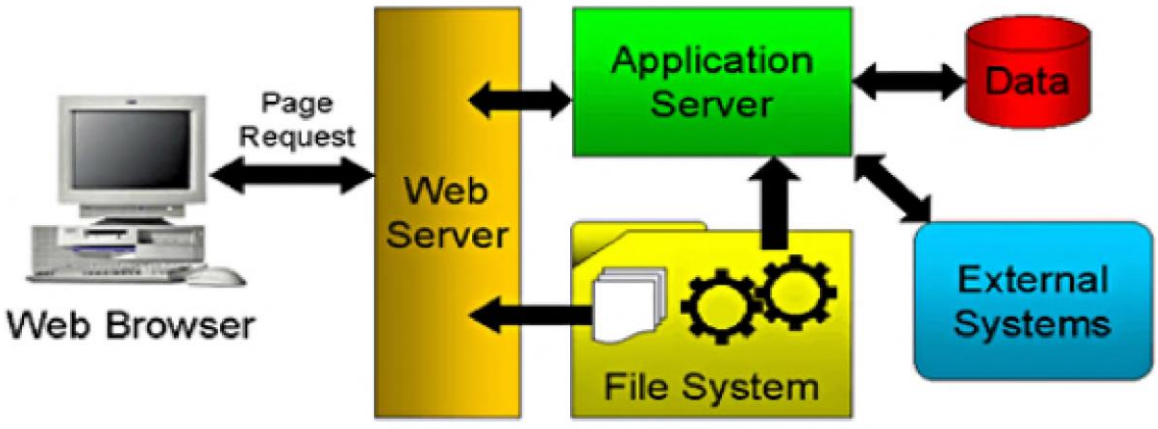

Рисунок 1.1 – Канонічна веб-архітектура

Дана веб-архітектура в цілому поділяється на три рівні обробки. Веб-браузер на рис. 1.1 представляє рівень представлення (показу, презентації). Веб-сервер, сервер застосунків і файлова система складають рівень сервера. Нарешті, дані представляють рівень БД системи веб-застосунків. Функції кожного з цих рівнів описані в і узагальнені нижче:

1. Рівень презентації: містить діалог з користувачами системи. Він має справу з користувацьким введенням, відповіддю від системи і є лицем веб-служби для користувача.

2. Рівень сервера: це основа всієї системи баз даних, оскільки вона містить логіку, необхідну для роботи бази даних.

3. Рівень БД – це точка зберігання, де зберігаються всі дані.

Заходи безпеки можуть бути реалізовані на рівні презентації (застосунок), на рівні сервера (простір сервера) або на рівні БД (фізичне розташування БД).

Захисні міри, які можуть застосовуватися до веб-застосунків, можуть включати виправлення програмного забезпечення для процесів збору даних на рівні презентації, шифрування даних на рівні БД. Однак у багатьох випадках заходи безпеки на рівні сервера системи веб-додатків особливо підходять для запобігання різних типів атак.

1.4 Існуючі загрози на бази даних

На даний момент існує велика кількість атак на БД. Вони проводяться з метою отримання важливої інформації, для досягнення результату можуть бути використані адміністратор БД. За даними OWASP, основні 10 з них описані в та наведені нижче:

1. Помилки конфігурації хмарної БД.

Помилки конфігурації хмарної БД викликані небезпечно налаштованими хмарними БД або службами зберігання. IP-адреси служби Public Cloud не є секретними і постійно перевіряються на наявність вразливостей зловмисниками і дослідниками безпеки. Багато порушень

пов'язані зі сховищами даних, про які організації не знали, чи які були створені небезпечно на неконтрольованій основі.

2. SQL-ін'єкція.

Уразливості SQL-ін'єкцій виникають, коли код додатка містить динамічні запити до БД, які безпосередньо включають вводимі користувачем дані. Це руйнівна форма атаки і тестери BSI Penetration регулярно знаходять вразливі додатки, які дозволяють повністю обійти аутентифікацію і витягти всю БД.

3. Слабка аутентифікація.

Слабка аутентифікація має багато аспектів, від перебору паролів до небезпечного зберігання облікових даних БД, використовуваних застосунком. Дана загроза може існувати через потребу часто міняти паролі, що може призвести до того, що користувач буде використовувати паролі, що легко запам'ятовуються і відповідно передбачувані паролі. Причиною може бути відсутність багатофакторної аутентифікації та зберігання паролів у відкритому вигляді.

4. Зловживання привілеями.

Користувачі можуть зловживати законними правами доступу до даних в несанкціонованих цілях. Наприклад, користувач в продажах з правами на перегляд окремих записів клієнтів може використовувати цей привілей для вилучення всіх записів клієнтів для передачі конкуренту. Належна політика найму зменшить ймовірність цього, але вона повинна забезпечуватися технічними заходами і ефективною реєстрацією і моніторингом для виявлення зловживань.

5. Надмірні привілеї.

Якщо користувачі мають привілеї, які перевищують вимоги їх посадових функцій, ці привілеї можуть бути порушені окремою особою або зловмисником, який скомпрометує їх обліковий запис. Коли люди переміщують ролі, їм можуть бути надані нові привілеї, в яких вони потребують, а ті, які їм більше не потрібні, будуть видалені.

6. Неповноцінна реєстрація і слабкий аудит.

Реєстрація та аудит є ключовими для запобігання та виявлення неправомірного використання і забезпечення адекватного розслідування підозрюваних компрометації даних. У цьому контексті ведення журналу – це збір даних, а аудит – це той, хто насправді дивиться на це. Необхідно думати, як ваші дані журналу будуть захищені. Якщо все це знаходиться в БД програми і є скомпрометованим, зловмисник може стерти або підробити дані журналу. Журнали можуть містити конфіденційну інформацію, що є небезпечним.

7. Відмова в обслуговуванні.

DoS (відмова в обслуговуванні) – хакерська атака на обчислювальну систему з метою довести її до відмови, тобто створення таких умов, при яких сумлінні користувачі системи не зможуть отримати доступ до надаваних системних ресурсів (серверів), або цей доступ буде утруднений. DoS викликається просто: або база заповнюється «сміттєвими» записами, або, що набагато небезпечніше, вона просто видаляється. Другий випадок особливо цікавий, якщо з яких-небудь причин не робилися (або не перевірялися) бекапи.

8. Неправильне управління виправленнями.

Дефекти які виникають у всіх типах програмного забезпечення, і операційних системах, таких як Windows, а також у СУБД, таких як SQL Server, не є винятком. Якщо говорити про операційну систему і СУБД, можна сказати, що вони містять помилки, і ці помилки можуть бути використані людьми зі зловмисними намірами. Якщо зловмисник знайде помилку, яка викликає уразливість, то він зможе отримати доступ до вашого сервера. І як тільки ця вразливість публікується, що часто відбувається в той момент, коли виробник програмного забезпечення робить виправлення доступним, його можуть використовувати всі. Гірше того, незабаром після публікації уразливості звичайні інструменти автоматизації злому починають включати в себе експлойти для її усунення. Як тільки це відбудеться, все, навіть люди з

невеликим досвідом хакерства або без нього, зможуть використовувати цю уразливість проти вас.

9. Remote Code Execution (RCE).

Є, мабуть, найнебезпечнішим вектором атаки. Зазвичай виконується з метою отримання shells і, отже, контролю над сервером цілком. Часто RCE здійснюється вже після атаки Privilege Escalation і через слабкі налаштування прав доступу в системі, але це відбувається не завжди так. Для реалізації атаки зловмисник завантажує файл-шкідник і або запускає його віддалено, або сам одним з доступних чином «чіпляється» до нього.

Наслідки можуть бути різними. Хтось починає використовувати сервер для майнінга криптовалюти, хтось може налаштувати реплікацію БД на «свій» сервер, а хтось просто піде далі і намагатиметься отримати управління іншими серверами в локальній мережі.

10. Небезпечне резервне копіювання.

Крадіжка стрічок з резервними копіями БД і жорстких дисків вже давно викликає занепокоєння, але виникли нові загрози доступності даних, і їх не можна ігнорувати. Всі резервні копії повинні бути зашифровані для захисту конфіденційності та цілісності даних, і це повинно включати належне управління ключами. Ключі не повинні потрапляти в чужі руки, але повинні бути доступні при необхідності для відновлення даних. Якщо ж говорити про стійкість в хмарних сервісах, наприклад, гео-реплікація, не те ж саме, що резервне копіювання. Зловмисник може видалити стільки хмарної інфраструктури і даних про клієнтів, що організація не зможе вижити.

1.5 Code injection атаки

Впровадження коду – це загальний термін для типів атак, які складаються з введення коду, який потім інтерпретується/виконується застосунком. Цей тип атаки використовує погану обробку ненадійних даних і стає можливими через відсутність належної перевірки даних введення/виводу, наприклад:

- дозволені символи (стандартні класи регулярних виразів або призначені для користувача);

- формат даних;

- кількість очікуваних даних.

Включення коду може бути використане зловмисником для введення (включення) коду в комп'ютерну програму, щоб змінити хід її виконання. Наприклад, включення коду використовується для поширення комп'ютерних хробаків.

Включення коду трапляється тоді, коли програма надсилає неперевірені дані інтерпретатору. Недоліки включення коду дуже поширені в унаслідуваному коді. Вони часто трапляються у SQL, LDAP, Xpath, або NoSQL запитах; командах операційної системи; синтаксичних аналізаторах XML, заголовках SMTP, аргументах програми. Включення коду легко виявити при перегляді коду, проте його дуже важко виявити тестуванням. Сканери та фузери допомагають зловмисникам виявляти вразливості включення коду. Включення коду може призвести до пошкодження чи втрати даних, відсутності звітності або відмови в доступі. Інколи включення коду може призвести навіть до зміни хосту.

Деякі типи включення коду призводять до помилок інтерпретації, надаючи спеціальне значення простому вводу користувача. Це чимось схоже на нездатність розрізняти імена і звичайні слова. За тим же принципом в деяких видах вставленого коду важко розрізнити ввід користувача і системні команди.

Техніка включення коду є поширеною при зломі з метою отримання інформації, отриманні привілейованого або анонімного доступу до системи. Включення коду можна використовувати у зловмисних цілях, зокрема:

- довільно змінювати вміст бази даних через так звані SQL ін’єкції. Наслідком може бути як порушення роботи сайту так і компроментація конфіденційних даних;

- встановлення шкідливих програм або виконання шкідливого коду на сервері через включення скрипт коду сервера (наприклад, PHP чи ASP);

- отримання доступу до кореневої папки, використовуючи вразливості включення Shell;

- атаки інтернет-користувачів за допомогою включення HTML/Script (міжсайтовий скриптинг). Користувачі можуть і не знати, що вони роблять включення коду, бо їхній ввід не був врахований розробниками системи. Наприклад:

- коректні вхідні дані (на думку користувача) можуть містити марковані символи або слова, що були зарезервовані програмістом для певних значень (це може бути символ "&" в назві компанії або символ "");

- користувач може надіслати файл невірного формату як вхідні дані. І хоч цей файл працює коректно, він заразить системі, яка отримує файл.

Основні типи Code injection атак.

1. Міжсайтовий скриптинг.

Міжсайтовий скриптинг (XSS) – це атака з використанням коду, коли шкідливий код впроваджується на веб-сайт і виконується в браузері. Зловмисник вставляє скрипт в браузер жертви, і при доступі до веб-сайту скрипт потім виконується на комп'ютері жертви. Через XSS зловмисник може дистанційно керувати браузером атакованого об'єкта. Це спосіб обійти концепцію стандартної робочої процедури (SOP). Наприклад, всякий раз, коли HTML-код генерується динамічно, а користувацький ввод не очищується і відображається на сторінці, зловмисник може вставити свій HTML-код на цю сторінку.

2. Підробка міжсайтових запитів.

Це атака, при якій зловмисник обманює користувача, виконуючи дії, які корисні зловмисникові при доступі до веб-застосунку. Уразливість CSRF дозволяє зловмисникові змусити користувача виконувати дії, які зловмисник хоче виконати, коли користувач заходить на веб-сайт. Наприклад, злочинець може підробити запит на переказ коштів на веб-сайт.

3. Shell ін'єкції.

Shell ін'єкції названі так завдяки командній оболонці Linux, але це стосується всіх операційних систем, які дозволяють запуск програм з командного рядка. Типові функції, пов'язані з shell ін'єкціями: system(), StartProcessQ, і System.Diagnostics.Process.StartQ.

4. SQL ін'єкції.

SQL ін’єкція – це атака, що є різновидом атаки з впровадженням коду. Цей експлойт виконується шляхом додавання коду SQL у вхідні дані користувача для отримання доступу до неавторизованих ресурсів. SQLIA можуть виникати, коли запит створюється шляхом об'єднання введених користувачем даних, таких як дані, введені в веб-форму, з ненавмисними даними, включаючи дані URL (Uniform Resource Locator), дані, отримані з файлів cookie і т.ін., без належної перевірки. SQL-ін'єкція є однією з улюблених атак для багатьох кіберзлочинців, тому що вона може бути виконана віддалено. Комерційно доступні інструменти виявлення вразливостей також доступні зловмисникам, і за допомогою цих ресурсів зловмисник може знайти лазівки в системі безпеки і веб-уразливості за долі секунди. SQL є дуже гнучкою мовою і ці атаки можуть бути надзвичайно скритними, а також можуть проходити через брандмауери і системи запобігання вторгнень без особливих зусиль.

1.6 SQL injection атаки

Відносна поширеність деяких з цих атак показана на рис. 1.2. Ці дані були отримані від HACKING & TRICKS блогу про злам і комп'ютерну безпеку.

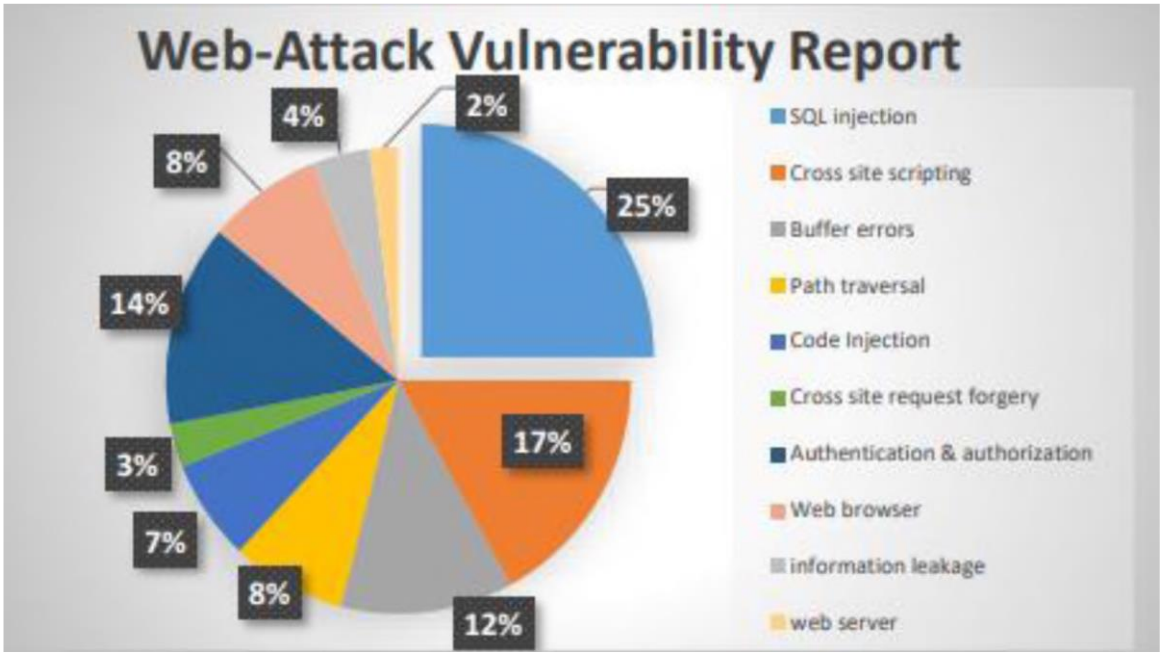

Рисунок 1.2 – Звіт про вразливості веб-атак

Як вже зазначалось вище, SQL injection є підмножиною code injection атаки і використовує переваги синтаксису SQL для включення команд, які можуть читати чи змінювати БД або змінити значення оригінального запиту. Впровадження SQL, залежно від типу СУБД та умов впровадження, може дати можливість атакуючому виконати довільний запит до БД (наприклад, прочитати вміст будь-яких таблиць, видалити, змінити або додати дані), отримати можливість читання та/або запису локальних файлів та виконання довільних команд на сервері.

Атака типу впровадження SQL може бути можлива за некоректної обробки вхідних даних, що використовуються в SQL-запитах. Розробник застосунків, що працюють з БД, повинен знати про таку вразливість і вживати заходів протидії впровадженню SQL.

Для прикладу, розглянемо веб-сторінку, що має два поля для введення імені користувача і пароля. Насправді код сторінки згенерує SQL запит, щоб перевірити існує такий користувач і чи належний пароль він ввів:

SELECT UserList.Usemame
FROM UserList
WHERE UserList.Username = 'Username'
AND UserList.Password = 'Password'

Якщо запит повертає рядки, то доступ надається. Однак, якщо зловмисник введе валідне ім'я користувача і валідний код ("password' OR '1'='1') в поле "Password", тоді запит буде мати вигляд:

SELECT UserList.Username
FROM UserList
WHERE UserList.Username = 'Username'
AND UserList.Password = 'password' **OR** '1'='1'

В цьому прикладі, припускається, що поле "Password" є пусте або містить нешкідливий рядок символів. Вираз '1'='1' завжди буде істинним, тому багато рядків повернуться, тим самим надаючи доступ.

Ця техніка може бути удосконалена, дозволяючи наприклад записувати декілька виразів або навіть завантажувати і запускати зовнішні програми.

Основні типи SQL injection атак.

Типи SQL injection атак наведені нижче:

1. Класична SQL Injection – проста і легка в експлуатації. Дозволяє зловмисникові атакувати БД і відразу бачити результат атаки. Останнім часом зустрічається нечасто. Можливість проведення класичної SQL-ін'єкції багато в чому спрощує отримання корисної інформації. Проведення атаки з використанням класичної техніки експлуатації SQL injection відбувається з використанням оператора union або з використанням поділу SQL запитів (крапка з комою). Але не завжди вразливість типу SQL injection можливо експлуатувати подібним способом. У таких випадках вдаються до техніки експлуатації уразливості «сліпим» методом.

2. Сліпа SQL-ін'єкція (blind SQL Injection) – з'являється в тому випадку, коли вразливий запит є деякою логікою роботи програми, але не дозволяє вивести будь-які дані в повертаєму сторінку Web застосунком. Сліпу SQL-ін'єкцію за своїми можливостями можна порівняти з класичною технікою впровадження операторів SQL. Аналогічно класичній техніці експлуатації подібних вразливостей, blind SQL Injection дозволяє записувати і читати файли, отримувати дані з таблиці, але тільки читання в даному випадку здійснюється посимвольно. Класична техніка експлуатації подібних вразливостей ґрунтується на використанні логічних виразів true/false. Якщо вираз істинний, то Web застосунок поверне один вміст, а якщо вираз є хибним, то інший. Покладаючись на відмінності виведення при істинних і хибних конструкціях в запиті, стає можливим здійснювати посимвольний перебір будь- яких даних в таблиці або в файлі.

Уразливість blind SQL injection з'являється в наступних випадках:

- атакуючий не може контролювати дані, що виводяться користувачеві в результаті виконання уразливого SQL-запиту;

- коли ін'єкція потрапляє в два різних SELECT-запити, які в свою чергу здійснюють вибірку з таблиць з різною кількістю стовпців;

- коли використовується фільтрація склеювання запитів.

3. Error-based SQL Injection – трохи складніший і витратніший за часом тип атаки, що дозволяє, на основі виведених помилок СУБД, отримати інформацію про всю БД і дані, що зберігаються в ній. Експлуатується, якщо хтось у поспіху забувся відключити вивід помилок. Суть Error-based полягає в тому, що ми можемо витягувати потрібну нам інформацію з запиту за допомогою перегляду помилок роботи викликаємих функцій. Однією з таких функцій в БД MySQL є extractvalue(). Error-based SQL Injection – це найшвидша техніка експлуатації сліпих SQL ін'єкцій. Суть даної техніки полягає в тому, що різні СУБД при певних некоректних SQL-виразах можуть поміщати в повідомлення про помилку різні запитувані дані (наприклад, версію БД). Дана техніка може використовуватися в разі, коли будь-яка

помилка обробки SQL-виразів, здійснювана в СУБД, повертається назад вразливим застосунком.

4. Boolean-based SQL Injection – одна з «сліпих» ін'єкцій. Суть атаки зводиться до додавання спеціального підзапиту в вразливий параметр, на який БД буде відповідати або True, або, несподівано, False. Атака не дозволяє відразу вивести всі дані БД «на екран» зловмисникові, але дозволяє, перебираючи параметри раз по разу, отримати вміст БД, хоча для цього буде потрібно часовий відрізок співставимий з вмістом БД.

5. Time-based SQL Injection – наступна з «сліпих» ін'єкцій. В даному випадку зловмисник додає підзапит, що приводить до уповільнення або паузі роботи БД при деяких умовах. Таким чином, атакуючий, порівнюючи час відповіді на «True» і на «False» запити, символ за символом може отримати весь вміст БД, але часу піде на це більше, ніж в разі експлуатації Boolean-based атаки.

6. Out-of-band SQL Injection – рідкісний тип. Атака може бути успішна тільки при певних обставинах, наприклад, якщо сервер БД може генерувати DNS- або HTTP-запити, що зустрічається нечасто. Також, як і Blind SQL, дозволяє посимвольно збирати інформацію про дані, що там зберігаються.

Вектори SQL injection атак.

Вектори SQL injection атак описані нижче:

1. SQL маніпуляції.

У цій атаці маніпулюють виразом, наступним за словом «where», щоб створити поведінку, неочікувану програмістом БД (наприклад, складання виразу where оператором union може забезпечити доступ до даних, до яких у користувача не повинно бути доступу).

2. Впровадження коду.

У цій атаці новий оператор SQL об'єднується з раніше представленим оператором SQL (наприклад, додаючи оператор execute в кінці загального оператора). Обмеження цього виду SQLIA полягає в тому, що БД повинна підтримувати кілька операторів SQL за запит.

3. Ін'єкція виклику функції.

Це вторинна ін'єкція атаки, при якій зловмисник використовує вбудовані функції БД, щоб викликати SQLIA, яка маніпулює даними відповідно до потреб зловмисника.

4. Атака переповнення буфера.

У цьому випадку дані, введені в якості вхідних даних, сильно перевищать межі пам'яті планованого простору зберігання. Вона буде перезаписувати покажчики даних і може також використовуватися для вказівки на виконуваний файл, що змушує систему виконувати будь-який файл, який є наміром зловмисника.

1.7 Існуючі методи захисту від загроз на бази даних

Firewall'и захищають комп'ютери і мережі від спроб несанкціонованого доступу з використанням уразливих місць, що існують у сімействі протоколів TCP/IP. Додатково вони допомагають вирішувати проблеми безпеки, зв'язані з використанням уразливих систем і з наявністю великої кількості комп'ютерів у локальній мережі. Існує кілька типів *firewall'ів,* починаючи від *пакетних фільтрів,* вбудованих у прикордонні роутери, що можуть забезпечувати керування доступом для IP-пакетів, до потужних *firewall'ів,* що можуть закривати уразливості у великій кількості рівнів сімейства протоколів TCP/IP, і ще більш потужних *firewall'ів,* що можуть фільтрувати трафік на підставі усього вмісту пакета та доступ до БД.

Технологічні можливості *firewall'ів* з початку 1990-х років суттєво покращилися. Спершу були розроблені прості *пакетні фільтри,* що поступово розвивалися в більш складні *firewall'и,* здатні аналізувати інформацію на декількох мережевих рівнях. Сьогодні *firewall'и* є стандартним елементом будь-якої архітектури безпеки мережі.

Сучасні *firewall'и* можуть працювати разом з такими інструментальними засобами, як системи виявлення проникнень до БД і сканери вмісту e-mail або web з метою перебування вірусів або небезпечного

прикладного коду. Але окремо *firewall* не забезпечує повного захисту від усіх проблем, породжених Інтернетом. Як результат, *firewall'и* є тільки однією частиною архітектури інформаційної безпеки. Звичайно вони розглядаються як перша лінія оборони, однак їх краще сприймати як останню лінію оборони в організації; організація в першу чергу повинна робити безпечними свої внутрішні системи. Для внутрішніх серверів, персональних комп'ютерів і інших систем повинні вчасно виконуватися усі відновлення як самих систем, так і інших систем забезпечення безпеки, наприклад, антивірусного ПЗ.

2 МЕРЕЖЕВІ ФІЛЬТРИ ТА ВИЗНАЧЕННЯ ПОЛІТИКИ БЕЗПЕКИ

2.1 Класифікація мережевих фільтрів

Firewall'u є пристроями або системами, що керують потоком мережевого трафіку між мережами з різними вимогами до безпеки. У більшості сучасних додатків *firewall'u* і їхнього оточення обговорюються в контексті з'єднань в Інтернеті і, відповідно, використання стека протоколів TCP/IP. Однак *firewall'u* застосовуються й у мережевих оточеннях, що не вимагають обов'язкового підключення до Інтернету. Наприклад, багато корпоративних мереж підприємства ставлять *firewall'u* для обмеження з'єднань із і у внутрішні мережі, що обробляють інформацію різного рівня чутливості, таку як бухгалтерська інформація або інформація про замовників. Ставлячи *firewall'u* для контролю з'єднань з цими областями, організація може запобігти неавторизованому доступу до відповідних систем і ресурсів всередині чуттєвих областей. Тим самим, використання *firewall'a* забезпечує додатковий рівень безпеки, який інакше не може бути досягнутий.

На сьогодні існує кілька типів *firewall'ів.* Одним зі способів порівняння їхніх можливостей є перерахування рівнів моделі OSI, які даний тип *firewall'a* може аналізувати. Модель OSI є абстракцією мережевої взаємодії між комп'ютерними системами і мережевими пристроями. Розглянемо тільки рівні моделі OSI, що відносяться до *firewall'ів.* Стек протоколів моделі OSI визначається як наведено в табл. 2.1.

Рівень 1 являє собою реальну апаратуру фізичного з'єднання і середовище, таку як Ethernet.

Рівень 2 – рівень, на якому мережевий трафік передається локальною мережею (LAN). Він також є першим рівнем, що володіє можливістю адресації, за допомогою якої можна ідентифікувати окрему машину. Адреси призначаються на мережні інтерфейси і називаються MAC (Media Access Control) адресами. Ethernet-адреса, що належить Ethernet-карті, являє приклад MAC-адреси рівня 2.

Таблиця 2.1

Стек протоколів моделі OSI

Рівень 7	Application
Рівень 6	Presentation
Рівень 5	Session
Рівень 4	Transport
Рівень 3	Network
Рівень 2	Data Link
Рівень 1	Physical

Рівень 3 є рівнем, що відповідає за доставку мережевого трафіку по WAN. В Інтернеті адреси рівня 3 називаються IP-адресами; адреси звичайно є унікальними, але при визначених обставинах, наприклад, при трансляції мережевих адрес (NAT) можливі ситуації, коли різні фізичні системи мають ту саму IP-адресу рівня 3.

Рівень 4 – ідентифікує конкретний мережевий додаток і комунікаційну сесію у додаток до мережевих адрес; система може мати велику кількість сесій рівня 4 з іншими ОС. Термінологія, пов'язана із сімейством протоколів TCP/IP, включає поняття портів, що можуть розглядатися як кінцеві точки сесій: номер порту джерела визначає комунікаційну сесію на вихідній системі; номер порту призначення визначає комунікаційну сесію системи призначення.

Більш високі **рівні (5, 6 і 7)** являють собою додатки і системи кінцевого користувача.

Стек протоколів TCP/IP співвідноситься з рівнями моделі OSI як наведено в табл. 2.2.

Таблиця 2.2

Взаємозв'язок рівнів стека протоколів TCP/IP і OSI

Рівень 7	Application	Поштові клієнти, web-браузери
Рівень 4	Transport	TCP-сесії
Рівень 3	Network	IP-адресація
Рівень 2	Data Link	Ethernet-адресація

Сучасні *firewall'и* функціонують на кожному з перерахованих рівнів. Спочатку *firewall'и* аналізували менше число рівнів; тепер більш могутні з них охоплюють більше число рівнів. З погляду функціональності, *firewall*, що має можливість аналізувати більше число рівнів, є більш досконалим та ефективним. За рахунок охоплення додаткового рівня також збільшується можливість більш тонкого настроювання конфігурації *firewall'у*. Можливість аналізувати більш високі рівні дозволяє *firewall'у* надавати сервіси, що орієнтовані на користувача, наприклад, аутентифікація користувача. *Firewall*, що функціонує на рівнях 2, 3 і 4, не має справа з подібною аутентифікацією.

Незалежно від архітектури *firewall* може мати додаткові сервіси. Ці сервіси включають трансляцію мережевих адрес (NAT), підтримку протоколу динамічної конфігурації хоста (DHCP) і функції шифрування, тим самим будучи кінцевою точкою VPN-шлюзу, і фільтрацію на рівні вмісту додатка.

Багато сучасні *firewall'и* можуть функціонувати як VPN-шлюзи. Таким чином, організація може посилати незашифрований мережний трафік від системи, розташованої поза *firewall'ом*, до вилученої системи, розташованої поза корпоративним VPN-шлюзом; *firewall* зашифрує трафік і перенаправляє його на вилучений VPN-шлюз, що розшифрує його і передасть цільовій системі. Більшість найбільш популярних *firewall'ів* сьогодні сполучають ці функціональності.

Багато *firewall'ів* також включають різні технології фільтрації активного вмісту. Даний механізм відрізняється від звичайної функції *firewall'у* тим, що *firewall* тепер також має можливість фільтрувати реальні прикладні дані на

рівні 7, що проходять через нього. Наприклад, даний механізм може бути використаний для сканування на предмет наявності вірусів у файлах, приєднаних до поштового повідомлення. Він також може застосовуватися для фільтрації найбільш небезпечних технологій активного вмісту в web, таких як Java, JavaScript і ActiveX. Або він може бути використаний для фільтрації вмісту або ключових слів з метою обмеження доступу до невідповідних сайтам або доменам. Проте компонент фільтрації, убудований у *firewall*, не повинен розглядатися як єдиний можливий механізм фільтрації вмісту; можливе застосування аналогічних фільтрів при використанні стиску, шифрування або інших технологій.

2.2 Пакетні фільтри

Самий основний (базовий) спочатку був розроблений тип *firewall'a* називається *пакетним фільтром. Пакетні фільтри* в основному є частиною пристроїв роутингу, що можуть керувати доступом на рівні системних адрес і комунікаційних сесій. Функціональність керування доступом забезпечується за допомогою безліч директив: ruleset або rules (правила).

Спочатку *пакетні фільтри* функціонували на рівні 3 (Network) моделі OSI. Дана функціональність розроблена для забезпечення керування мережним доступом, ґрунтуючись на декількох блоках інформації, що міститься в мережному пакеті. В даний час усі *пакетні фільтри* також аналізують і рівень 4 (Transport).

Пакетні фільтри аналізують наступну інформацію, що міститься в заголовках пакетів 3-го і 4-го рівнів:

Адреса джерела пакету, наприклад, адреса рівня 3 системи або пристрою, відкіля отриманий вихідний мережний пакет (IP-адреса, така як 192.168.1.1).

Адреса призначення пакету, наприклад, адреса рівня 3 пакету, що він намагається досягти (наприклад, 192.168.1.2).

Тип комунікаційної сесії, тобто конкретний мережний протокол, що використовується для взаємодії між системами або пристроями джерела і призначення (наприклад, TCP, UDP або ICMP).

Можливо деякі характеристики комунікаційних сесій рівня 4, такі як порти джерела і призначення сесій (наприклад, TCP:80 для порту призначення, звичайно приналежному web-серверові, TCP:1320 для порту джерела, що належить персональному комп'ютеру, що здійснює доступ до сервера).

Іноді інформація, що відноситься до інтерфейсу роутера, на який прийшов пакет, і інформація про те, якому інтерфейсу роутера вона призначена; це використовується для роутерів із трьома і більш мережними інтерфейсами.

Іноді інформація, що характеризує напрямок, у якому пакет перетинає інтерфейс, тобто вхідний або вихідний пакет для даного інтерфейсу.

Іноді можна також вказати властивості, що відносяться до створення логів для даного пакету.

Пакетні фільтри звичайно розміщуються в мережній інфраструктурі, що використовує TCP/IP. Однак вони можуть також бути розміщені в будь-якій мережній інфраструктурі, що має адресацію рівня 3, наприклад, IPX (Novell NetWare) мережі. У сучасних мережних інфраструктурах *firewall'u* на рівні 2 можуть також використовуватися для забезпечення балансування навантаження і (або) в додатках з високими вимогами до доступності, в яких два або більш *firewall'a* використовуються для збільшення пропускної здатності або для виконання операцій відбудови.

Деякі *пакетні фільтри,* вбудовані в роутери, можуть також фільтрувати мережний трафік, ґрунтуючись на визначених характеристиках цього трафіку, для запобігання DoS і DDoS-атак.

Пакетні фільтри можуть бути реалізовані в наступних компонентах мережної інфраструктури:

- прикордонні роутери;

- ОС;

- персональні *firewall'u.*

2.3 Прикордонні роутери

Основною перевагою *пакетних фільтрів* є їхня швидкість. Тому що *пакетні фільтри* звичайно перевіряють дані до рівня 3 моделі OSI, вони можуть функціонувати дуже швидко. Також *пакетні фільтри* мають можливість блокувати DoS-атаки і пов'язані з ними атаки. З цих причин *пакетні фільтри*, вбудовані в прикордонні роутери, ідеальні для розміщення на границі з мережею з меншим ступенем довіри. *Пакетні фільтри*, вбудовані в прикордонні роутери (рис. 2.1), можуть блокувати основні атаки, фільтруючи небажані протоколи, виконуючи найпростіший контроль доступу на рівні сесій і потім передаючи трафік іншим *firewall'ам* для перевірки більш високих рівнів стека OSI.

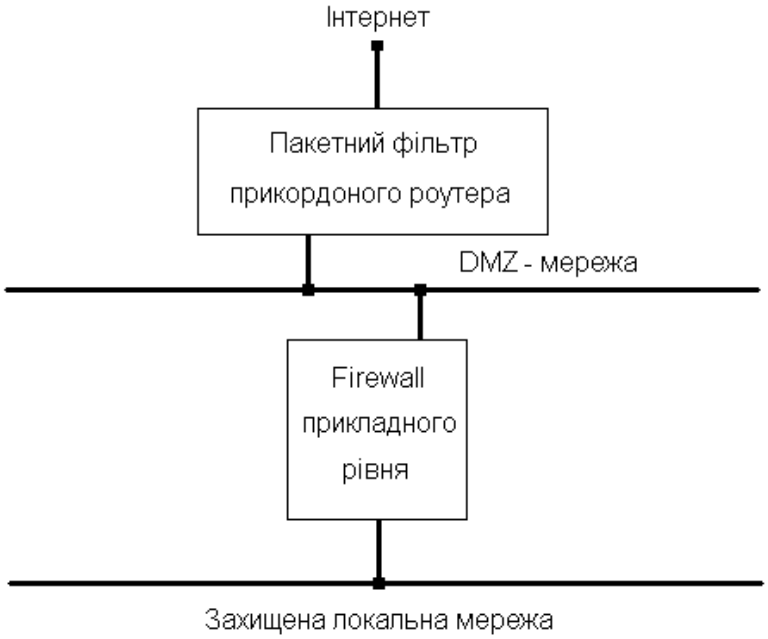

Рисунок 2.1 – Використання прикордонного роутера з можливостями пакетного фільтра

На рис. 2.1 показана топологія мережі, що використовує прикордонний роутер з можливостями *пакетного фільтра* як першу лінію оборони. Роутер приймає пакети від недовірливої мережі, що звичайно приходять від іншого

роутера або від Інтернет Сервіс Провайдера (ISP). Потім роутер виконує контроль доступу у відповідності зі своєю політикою, наприклад, блокує SNMP, дозволяє HTTP і т. п. Потім він передає пакети більш потужному *firewall'y* для подальшого керування доступом і фільтрування операцій на більш високих рівнях стека OSI. На рис. 1.1 також показана проміжна мережа між прикордонним роутером і внутрішньому *firewall'ом,* що має назву DMZ-мережа.

Переваги *пакетних фільтрів*:

Основною перевагою *пакетних фільтрів* є їхня швидкість. *Пакетний фільтр* прозорий для клієнтів і серверів, тому що не розриває TCP-з'єднання.

Недоліки *пакетних фільтрів:*

Пакетні фільтри не аналізують дані більш високих рівнів, вони не можуть запобігти атакам, що використовують уразливості або функції, специфічні для додатку. Наприклад, *пакетний фільтр* не може блокувати конкретні команди додатку; якщо *пакетний фільтр* дозволяє даний трафік для додатку, то всі функції, доступні даному додатку, будуть дозволені.

Firewall'y доступна обмежена інформація, можливості логів у *пакетних фільтрах* обмежені. Логи *пакетного фільтра* звичайно містять ту ж інформацію, що використовувалася при прийняті рішення про можливості доступу (адреса джерела, адреса призначення, тип трафіку і т. п.).

Більшість *пакетних фільтрів* не підтримують можливість аутентифікації користувача. Дана можливість забезпечується *firewall'ами,* що аналізують більш високі рівні.

Вони, звичайно, вразливі для атак, що використовують такі проблеми TCP/IP, як підробка (spoofing) мережної адреси. Багато *пакетних фільтрів* не можуть визначити, що в мережному пакеті змінена адресна інформація рівня 3 OSI. Spoofing-атаки виконуються для обходу керування доступу, здійснюваного *firewall'ом.*

При прийнятті рішень про надання доступу використовується невелика кількість інформації.

Пакетні фільтри важко зконфігурувати. Можна випадково переконфігурувати *пакетний фільтр* для дозволу типів трафіку, джерел і призначень, що повинні бути заборонені на основі політики безпеки організації.

Отже, *пакетні фільтри* більше всього підходять для високошвидкісних оточень, коли створення логів і аутентифікація користувача для мережних ресурсів не настільки важлива.

Сучасна технологія *firewall'a* включає багато можливостей і функціональностей, важко знайти *firewall,* що має можливості тільки *пакетного фільтра.* Прикладом може бути мережний роутер, що здійснює перевірку списку контролю доступу для керування мережним трафіком. Висока продуктивність *пакетних фільтрів* також сприяє тому, що вони реалізуються в пристроях, що забезпечують високу доступність і особливу надійність; деякі виробники пропонують апаратні і програмні рішення як високо доступні, так і особливо надійні. Також більшість SOHO (Small Office Home Office) пристроїв *firewall'ов* і *firewall'ов,* вбудованих в ОС, є *пакетними фільтрами.*

Припустимо, що в організації існує наступна топологія (рис. 2.2).

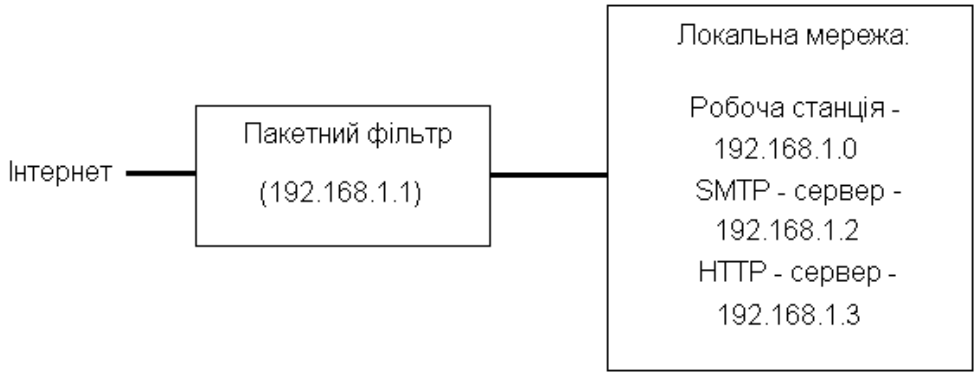

Рисунок 2.2 – Топологія мережі з використанням пакетного фільтра

Проксі-сервер, що аналізує конкретний протокол прикладного рівня, називається проксі-агентом (рис. 2.3).

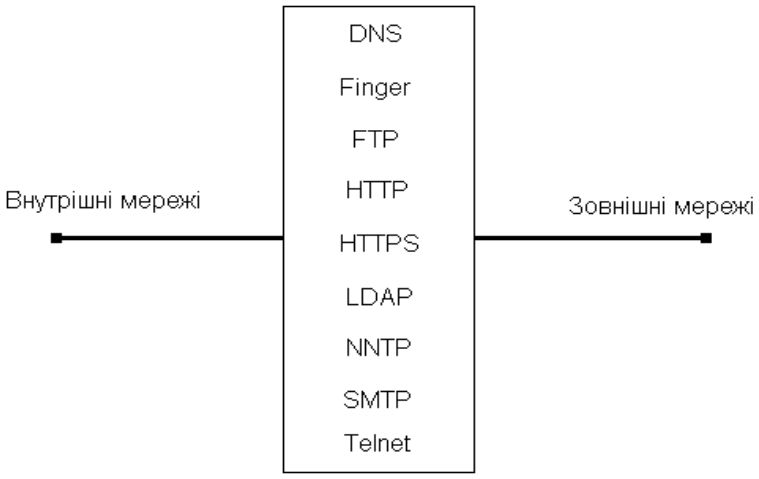

Рисунок 2.3 – Типові проксі-агенти

На рис. 2.4 показано приклад топології мережі, що має виділені *проксі-сервери* для HTTP і e-mail, розташовані позаду основного *firewall'a.* В цьому випадку e-mail проксі може бути SMTP-шлюзом організації для вхідної пошти. Основний *firewall* буде перенаправляти вхідну пошту до проксі для сканування вмісту, після чого пошта може ставати доступною внутрішнім користувачам на SMTP-сервері, наприклад, за протоколами POP3 або IMAP. HTTP-проксі повинен обробляти вихідні з'єднання до зовнішніх web-серверів і, можливо, фільтрувати активний вміст. *Проксі-сервером* може виконуватися кэшування часто використовуваних web-сторінок, тим самим зменшуючи трафік до *firewall'y.*

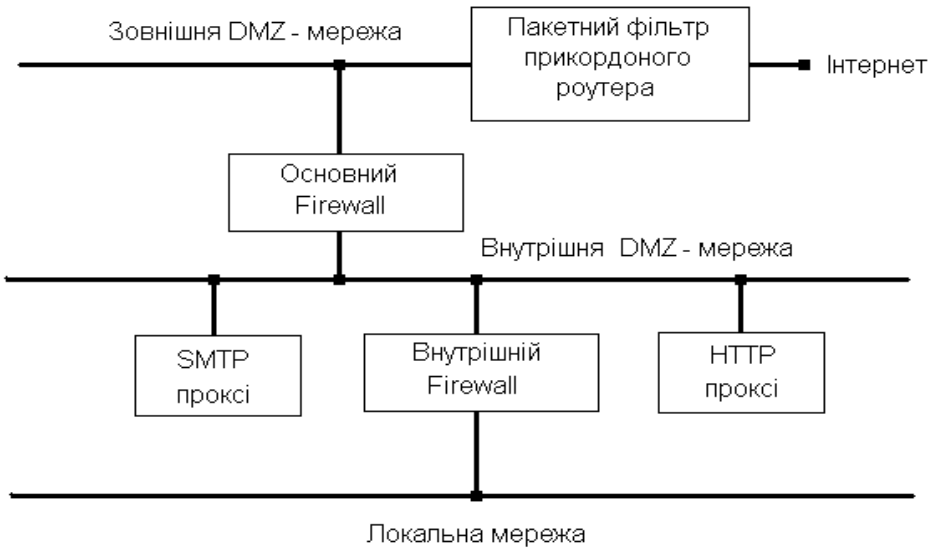

Рисунок 2.4 – Приклади виділених прикладних проксі-серверів

3 РОЗРОБКА МЕРЕЖЕВОГО ФІЛЬТРУ ДЛЯ ЗАХИСТУ ЛОКАЛЬНОГО ХОСТА

3.1 Загальне представлення локальної мережі з базою даних

У даній роботі, розглядається локальна мережа з виходом в Інтернет через виділений хост. Задача полягає в організації захисту хоста, від зовнішніх і внутрішніх атак і витоку інформації. Передбачається, що доступ до хосту будуть мати не тільки співробітники компанії, але і клієнти, а так само можливо зловмисники, що будуть зацікавлені в розкраданні інформації, зупинці WEB служби або одержання доступу до бази даних. Виникає резонне питання, можна адже розділити основні документи і дані від сервера, що б до них не було доступу з зовнішнього світу, але ми будемо виходити з тієї ситуації, що співробітники компанії бувають у відрядженні і можливо їм доведеться звернутися до даних або документів віддалено, з зовнішнього світу. Так само ще варто врахувати одну групу ризику, це самі співробітники компанії, що є потенційними факторами витоку інформації або несанкціонованого доступу до даних на сервер. Тому наша задача не тільки захистити канал інформації, а ще правильно розробити політику доступу співробітників і клієнтів до інформації на сервері.

Локальна мережа розташована в комп'ютерній компанії ComputerLand. Компанія спеціалізується на продажу комплектуючих і периферії для комп'ютерів. В офісі, де розташована компанія, нараховується п'ятнадцять комп'ютерів і один сервер з БД. На сервері зберігатися робоча документація компанії, бухгалтерські звіти і БД товарів. Сервер має виділений IP адреса. На даний IP є зареєстрованим доменне ім'я компанії.

Основні функції хоста:

1. Вихід через шлюз (proxy) в Інтернет для співробітників.
2. WEB-сервер (сайт компанії).
3. Поштовий сервер компанії.

4. Сервер БД (бухгалтерія й облік товарів), а так само для взаємозв'язку з WEB-сервером.

5. FTP-сервер БД для клієнтів і співробітників компаній з різними рівнями доступу до інформації, а так само для внутрішнього обміну документами.

3.2 Програмне та апаратне забезпечення мережі

Конфігурація захисного хоста: Intel Xeon 3.0 GHz (HT) DDR IV / 128 GB. Дана конфігурація хоста дозволяє гнучко обробляти багато задач й обробляти великі запити до бази даних і WEB-порталу компанії. Так само за рахунок організації на апаратному рівні Hyper Threading навантаження на CPU буде зменшена.

Встановлене програмне забезпечення на сервері:

- операційна система Windows Server 2003;
- WEB-сервер Apache 2.2.x + PHP 5.x.;
- база даних MySQL 5.x;
- мережевий фільтр;
- антивірусне ПЗ;
- поштовий сервер ;
- FTP-сервер;
- проксі-сервер;
- ПЗ для виявлення „троянів”.

Для детального розгляду і аналізу пропонується наступна офісна комп’ютерна мережа (рис. 3.1).

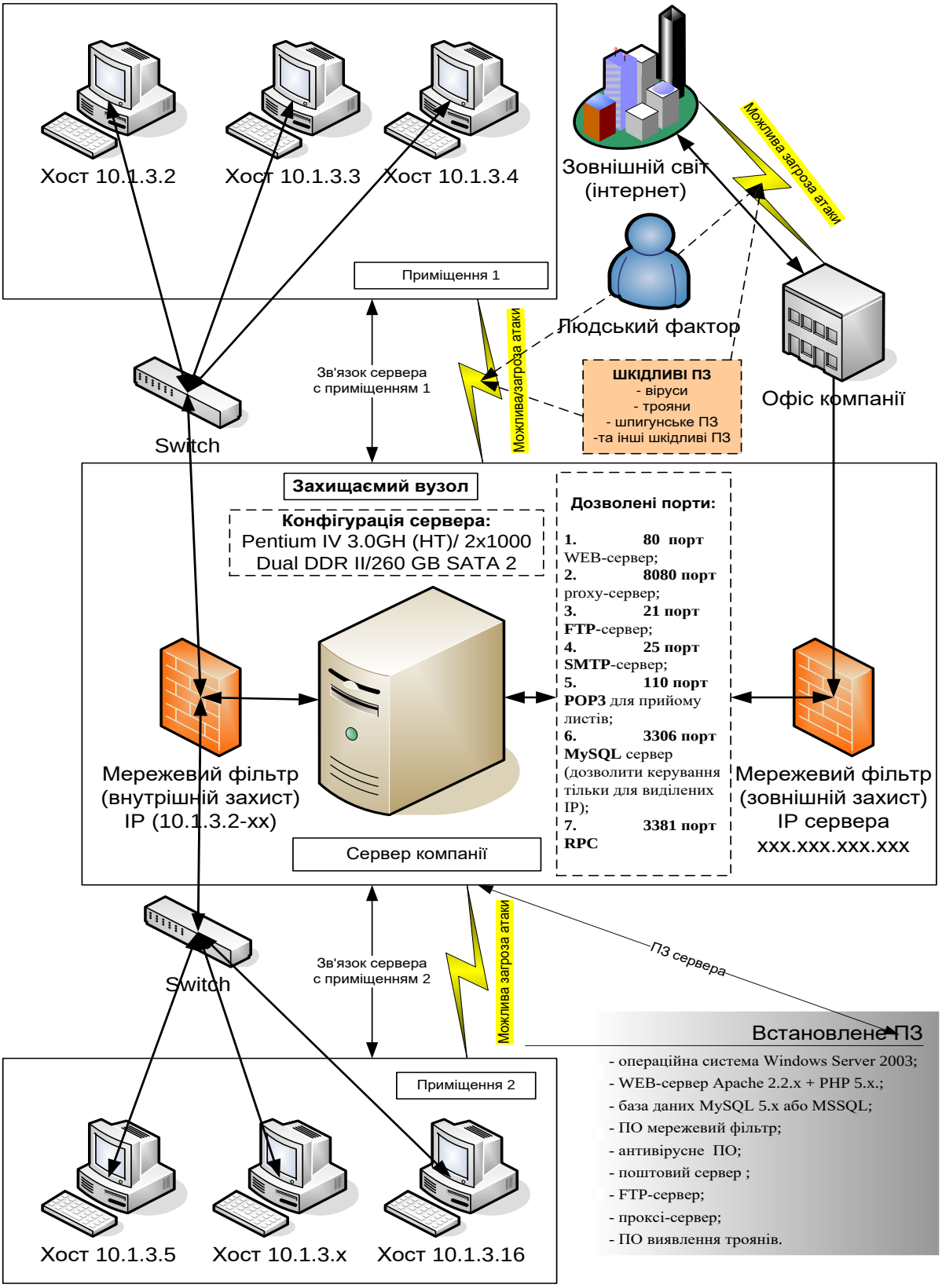

Рисунок 3.1 – Комп’ютерна офісна мережа

Обидві БД розташовані на сервері компанії і мають права розмежування.

3.3 Мережева безпека серверного хоста

Для забезпечення мережевої безпеки хоста, необхідно відкрити наступні порти:

1. **80 порт** WEB-сервера;
2. **8080 порт** proxy-сервера;
3. **21 порт FTP-**сервер;
4. **25 порт SMTP**-сервер;
5. **110 порт POP3** для прийому листів;
6. **3306 порт MySQL** сервер (дозволити керування тільки для виділених IP адрес);
7. **3381 порт RPC** вилучене адміністрування (дозволити керування тільки для виділених IP адрес).

Виходячи з мір безпеки в мережі, всі інші порти повинні бути закриті, або відкриті тільки для довірених IP адрес. Дану задачу за контролем над доступом до портів виконує в нас мережевий фільтр. Так само за допомогою мережевого фільтру ми повинні встановити контроль за додатками запущеними на сервері для обліку – з якими портами ці додатки працюють, а також який мережевий протокол вони використовують.

Так само для загальної безпеки рекомендується закрити

ECHO порт (через пінг хоста можлива організація флуд атаки, що може привести до не стабільної роботи всього хоста).

3.4. Існуючі брандмауери

Більшість організацій працюють із операційною системою Windows, перекладаючи всі можливі проблеми з безпекою під час своєї присутності в мережі на її плечі.

Однак, як показує практика, з питанням безпеки в Microsoft, незважаючи на всі її зусилля, існують чималі проблеми. Тому зараз і виходить на перший план питання охорони особистої інформації користувача сторонніми засобами.

Мережа надає прекрасні можливості для існування й розмноження різного роду шкідливих програм: хробаків, вірусів, троянських коней. Крім того, звичайно, дуже неприємна діяльність спливаючої реклами й спаму. Чого коштують, хоча б, недавні наслідки діяльності багатостраждального Blaster'a. І тільки завдяки відсутності у автора цього вірусу деструктивних помислів, величезна кількість людей крім морального збитку не одержали більше серйозних наслідків, одним з найстрашніших, звичайно, могла б бути втрата інформації.

Однак, зараження можна було б уникнути при наявності програм, що мають назву firewall (файєрвол), здатними позбавити користувача від багатьох проблем, пов'язаних з особистою безпекою. Ці програми дозволяють користувачеві самому визначати, чи дозволяти одержання або, навпаки, відправлення TCP/IP-пакетів з робочого комп'ютера.

Можна настроїти доступ до Інтернет тільки дозволеним програмам, наприклад, Web-браузеру, клієнтові електронної пошти, менеджерові закачування, ICQ або IRC. Таким чином, інші програми, які раніше могли самовільно приймати або пересилати дані на віддалений комп'ютер, не зможуть зробити свої дії. Крім того, файєрвол здатний відслідковувати доступ до машини через відкриті порти.

Користувач сам може настроїти безпечні з'єднання, забороняючи, таким чином, потенційно небезпечні. Безумовно, в епоху інформаційного розвитку, клас таких програм користується величезною популярністю. І гідних продуктів на ринку чимало.

Для розгляду я взяв декілька продуктів, що б вибрати найбільш корисніший:

- Sunbelt Kerio Personal Firewall;
- McAfee Firewall;
- Norton Personal Firewall;
- Outpost;
- Sygate Pro's firewall;

- ZoneAlarm Pro.

3.4.1 Sunbelt Kerio Personal Firewall

Sunbelt Kerio Personal Firewall – представляє собою надійну, легку у використанні персональну технологію безпеки, яка повністю захищає персональний комп'ютер від хакерських атак та витоку даних.

Огляд Інтернет з'єднань

Кожний Windows-комп'ютер, під'єднаний до Інтернету надсилає та одержує деякі дані (рис. 3.2). Sunbelt Kerio Personal Firewall представляє користувачам огляд того, які додатки відправляють дані, а які одержують.

Рисунок 3.2 – Огляд Інтернет з'єднань

Створення політики безпеки

Sunbelt Kerio Personal Firewall не пропонує попередньо встановлену універсальну політику безпеки (рис. 3.3), скоріше він постачає засобами для створення та дотримання такої політики.

Перший крок в створенні політики безпеки – це визначення того, який тип доступу до Інтернету є дозволеним. Sunbelt Kerio Personal Firewall точно знає, які додатки намагаються вийти в Інтернет. Користувачі можуть дозволити доступ до Інтернету для надійних додатків, і в той час блокувати інші.

Наприклад, додаткам для спілкування доступ може бути дозволеним, а для програм, що використовують розділені файлові ресурси – заборонений.

Продвинуті користувачі або мережеві адміністратори можуть створювати правила фільтрації пакетів, які блокують або лімітують трафік для визначених портів, протоколів або IP-адрес, надаючи рівень контролю та безпеки, який можна знайти тільки в складних мережевих файєрволах.

Рисунок 3.3 – Створення політики безпеки

Огляд трафіку та ведення логу

Огляд трафіку показує, що комп'ютер робить у визначений момент часу, відображаючи, що блокується та що дозволено. Це допомагає користувачу легко бачити чи необхідно коригувати правила. Історія трафіку може бути записана з використанням настоюваного рівня деталізації та відправлена на віддалений сервер для огляду адміністратором.

3.4.2 McAfee Firewall

Коментар: McAfee Firewall – легкий в установці й використанні firewall. Після першого запуску файєрвола, "Помічник" допомагає без особливих проблем настроїти програму.

Після того, як файєрвол був встановлений, він автоматично запускається в режимі "Стандартної безпеки" (рис. 3.4), блокуючи будь-який сумнівний трафік з Інтернету і відзначаючи все це в журналі безпеки. Щоб змінити режим, варто нажати правою клавішею миші на іконку McAfee Firewall, вибрати "Personal Firewall", а потім вибрати "Set Security Level".

Рисунок 3.4 – Зовнішній вигляд McAfee Firewall

Вікно програми складається з верхнього меню й поля інформації, де з лівої сторони традиційно доступні елементи меню, а праворуч – їхній зміст.

McAfee Firewall містить наступні пункти меню: Summary (загальна статистика, останні подія, останні новини), Internet Applications – програми, що намагаються з'єднатися з Інтернет, Inbound Events (вхідні події, потенційно –спроби атаки), Utilities (додаткові утиліти).

З підменю в "Personal Firewall" можна також вибрати наступні:

- View Summary – перегляд загальної повної статистики блокованого трафіку й атак на комп'ютер;

- View Applications – перегляд дозволених і блокованих додатків, що робили спробу доступу до Інтернет;

- View Events – перегляд історії подій.

У процесі роботи, McAfee показав відкритість 139 порту (NetBIOS) при всіх скануваннях, залишаючи, таким чином, машину уразливою до

дослідження програмами ShieldsUP!, PC Flank, SMBDie й Retina. Хоча сканування від NAT і спроби атаки від SMBDie були виявлені й блоковані.

У цілому, відкриття 139 порту в настроюваннях за замовчуванням – погана ідея. Retina також знайшла відкритий 1025 порт (потрібний для роботи мережевої гри Windows XP Blackjack, що використовується деякими троянськими кіньми).

Діяльність spyware (шпигунського програмного забезпечення) була виявлена в додатках SaveNow, Adware й Brilliant, однак, блокування їхньої діяльності було дуже слабким, або було відсутнє зовсім.

McAfee також дозволяє браузеру збирати й пересилати cookies з потенційно особистою інформацією, небажаної для розголосу.

У цілому, McAfee – легкий у використанні, і є непоганим захистом від діяльності розповсюджених експлоїтів. Це гарний вибір для початківців, яким складно розбиратися з тонкостями настроювання програми, однак занадто спрощений і неповноцінний для досвідчених користувачів.

До додаткових можливостей програми можна віднести:

- надбудовані візуальні й звукові попередження;

- безкоштовна технічна підтримка по електронній пошті або в режимі реального часу (Chat);

- автоматичне оновлення через Інтернет.

3.4.3 Norton Personal Firewall

Коментар: Norton Personal Firewall – досить простий для домашнього користувача firewall, має проте необхідні тонкі настроювання для більше досвідчених користувачів.

При першому запуску програма відкриває "Помічника", що допомагає встановити програму, проробивши наступне його настроювання, без яких-небудь проблем. Подібно McAfee, NPF (рис. 3.5) поділяє жорсткий диск на додатки, що працюють з Інтернет, але на відміну від конкурента, може автоматично надавати доступ.

Рисунок 3.5 – Зовнішній вигляд Personal Firewall

Можна також призначити пароль для зміни настроювань. На жаль, "Помічник" не зміг визначити домашню мережу, повідомляючи про незнайдені адаптери для настроювання з'єднання з мережею. Проте, незважаючи на це повідомлення, комп'ютер міг з'єднуватися з Інтернет, розділяючи для віддаленого використання також файли й принтери.

Всі інші установки виробляються вже в процесі роботи Norton Personal Firewall. На скриншоті показане основне вікно програми, де основні настроювання програми для користувача здійснюються на рівні on-off (включене/виключене):

- захист комп'ютера від вторгнень через Інтернет (Security);
- захист даних від спроб несанкціонованого доступу (Personal Firewall);
- виявлення й відбиття атак (Intrusion Detection);
- блокування банерів і рекламних блоків (Ad blocking).

Тут же за допомогою повзунка виставляється низький, середній або високий рівень захисту.

Верхнє піктограмне меню програм дозволяє вибирати між Монітором Захисту (Security Monitor), Блокуванням Трафіку (Block Traffic), Оновленням через Інтернет (Live Update), і Настроюваннями (Options). При необхідності однією кнопкою з головного вікна можна заблокувати/розблокувати доступ в Інтернет всіх програм.

NPF – єдиний файєрвол, що повністю запобігає пересиланню особистої інформації на віддалений комп'ютер. Він також виявляє й блокує будь-яку спробу атаки на машину. При скануванні кожен порт був закритий від очей хакера, cookies були блоковані. Відмінно організована робота перелогів-файлів, що надають детальну інформацію і ясні пояснення подій, що відбулися. Ця інформація надалі може допомогти в ідентифікації атакуючіх.

Зручне невелике функціональне вікно, у якому з'являється гістограма поточного трафіку, виводяться повідомлення про атаки й доступні всі основні функції програми.

На відмінність від більшості інших файєрволів, які лише повідомляють про атаку, NTF дозволяє оцінити характер і серйозність погрози й порадити виконати необхідні дії. Подібний консультативний підхід був впроваджений також у визначення роботи spyware, хоча його блокування було недостатньо гарним, а часами було відсутнє повністю.

3.4.4 OutPost

Коментар: створений як самий "просунутий" файєрвол (рис. 3.6) для Windows (має величезну кількість опцій, що набудовуються,), OutPost використовується в домашніх умовах, хоча й розроблений переважно для фахівців.

Настроювання для файєрвола передбачається через завантаження з офіційного сайту розроблювача в Інтернеті, які надалі можуть бути доповнені досвідченими користувачами через вбудовані програмні засоби. Призначення "Помічника" настроювання досить специфічне: він не дозволяє

визначати вам власні настроювання для файєрвола, даючи лише інформацію з використання продукту.

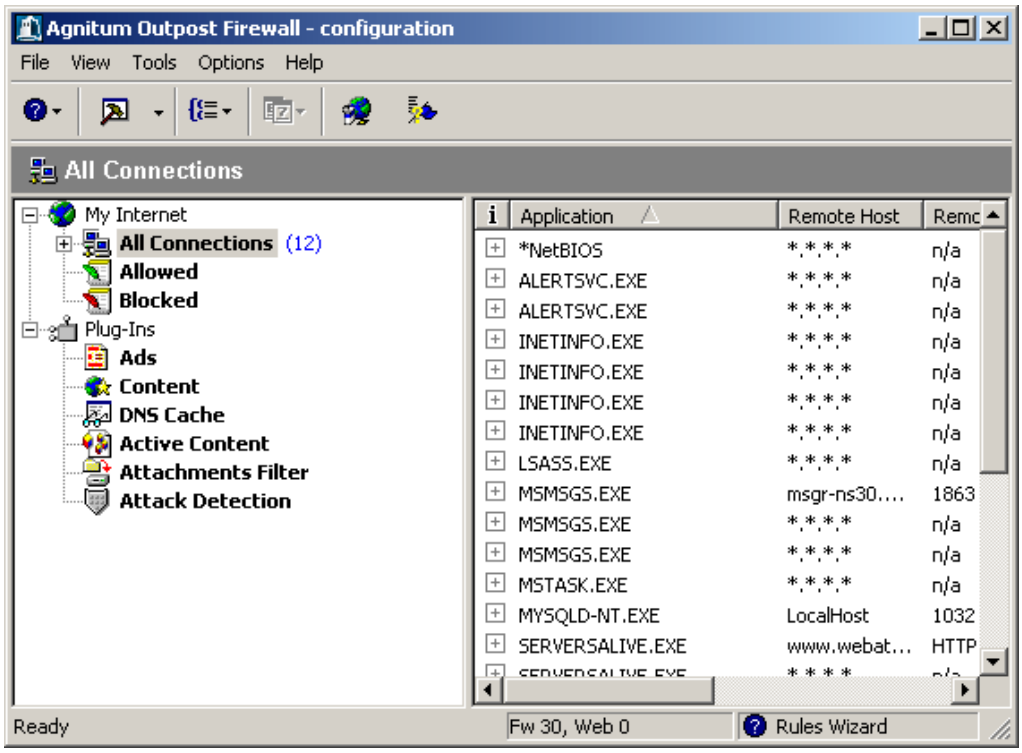

Рисунок 3.6 – Зовнішній вигляд OutPost

Вікно програми складається з верхнього меню, меню піктограм, куди винесені найбільш необхідні функції й вікна З'єднання. Ліворуч у цьому вікні перебуває меню, а праворуч – виводиться інформація з кожного пункту. У меню у вікні З'єднання 2 більші закладки: My Internet – статистика по з'єднанню з Інтернет і Plug-Ins, що підключають модулі, серед яких, варто відзначити:

- **ADS** – модуль видалення рекламних блоків при відображенні в браузері Web-сайтів. Він має доповнювати базу стандартних для реклами рядків гіпертексту й розмірів найбільш використовуваних блоків реклами (100x100, 125x125, 468x60, 470x60, 234x60, 120x80, 88x31 пікселів), по яких і відбувається фільтрація. Використання цієї функції дозволяє значно збільшити швидкість завантаження сайтів;

- **Content** – блокування доступу до Web-сайтів з певними адресами або вбудованими в html-коді певними рядками.

- **Active Content** – блокування активних елементів Web-сайтів, наприклад виконання Active, сценаріїв, написаних на Java й Visual Basic;

- **Attachments Filter** – виведення повідомлення при спробі одержання/запуску файлу, що виконується;

- **Attack Detection** – детектор і блокування атак. При використанні цієї функції можна вибрати один із трьох режимів безпеки: блокування IP-адреси атакуючого (у випадку точної ідентифікації атаки), блокування спроби сканування декількох портів або порту з певним номером, блокування спроби сканування одного порту, можливе також блокування DoS-атак.

Тестування OutPost Retin'ой показало відкритими порти 135, 1025 й 5000 (універсальний Plug and Play), залишаючи машину потенційно уразливою. ShieldsUP! також показав відкритий 5000 порт, однак PC Flank не виявив ніяких відкритих портів. Cookies також не блокуються, незважаючи на твердження розроблювачів про відсутність можливості Web-сайтів збирати й пересилати інформацію про переваги користувачів при серфінгу. У настроюваннях за замовчуванням спливаючі вікна не показуються, хоча й відзначаються в перелогах-файлах. Це не дуже зручно, тому що не кожен користувач буде заглядати в історію подій. Крім того, організація перелогів-файлів пророблена недостатньо чітко, записуючи лише факт атаки, не залишаючи навіть дати й часу нападу.

Діяльність KaZa прирівняна до spyware, тому для роботи із цим сервісом користувач повинен вибирати між блокуванням сервісу або роботою з ним без якого-небудь захисту. Розглянута версія – перший випуск даного продукту, що має безліч додаткових корисних настроювань, малопридатна для початківців, тому що робота програми з настроюваннями за замовчуванням майже безглузда.

3.4.5 Sygate Pro’s firewall

Коментар: Sygate Pro's firewall настроюється за допомогою відповідей користувача, на можливість певних додатків мати доступ до Інтернет: коли

додаток запускається – файєрвол виводить вікно діалогу для вибору доступу цього додатка: дозволити або заборонити. Користувач може створювати власний список правил (по порту, за адресою).

Передбачається, що технічно користувач непогано підкований, хоча по цьому питанню є додаткова система допомоги.

Користувачеві надаються журнали всіх дій, що відбуваються під час виконання програми:

- журнал атак, сканування портів комп'ютера й інших зазіхань на його безпеку;

- журнал з найдокладнішою інформацією по вхідному й вихідному трафіку із вказівкою програми, IP-адреси, порту, часу початку й закінчення процесу;

- журнал проходження пакетів;

- журнал запуску й закриття брандмауера.

Настроювання програми включають наступні основні функції:

- захист настроювань програми паролем для запобігання несанкціонованої зміни їх іншими користувачами комп'ютера;

- включення/відключення детектора сканування портів й інших видів найпоширеніших атак;

- завдання часу, протягом якого буде блокуватися надходження інформації з IP-адреси атакуючого;

- автоматичне відсилання інформації про атаку на певну електронну адресу;

- завдання максимального розміру журналів роботи;

- автоматичне оновлення через Інтернет.

На екрані користувачеві надається інформація в графічному виді (рис. 3.7) про вхідний/вихідний трафік, про напади.

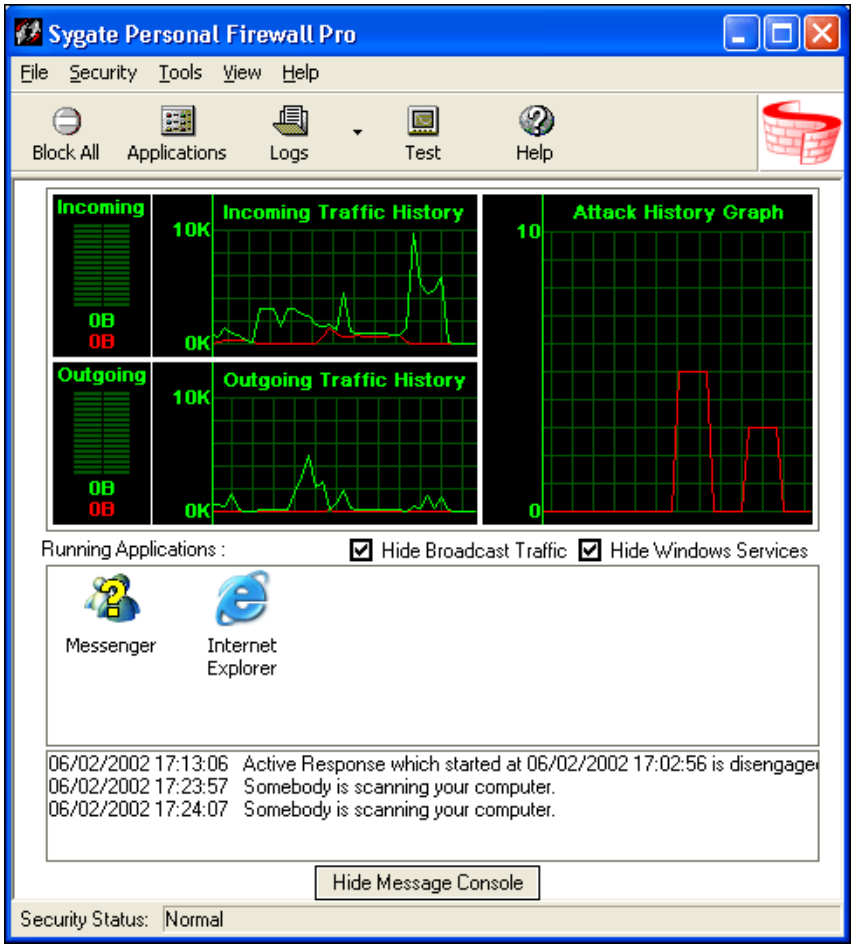

Рисунок 3.7 – Зовнішній вигляд Sygate Pro's

Цей файєрвол добре захищає машину від проникнення з боку порушника, автоматично виявляючи й забороняючи доступ атакуючого на певний, заданий проміжок часу.

Порти 5000 й 135 при скануванні здалися відкритими, однак вторгнення PC Flank й ShieldsUP! були блоковані. Інші порти були закриті й невидимі для хакерів. DoS-атаки, NAT сканування й експлоїти SMBDie були дозволені з довіреної зони, однак, повністю блоковані із зовнішньої сторони. Виявлення діяльності spyware відмінне, отриманої інформації досить для блокування цих програм.

Просунуті користувачі, безперечно, залишаться задоволені гарною організацій перелогів-файлів, роботою з електронною поштою, прихованню системної панелі й блокуванню TCP й IPX-трафіку. У цілому, цей файєрвол

дуже гарний, не вважаючи відкритих за замовчуванням уразливих портів, "закриття" яких може викликати труднощі для більшості користувачів.

3.4.6 ZoneAlarm Pro

Коментар: ZoneAlarm Pro – значно вдосконалена версія ZoneAlarm, з організацією особистого захисту машини, безліччю можливостей, що набудовують самі користувачі, захистом електронної пошти й Інтернет-гейтів (рис. 3.8).

Незважаючи на додаткові можливості, розмір дистрибутива усе ще перебуває в межах 4 Мбайт. При установці всю роботу виконує "Помічник", крім того, опції роботи з cookies задаються вже при інсталяції.

Рисунок 3.8 – Зовнішній вигляд ZoneAlarm Pro

Настроювання спрощене в порівнянні з вільно розповсюджуваною версією файєрвола (ZoneAlarm): менше спливаючих діалогів, що запитують дії користувача для якоїсь програми.

Основне (і єдине) функціональне вікно розділене на дві частини. У верхній, меншій за розміром містяться:

- графічно представлені відомості про обсяг вхідної і вихідної інформації. Ця ж гістограма є присутня як значок ZoneAlarm Pro у нижньому правому куті екрана поруч із годинниками;

- кнопка блокування доступу в Інтернет;

- піктограма, що відображає наявність підключення до Інтернету й IP-адресу;

- інформація про програми, що працюють із Інтернетом у даний момент.

Нижня, більша за розміром, частина вікна містить усілякі настроювання й інформацію, розділену по групах:

- Overview (огляд) включає відомості про число відбитих атак і кількість програм, що мають доступ в Інтернет; дані про ZoneAlarm Pro (реєстрації, версії й т.д.), короткі настроювання програми: установка пароля на користувальницькі настроювання, включення автоматичного (або ручного) оновлення ZoneAlarm через Інтернет, включення автоматичного запуску ZoneAlarm Pro після завантаження комп'ютера, режиму приховання IP-адреси й т.ін.;

- Firewall (захист від несанкціонованого доступу) включає змінювані за допомогою повзунка установки захищеності (високий рівень, коли всі підозрілі пакети й інформація блокуються, а комп'ютер при спробах його ідентифікації ззовні "небачимий"; середній рівень, коли ПК за тих самих умов ззовні "бачимий", але ресурси його блокуються; низький рівень – захист відключений). Режими встановлюються окремо для трьох зон, дві з яких визначаються користувачем. Це так званий "білий" список, що включає комп'ютери, інформація з яких не блокується й вважається безпечною, наприклад, комп'ютери тієї ж локальної мережі. У так званий "чорний" список входять комп'ютери з небезпечною інформацією. Сюди користувач звичайно заносить ті IP-адреси, з яких він раніше був атакований. Третя зона, не обумовлена користувачем, – це всі інші сервери, що не ввійшли ні в "білий", ні в "чорний" список. Крім того, тут утримується інформацію про з'єднання з Інтернетом, IP-адресу, маску під мережі;

- **Program Control** (контроль над програмами) включає вибір рівня контролю над програмами: високий припускає, що всі програми запитують доступ в Інтернет і встановлений контроль над використовуваними програмами динамічно підключеними бібліотеками (dll), при середньому всі програми запитують доступ в Інтернет, а контроль над використовуваними dll-файлами перебуває в режимі навчання (ZoneAlarm Pro запитує користувача про необхідність контролю в конкретних випадках), низький рівень має на увазі, що й контроль над програмами, і контроль над dll-файлами перебуває в режимі навчання, коли всілякий контроль відключений. Program Control також містить список програм із вказівкою наявності доступу в Інтернет і перелік всіх бібліотек, що підключають динамічно, використовуваних програмами;

- **Alerts & Logs** (сигнали й журнали) визначає рівень установки виводу сигналів при атаках і ведеться журнал з повною інформацією про них. Можливе відображення сигналів при всіх атаках, тільки при особливо небезпечних або при відсутності відображення. Аналогічно настроюється ведення журналу з описом виду атаки, позначенням її часу й дати, а також IP-адреси що атакує й порту, на який йде атака;

- **Privacy** (таємність) користувач вибирає режим роботи з інформаційними файлами. Можливе блокування всіх інформаційних файлів (що напевно спричинить проблеми з відображенням деяких Web-сайтів), блокування надходження даних з перерахованих користувачем сайтів і відсутність блокування;

- **E-mail protection** (захист електронної пошти). Спеціальна функція MailSafe перевіряє всі прийняті по електронній пошті файли на наявність вірусів й інших деструктивних об'єктів і попереджає про них користувача.

Pro-версія надає прекрасне настроювання для завдання доступу до Інтернет кожного додатка (можливість доступу по портах).

ZoneAlarm Pro добре виявив себе при рясних атаках, виявляючи й блокуючи будь-яку спробу проникнення. Всі порти були закриті, NetBIOS-

доступ, так само як і більшість cookies, були заборонені. Блокування спливаючої реклами також працювало, причому іноді надмірно агресивно, порушуючи нормальну роботу деяких скриптів браузера Internet Explorer'a. Організація перелогів-файлів – чудова. Інформація там деталізована, існує навіть можливість пошуку географічного місця розташування нападника. Spyware виявляє добре, KaZa функціонує, причому програма не здатна завантажити рекламу з http або ftp.

ZoneAlarm Pro – один із кращих файєрволов огляду, досить простий і ручний для починаючих користувачів, але проте, він має тонкі й детальні настроювання для більше досвідчених користувачів.

3.5. Програми для виявлення та знешкодження „троянських коней”

Проблема інформаційної безпеки стає усе більше актуальною з кожним днем, а шкода, заподіювана вірусами нового покоління, усе більше відчутною.

Розроблювачам антивірусного ПЗ дуже складно оперативно випускати оновлення для своїх продуктів, а системні адміністратори не завжди встигають вчасно реагувати на нові епідемії.

Основна категорія комп'ютерів, що ризикують піддатися вірусній атаці – робочі станції в офісах різних організацій, об'єднані в локальні мережі.

Використання таких потужних антивірусних моніторів, як DrWeb або Антивірус Касперського, значно знижують імовірність інфікування комп'ютера, однак їх застосовують далеко не завжди.

Причини можуть бути найрізноманітнішими: висока вартість подібного програмного забезпечення, неможливість установки програм на малопотужні комп'ютери, тому що це спричинить значне зменшення швидкості роботи системи й т.п.

Подібне відношення до питань безпеки нерідко стає причиною зараження комп'ютерів. І отоді встає питання лікування. Боротися з

наслідками шкідливої дії вірусу, що потрапив у локальну мережу, часто буває дуже складно.

Один з найефективніших способів – використання невеликих спеціалізованих утиліт, спрямованих на виявлення й видалення певного типу вірусів, а також на відновлення ушкоджених файлів. Подібні програмні рішення мають дуже багато переваг:

- інсталяція, як правило, не потрібна;

- дистрибутив настільки невеликого розміру, що він уміщається на звичайну дискету. Скачати ж його можна швидко, навіть використовуючи невисоку швидкість з'єднання:

- утиліти мають безкоштовний статус;

- перевірка здійснюється тільки на наявність самих популярних у цей момент вірусів, що істотно прискорює процес сканування;

- робота цих програм не вимагає великої кількості системних ресурсів.

Подібні утиліти можна назвати швидкою допомогою для зараженого комп'ютера або ж для такого, котрий підозрюється в зараженні. Випуском подібних утиліт займаються відомі антивірусні компанії, такі як Symantec, MacAfee й ін. Кілька утиліт були випущені в самій Microsoft.

У цьому невеликому огляді ми розглянемо самі актуальні на сьогоднішній день програми, здатні допомогти у вирішенні подібного роду проблем.

Почнемо з утиліт від Symantec. Отут ми розглянемо лише трохи із програм, випущених цією компанією. Повний список інструментів для видалення самих популярних вірусів, що випускають компанією Symantec, можна знайти в Інтернеті.

3.5.1 W32.Novarg@mm/W32.Mydoom@mm Fix Tool

Остання версія утиліти від компанії Symantec. Ця програма виявляє й видаляє віруси типу W32.Mydoom із зараженого комп'ютера. У процесі аналізу системи, програма видаляє самі файли, а також зміни в системному

реєстрі, які були зроблені після зараження. Утиліта підтримує наступні модифікації вірусу:

W32.Mydoom.A@mm, W32.Mydoom.B@mm, W32.Mydoom.F@mm, W32.Mydoom.G@mm, W32.Mydoom.H@mm, W32.Mydoom.L@mm, W32.Mydoom.M@mm, W32.Mydoom.Q@mm, Backdoor.Zincite.A, W32.Zindos.A, Backdoor.Nemog, W32.Bofra.A@mm (renamed from W32.Mydoom.AI@mm), W32.Bofra.C@mm (renamed from W32.Mydoom.AK@mm), W32.Bofra.D@mm (renamed from W32.Mydoom.AH@mm).

Вірус цього типу являє собою поштового хробака, що поширюється у вигляді атачмента з розширенням: .bat, .cmd, .exe, .pif, .scr і .zip. Коли комп'ютер інфікований, хробак відкриває на комп'ютері TCP порти від 3127 до 3198, що дозволяє потенційному недоброзичливцеві підключитися до комп'ютера й використати його як проксі для одержання доступу до мережених ресурсів.

3.5.2. Backdoor.Agent.B removal tool

Утиліта від Symantec для пошуку на комп'ютері вірусу Backdoor.Agent.B і його видалення. Цей вірус був виявлений наприкінці липня цього року. Після відвідування користувачем певних сайтів вірус установлює файл.dll, що дозволяє шкідливим додаткам робити на комп'ютері різні дії.

Як і попередня утиліта, ця програма дуже проста у використанні (рис. 3.9).

Рисунок 3.9 – Зовнішній вигляд Backdoor.Agent.B removal tool

Однак потрібно пам'ятати про те, що оскільки вірус, що вона видаляє, поширюється через інтернет і локальну мережу, перед її використанням необхідно відключити комп'ютер від всіх мереж.

3.5.3. Trojan.Vundo Removal Tool

Утиліта від Symantec для виявлення й видалення модуля Trojan.Vundo. Цей вірус був виявлений зовсім недавно.

Vundo – це компонент рекламного модуля, що закачує і відображає рекламні оголошення у вигляді pop-up вікон.

Крім закриття всіх додатків від відключення комп'ютера від Інтернету й локальної мережі, перед використанням цієї утиліти в середовищі Windows Me або XP, необхідно також відключити опцію System Restore.

3.5.4 W32.Bofra@mm FixTool

Утиліта від Symantec для виявлення й видалення одного з найпоширеніших у світі на сьогоднішній день сімейства вірусів Bofra. Bofra використає поки незакриті уразливості браузера Internet Explorer для поширення.

Bofra – це нова модифікація вірусу MyDoom. Перша інформація про виникнення цього вірусу з'явилася в перших числах листопада минулого року. Ще тоді співробітники компанії McAfee попереджали, що у вірусу є всі шанси для того, щоб стати значимим. Уже наприкінці листопада було оголошено, що по ступені поширеності Bofra посідає шосте місце у світі.

Саме через цей вірус співробітники Міністерства зв'язку Фінляндії рекомендували не використати Internet Explorer службовцям на підприємствах. Підхопити Bofra можна, просто клацнувши по посиланню у вікні браузера IE. Вірус також поширюється поштою.

W32.Bofra@mm FixTool видаляє наступні різновиди вірусу:
W32.Bofra.A@mm (renamed from W32.Mydoom.AI@mm),
W32.Bofra.B@mm (renamed from W32.Mydoom.AJ@mm),

W32.Bofra.C@mm (renamed from W32.Mydoom.AK@mm),
W32.Bofra.D@mm (renamed from W32.Mydoom.AH@mm).

3.5.5. Win32.Sobig.F@mm Removal Tool

Утиліта для видалення вірусу Win32.Sobig.F від компанії BitDefender. Даний вірус поширюється поштою. Його можна одержати в листах з темами: "Re: That movie", "Re: Wicked screensaver", "Re: Your application", "Re: Re: My details", "Re: Thank you!". Тіло зараженого листа звичайно виглядає так: "Please see the attached file for details".

Утиліта Win32.Sobig.F@mm Removal Tool (рис. 3.10) визначає всі відомі версії вірусу, видаляє інфіковані файли, видаляє процес із пам'яті, а також виправляє внесені вірусом зміни до реєстру Windows.

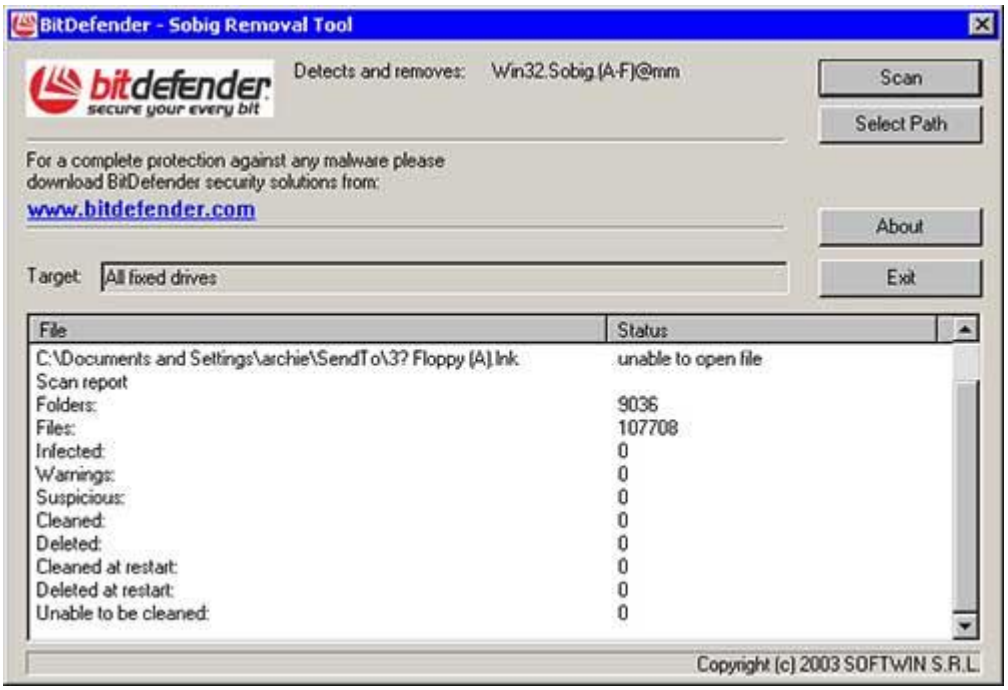

Рисунок 3.10 – Зовнішній вигляд Win32.Sobig.F@mm Removal Tool

3.5.6. McAfee AVERT Stinger

Одна із самих популярних і часто обновлюваних антивірусних утиліт, що випускає компанія McAffee. Вона допомагає відшуковувати й видаляти віруси різних типів. На даний момент у базі програми більше сорока самих

популярних у Мережі шкідливих модулів. Це всі відомі варіанти таких вірусів:

BackDoor-AQJ, BackDoor-CFB, BackDoor-CHR, BackDoor-JZ, Bat/Mumu.worm, Exploit-DcomRpc, IPCScan, IRC/Flood.ap, IRC/Flood.bi, IRC/Flood.cd, NTServiceLoader, PWS-Narod, PWS-Sincom.dll, W32/Anig.worm, W32/Bagle@MM, W32/Blaster.worm(Lovsan), W32/Bugbear@MM, W32/Deborm.worm.gen, W32/Doomjuice.worm, W32/Dumaru, W32/Elkern.cav, W32/Fizzer.gen@MM, W32/FunLove, W32/Klez, W32/Korgo.worm, W32/Lirva, W32/Lovgate, W32/Mimail, W32/MoFei.worm, W32/Mumu.b.worm, W32/MyDoom, W32/Nachi.worm, W32/Netsky, W32/Nimda, W32/Pate, W32/Polybot, W32/Sasser.worm, W32/Sdbot.worm.gen, W32/SirCam@MM, W32/Sober, W32/Sobig, W32/SQLSlammer.worm, W32/Swen@MM, W32/Yaha@MM, W32/Zafi, W32/Zindos.worm.

Програма не вимагає інсталяції й дуже проста у використанні (рис. 3.11).

Рисунок 3.11 – Зовнішній вигляд McAfee AVERT Stinger

Її особливістю є можливість настроювання деяких параметрів за допомогою кнопки Preferences. Так, можна визначити поводження утиліти у випадку виявлення вірусу на комп'ютері, настроїти параметри сканування. Також програма може відображати в процесі сканування всі файли, які були перевірені.

Наврядчи для кого-небудь секретом є той факт, що комп'ютери потрібно захищати. Особливо якщо ви використовуєте сервер на базі Windows 2000 для виходу в Інтернет. Це типово для багатьох організацій. Windows 2000 набагато простіше й зручніше в настроюванні, чим різні версії Unix. А при належному настроюванні ця система не менш стійка до злому й стабільна. Втім, багато аматорів Unix можуть із цим не погодитися, але це моя думка. Звичайно до сервера Windows 2000 підключається модем виділеної лінії (або будь-який інший пристрій), паралельно з локальною мережею. При цьому організується або роздача з'єднання з Інтернетом по локальній мережі (Internet Connection Sharing, найпростіший варіант NAT), або встановлюється проксі-сервер. Кожний із цих варіантів має свої плюси й мінуси, але з погляду гнучкості керування й обліку діяльності користувачів я предпочитаю другий. У кожному разі потрібно ясно зрозуміти – сервер прийдеться захищати. Саме дивне – Windows 2000 (так само як й XP) мають багаті вбудовані можливості з захисту.

ВИСНОВКИ

У даній роботі представлена методика виявлення та виправлення порушень цілісності схеми БД SQL на основі скриптів ініціалізації.

Було розглянуто та проаналізовано типи БД, існуючі загрози на БД за даними, code injection атаки і особливу увагу було приділено SQL injection атакам, їх наслідкам для реляційних БД та існуючим методам захисту від загроз на БД.

У ході дослідження також було проаналізовано існуючі методи виявлення та виправлення порушень БД та, основуючись на їх недоліках, запропоновано нову методику. Дана методика складається з наступних етапів: етап представлення та аналізування початкових скриптів, етап аналізування поточного стану схеми БД, етап порівняння схем, етап реагування та виправлення, результатом якого буде приведення схеми до початкової схеми.

Запропонована методика може бути використана для реалізації корпоративного продукту або продукту з відкритим початковим кодом, який зможе бути корисним як провідним компаніям в IT сфері, так і невеликим компаніям, які використовують БД в бізнесі. Розроблена система захисту інформації:

1) забезпечує захист БД на сервері;

2) відповідає принципам забезпечення безпеки інформації в системах та мережах;

3) захищає інтереси користувачів і співробітників підприємства на основі відповідного програмного забезпечення;

4) дозволяє запобігти можливим збиткам від атак на компоненти і ресурси БД.

ПЕРЕЛІК ЛІТЕРАТУРИ

1. Концепція технічного захисту інформації в Україні. Затверджено постановою Кабінету Міністрів України від 8 жовтня 1997 р. - N 1126 // Урядовий кур'єр, 1997, 12 листопада.

2. Закон України "Про Національну програму інформатизації" від 4 лютого 1998 р. - № 74/98-ВР.

3. Голубєв В.О. Програмно-технічні засоби захисту інформації від комп'ютерних злочинів. - З.: Павло, 1998. - 144 с.

4. Ларионов А.М., Майоров С.А., Новиков Г.Н. Обчислювальні комплекси, системи і мережі. - Л.: Энергоатомиздат, 1987. - 165 с.

5. Ухлинов Л.М. Міжнародні стандарти в області забезпечення безпеки даних у мережах ЕОМ. Стан і напрямки розвитку // Електрозв'язок. -1991. - № 6. - С.54-66.

6. Давыдовский А.И., Дорошкевич П.В. Захист інформації у обчислювальних мережах // Закордонна радіоелектроніка. - 1989. - № 12. - С.17-22.

7. Что такое база данных? Система управления базами данных [Електронний ресурс]. - 2019. - Режим доступу до ресурсу: https://hostiq.ua/wiki/database/.

8. Мирошниченко Е.А. К формальному определению понятия «база данных» // Проблемы. информатики. - М., 2011. - №2. - С.83-87.

9. Dinesh Thakur in his website named ECOMPUTER NOTES in chapter titled as "What are Relational Algebra and Relational Calculus" [Електронний ресурс]. - 2019. - Режим доступу до ресурсу: http://ecomputernotes.com/database-system/rdbms/relational-algebra-and-relationalcalculus.

10. Relational algebra [Електронний ресурс]. - 2019. - Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Relational_algebra.

11. Browser Architecture and Mobile Applications [Електронний ресурс]. - 2017. - Режим доступу до ресурсу: https://seng130.wordpress.com/lectures-2/web-servers.

12. Multitier architecture [Електронний ресурс]. - 2019. - Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Multitier_architecture.

13. Top 10 Database Attacks [Електронний ресурс]. - 2019. - Режим доступу до ресурсу: https://www.bcs.org/content-hub/top-ten-database-attacks.

14. SQL injection [Електронний ресурс]. - 2020. - Режим доступу до ресурсу: https://en.wikipedia.org/wiki/SQL_injection.

15. Защита от SQL инъекций [Електронний ресурс]. - 2022. - Режим доступу до ресурсу: http://www.securityscripts.ru/articles/sql-injection.html.