Міністерство освіти і науки України
Національний технічний університет
«Дніпровська політехніка»

Інститут електроенергетики
Факультет інформаційних технологій
Кафедра безпеки інформації та телекомунікацій

ПОЯСНЮВАЛЬНА ЗАПИСКА
кваліфікаційної роботи ступеню магістра

студента *Ярощука Владислава Володимировича*

академічної групи *125м-20-1*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Розробка системи виявлення мережевих атак на інформаційну систему приватного підприємства "АрдКом"*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	д.т.н, проф. Корнієнко В.І.			
розділів:				
спеціальний	д.т.н, проф. Корнієнко В.І.			
економічний	к.е.н., доц. Пілова Д.П.			
Рецензент				
Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2022

ЗАТВЕРДЖЕНО:
завідувач кафедри
безпеки інформації та телекомунікацій
_____ д.т.н., проф. Корнієнко В.І.

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістр

студенту *Ярощуук Владиславу Володимировичу* **академічної групи** *125м-20-1*
(прізвище ім'я по-батькові) (шифр)

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

на тему ***Розробка системи виявлення мережевих атак на інформаційну систему приватного підприємства "АрдКом"***

затверджену наказом ректора НТУ «Дніпровська політехніка» від р №

Розділ	Зміст	Термін виконання
Розділ 1	Проаналізувати роботу мережі ПП «АрдКом», обрати та забезпечити профіль захищеності, провести аналіз інформаційних загроз підприємства	02.11.2021
Розділ 2	Синтез комплексної системи виявлення мережевих вторгнень, що дозволяє мінімізувати ймовірності проведення вдалих атак на інформаційну систему приватного підприємства "АрдКом"	22.12.2021
Розділ 3	Розрахунок даних економічного розділу	08.01.2022

Завдання видано _____ Корнієнко В.І.
(підпис керівника) (прізвище, ініціали)

Дата видачі: 02.09.2021 р.

Дата подання до екзаменаційної комісії: 14.01.2022 р.

Прийнято до виконання _____ Ярощук В.В.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: __ с., __ рис., _ табл., __ додатка, __ джерел.

Мета роботи: синтез комплексної системи виявлення мережевих вторгнень, що дозволяє мінімізувати ймовірності проведення вдалих атак на інформаційну систему приватного підприємства "АрдКом".

У спеціальній частині проаналізовано роботу мережі ПП «АрдКом», обрано та забезпечено профіль захищеності, проведено аналіз інформаційних загроз підприємства, запропоновано синтез комплексної системи виявлення мережевих вторгнень на інформаційну систему підприємства.

В економічному розділі виконано розрахунок вартості заходів щодо зниження ймовірності проведення атак, а також розрахунок збитку від атаки на обчислювальну мережу ПП «АрдКом». Надано оцінку економічної ефективності впровадження заходів щодо зниження ймовірності проведення атак, запропонованих для ПП «АрдКом».

Новизна результатів, що очікуються, полягає у вирішенні проблеми комплексного захисту інформаційних ресурсів ІС від спектру мережевих атак, спрямованих на отримання несанкціонованого доступу до інформації та на дестабілізацію функціонування ІС в цілому.

СИСТЕМА ВИЯВЛЕННЯ ВТОРГНЕНЬ, МОДЕЛЬ ЗАГРОЗ, МЕРЕЖЕВА БЕЗПЕКА, ПРОФІЛЬ ЗАХИЩЕНОСТІ, СПУФІНГ, СНІФІНГ.

РЕФЕРАТ

Пояснительная записка:__ с., __рис., __табл., __ приложения, __источников.

Цель работы: синтез комплексной системы обнаружения сетевых вторжений, позволяющий минимизировать вероятность проведения удачных атак на информационную систему частного предприятия "АрдКом".

В специальной части проанализирована работа сети ЧП «АрдКом», избран и обеспечен профиль защищенности, проведен анализ информационных угроз предприятия, предложен синтез комплексной системы выявления сетевых вторжений на информационную систему предприятия.

В экономическом разделе выполнен расчет стоимости мероприятий по снижению вероятности проведения атак, а также расчет ущерба от атаки на вычислительную сеть ЧП АрдКом. Дана оценка экономической эффективности внедрения мероприятий по минимизации вероятности проведения атак, предложенных для ЧП «АрдКом».

Новизна ожидаемых результатов заключается в решении проблемы комплексной защиты информационных ресурсов ИС от спектра сетевых атак, направленных на получение несанкционированного доступа к информации и на дестабилизацию функционирования ИС в целом.

СИСТЕМА ОБЪЯВЛЕНИЯ ВТОРЖЕНИЙ, МОДЕЛЬ УГРОЗ, СЕТЕВАЯ БЕЗОПАСНОСТЬ, ПРОФИЛЬ ЗАЩИЩЕННОСТИ, СПУФИНГ, СНИФИНГ.

ABSTRACT

Explanatory note: ___ p., ___ pic., ___ tabl., ___ app., ___ sources.

Purpose: synthesis of a comprehensive system for detecting network intrusions, which minimizes the likelihood of successful attacks on the information system of the private enterprise "ArdCom".

In the special part the work of the network of PE "ArdCom" is analyzed, the security profile is selected and provided, the analysis of information threats of the enterprise is carried out, the synthesis of the complex system of detection of network intrusions on the information system of the enterprise is offered.

In the economic section, the calculation of the cost of measures to reduce the probability of attacks, as well as the calculation of the damage from the attack on the computer network of PE "ArdCom". An assessment of the economic efficiency of the implementation of measures to minimize the likelihood of attacks proposed for PE "ArdCom".

The novelty of the expected results is to solve the problem of comprehensive protection of IP information resources from a range of network attacks aimed at gaining unauthorized access to information and destabilizing the functioning of IP in general.

INVASION DETECTION SYSTEM, THREAT MODEL, NETWORK SECURITY, SECURITY PROFILE, SPOOFING, SNIFING.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

АС – автоматизована система;

БД – база даних;

БІТ – безпеки інформації та телекомунікацій;

ІС – інформаційна система;

ІТС – інформаційно-телекомунікаційна система;

КЗЗ – комплекс засобів захисту;

КМ – комп'ютерна мережа;

ЛОМ – локальна обчислювальна мережа;

НД – нормативний документ;

НДР – науково-дослідна робота;

НСД – несанкціонований доступ;

ОС – операційна система;

ПЗП – постійний запам'ятовуючий пристрій;

ПП – приватне підприємство;

СЗІ – система захисту інформації;

СКБД – система керування базами даних;

ТЗІ – технічний захист інформації;

ACL – Access Control List;

ARP – Address Resolution Protocol;

DOS – Disk Operating System;

IDS – Intrusion Detection Systems;

IPS – Intrusion Prevention Systems;

ICMP – Internet Control Message Protocol;

IP – Internet Protocol;

ISP – Internet Service Provider;

MAC – Media Access Control;

MMC – Microsoft Management Console;

NIDS – Network Intrusion Detection Systems;

NTFS – New Technology File System;

SYN – Synchronize;

SMTP – Simple Mail Transfer Protocol;

SSL – Secure Sockets Layer;

SSH – Secure Shell;

TCP – Transmission Control Protocol;

VMM – Virtual Memory Management;

VPN – Virtual Private Network.

ЗМІСТ

с.

ВСТУП 10
1 СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ 16
1.1 Актуальність обраної теми 16
1.2 Основні етапи реалізації атаки на мережу 16
1.2.1 Збирання інформації 18
1.2.2 Соціальна інженерія 22
1.2.3 Реалізація атаки 24
1.2.4 Завершення атаки 26
1.3 Класифікація атак 26
1.4 Виявлення вторгнень 28
1.4.1 Вузлові IDS 28
1.4.2 Мережеві IDS 30
1.5 Розгляд потенційно небезпечних атак 31
1.6 Висновок 49
РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА 50
2.1 Аналіз роботи локальної обчислювальної мережі «АрдКом» 50
2.1.1 Загальні відомості про роботу підприємства 50
2.1.2 Характеристика обчислювальної підсистеми КМ 52
2.1.3 Типові адміністративні та організаційні вимоги до КМ підприємства, що стосуються питань ТЗІ 53
2.1.4 Характеристика фізичного середовища КМ 54
2.1.5 Характеристика користувачів КМ 54
2.1.6 Характеристика інформації, що обробляється в КМ 55
2.1.7 Матриця доступу користувачів мережі до інформації з обмеженим доступом 55
2.1.8 Характеристика програмного забезпечення, що використовується в процесі роботи КМ 56
2.1.9 Характеристика захищеності КМ підприємства від атак 56

2.2 Вибір профілю захищеності 57
2.3 Аналіз інформаційних загроз підприємства 61
2.4 Вибір програмного забезпечення 63
2.4.1 Програмне забезпечення, необхідне для виявлення та усунення ARP-спуфінгу 63
2.4.2 Програмне забезпечення, необхідне для шифрування мережевого трафіку 64
2.4.3 Програмне забезпечення, необхідне для виявлення мережевих вторгнень 64
2.5 Організація заходів щодо забезпечення захищеності робочих станцій для ОС Windows 66
2.5.1 Перевірка серверів та оновлень, що використовуються на даний момент 66
2.5.2 Здобуття переліку відкритих файлів і процесів, що володіють ними 70
2.5.3 Перелік запущених служб та відкритих портів 70
2.5.4 Включення аудиту 72
2.5.5 Захист журналів подій 73
2.5.6 Зміна максимальних розмірів файлів протоколів 74
2.5.7 Відключення стандартних загальних ресурсів 76
2.5.8 Шифрування папки Temp 77
2.5.9 Очищення файлу підкачування при відключенні 78
2.5.10 Обмеження доступу користувача до застосунків 80
2.6 Організація заходів щодо забезпечення мережевої безпеки ИКС підприємства 81
2.6.1 Виявлення ARP-спуфінгу 81
2.6.2 Створення статичної ARP-таблиці 85
2.7 Організація заходів щодо підвищення захищеності мережевих каналів зв'язку ІС підприємства 87
2.8 Планування та впровадження системи виявлення вторгнень в ІС підприємства 91

2.8.1 Виявлення вторгнення за допомогою Snort 91
2.8.2 Відстеження сигналів тривоги 96
2.8.3 Запобігання і стримування вторгнень за допомогою Snort_inline 98
2.8.4 Виявлення аномальної поведінки 100
2.8.5 Керування датчиками Snort 102
2.8.6 Створення власних правил Snort 106
2.9 Висновок 114
РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ 115
3.1 Розрахунок (фіксованих) капітальних витрат 115
3.1.1 Розрахунок поточних витрат 118
3.2 Оцінка можливого збитку 120
3.2.1 Загальний ефект від впровадження системи інформаційної безпеки 124
3.3 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки 124
3.4 Висновок 125
ВИСНОВКИ 126
ПЕРЕЛІК ПОСИЛАНЬ 127
ДОДАТОК А 129
ДОДАТОК Б 130
ДОДАТОК В 131
ДОДАТОК Г 132

ВСТУП

Розвиток нових інформаційних технологій та загальна комп'ютеризація призвели до того, що інформаційна безпека не лише стає обов'язковою, вона ще й одна з важливих характеристик ІС. Існує досить обширний клас систем обробки інформації, при розробці яких чинник безпеки грає первинну роль (наприклад, банківські інформаційні системи).

Під загрозою безпеки інформації розуміються події або дії, які можуть призвести до спотворення, несанкціонованого використання або навіть до руйнування інформаційних ресурсів керованої системи, а також програмних і апаратних засобів.

Якщо виходити з класичного розгляду кібернетичної моделі будь-якої керованої системи, обурюючі дії на неї можуть носити випадковий характер. Тому серед погроз безпеці інформації слід виділяти як один з видів загрози випадкові, або неумисні. Їх джерелом може бути вихід з ладу апаратних засобів, неправильні дії працівників ІС або її користувачів, неумисні помилки в програмному забезпеченні і так далі. Такі погрози теж слід приймати до уваги, оскільки збиток від них може бути значним. Проте в приведеній дипломній роботі найбільша увага приділяється погрозам умисним, які, на відміну від випадкових, переслідують мету нанесення збитку керованій системі або користувачам. Дуже часто це робиться заради здобуття особистої вигоди.

У своїх протиправних діях, спрямованих на опанування чужих секретів, зловмисники прагнуть знайти такі джерела конфіденційної інформації, які б давали їм найбільш достовірну інформацію в максимальних об'ємах з мінімальними витратами на її здобуття.

За допомогою безлічі прийомів і засобів підбираються шляхи і підходи до таких джерел. В даному випадку під джерелом інформації мається на увазі матеріальний об'єкт, що володіє певними відомостями, що представляють конкретний інтерес для зловмисників або конкурентів.

Різноманітні публікації останніх років показують, що види атак на інформацію, що циркулює в ІС або передається по каналах зв'язку, удосконалювалися не менш інтенсивніше, ніж заходи захисту від них. Сьогодні для забезпечення захисту інформації потрібна не просто розробка приватних механізмів захисту, а реалізація системного підходу, що включає комплекс взаємозв'язаних заходів (використання спеціальних технічних і програмних засобів, організаційних заходів, нормативно-правових актів, морально-етичних заходів протидії і так далі). Комплексний характер захисту виникає з комплексних дій зловмисників, прагнучих будь-якими засобами здобути важливу для них інформацію.

Дуже важливою складовою системи захисту інформаційно-комунікаційної системи є здатність виявлення атаки зловмисника ще на першій її стадії. Дану функцію виконують системи виявлення вторгнень (IDS). Установка і грамотне налаштування такої системи значно підвищує захищеність мережі і дозволяє блокувати більшість атак, які могли б завдати підприємству великих збитків.

Мета роботи полягає в тому, щоб провести синтез заходів щодо зниження ймовірності проведення спектру атак на інформаційно-комунікаційну систему, які б відповідали вимогам конкретного підприємства, а також складались з оптимальної комбінації організаційних зауважень та програмних модулів.

Будь-яка інформаційно-комунікаційна система потребує індивідуального підходу до забезпечення якісного захисту та ефективної роботи. В будь-якій системі обробляється різна інформація з різною вартістю. Тому застосування якогось універсального комплексу захисту не є на сто відсотків ефективним.

Головні питання, які виникають при проектуванні подібних систем:

– навіщо знижувати ймовірність проведення атак;
– які конкретно елементи ІС будуть модифіковані та вдосконалені;
– які методи та засоби при цьому використовувати;

– який бюджет можна виділити на заходи щодо зниження ймовірності проведення атак.

Для відповіді на ці питання необхідно проаналізувати структуру інформаційно-комунікаційної системи, а також проаналізувати характер інформації, що в ній циркулює. Вже після цього можна буде зробити висновок, які заходи потрібно застосувати, щоб підвищити захищеність інформації в ІС від НСД та яким чином мінімізувати ймовірність проведення вдалих атак на вразливі елементи ІС.

РОЗДІЛ 1. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність обраної теми

Останнім часом зріс інтерес до питань захисту інформації. Це пов'язують з тим, що обчислювальні мережі стали більш широко використовуватися, що призводить до того, що з'являються великі можливості для несанкціонованого доступу до інформації, що передається. Небезпека насанкціонованих зловмисницьких дій в обчислювальних засобах і системах є досить реальною і з подальшим розвитком обчислювальної техніки загроза пошкодження інформації, незважаючи на всі зусилля щодо її захисту, незмінно зростає. Все це зумовлює необхідність поглибленого аналізу досвіду захисту інформації та комплексної організації методів і механізмів захисту.

Синтез комплексної системи виявлення мережевих вторгнень є актуальною в нинішній час з кількох причин:

– впровадження в ІС системи виявлення вторгнення, що є одним з головних заходів щодо мінімізування ймовірності проведення атак, дозволяє при якісному обслуговуванні значно знизити загрозу непомітної реалізації атаки ;

– впровадження запропонованого синтезу комплексної системи виявлення мережевих вторгнень є економічно вигідним методом підвищення захищеності інформації, що циркулює в ІС, у порівнянні з іншими методами;

– впровадження запропонованого синтезу комплексної системи виявлення мережевих вторгнень дозволяє не тільки знизити рівень збитків внаслідок успішної діяльності зловмисників, але й усунути більшість вразливостей ІС, що автоматично унеможливлює реалізацію деяких мережевих атак.

1.2 Основні етапи реалізації атаки на мережу

Можна виділити наступні етапи реалізації атаки:

– збирання інформації;

– реалізація атаки;

– завершення атаки.

Зазвичай, коли говорять про атаку, то мають на увазі саме третій етап, забуваючи про перший, другий і останній. Збір інформації і завершення атаки ("замітання слідів") у свою чергу також можуть бути атакою і можуть бути розділені на три етапи:

– передумови реалізації атаки;

– реалізація атаки;

– завершення атаки.

Збирання інформації – це основний етап реалізації атаки. Саме на даному етапі ефективність роботи зловмисника є запорукою "успішності" атаки. Спочатку вибирається мета атаки і збирається інформація про неї (тип і версія операційної системи, відкриті порти і запущені мережеві сервіси, встановлене системне і прикладне програмне забезпечення і його конфігурація і так далі). Потім ідентифікуються найбільш вразливі місця системи, що атакується, вплив на які призводить до потрібного зловмиснику результату.

Зловмисник намагається виявити всі канали взаємодії цілі, що атакується, з іншими вузлами. Це дозволить не лише вибрати тип атаки, що реалізовується, але і джерело її реалізації. Наприклад, вузол, що атакується, взаємодіє з двома серверами під управлінням ОС Unix і Windows. З одним сервером вузол, що атакується, має довірені відношення, а з іншим – ні.

Від того, через який сервер зловмисник буде реалізовувати напад, залежить, яка атака буде задіяна, який засіб реалізації буде вибраний і так далі. Потім, залежно від отриманої інформації і бажаного результату, вибирається атака, що дає найбільший ефект.

Традиційні засоби захисту, такі як міжмережеві екрани або механізми фільтрації в маршрутизаторах, вступають в дію лише на третьому етапі реалізації атаки, абсолютно "забуваючи" про перший і четвертий. Це призводить до того, що дуже часто здійснювану атаку дуже важко зупинити навіть за наявності потужних і дорогих засобів захисту. Приклад тому – розподілені атаки. Логічно

було б, щоб засоби захисту починали працювати ще на першому етапі, тобто запобігали б можливості збору інформації про систему, що атакується. [1]

Це дозволило б якщо і не повністю запобігти атаці, то хоч би істотно ускладнити роботу зловмисника. Традиційні засоби також не дозволяють виявити вже здійснені атаки і оцінити збиток після їх реалізації, тобто не працюють на третьому етапі реалізації атаки. Отже, неможливо визначити міри по запобіганню таким атакам надалі.

Залежно від бажаного результату порушник концентрується на тому або іншому етапі реалізації атаки. Наприклад:

- для відмови в обслуговуванні детально аналізується мережа, що атакується, в ній знаходяться лазівки і слабкі місця;

- для розкрадання інформації основна увага приділяється непомітному проникненню на вузли, що атакуються, за допомогою виявлених раніше вразливостей.

Далі будуть розглянуті основні механізми реалізації атак. Це необхідно для розуміння методів виявлення цих атак. Крім того, розуміння принципів дій зловмисників - запорука успішної оборони мережі.

1.2.1 Збирання інформації

Перший етап реалізації атак – це збір інформації про систему, що атакується, або вузол. Він включає такі дії як визначення мережевої топології, типа і версії операційної системи вузла, що атакується, а також доступних мережевих і інших сервісів і тому подібне. Ці дії реалізуються різними методами.

Вивчення оточення

На цьому етапі зловмисник досліджує мережеве оточення навколо передбачуваної цілі атаки. До таких областей, наприклад, відносяться вузли internet-провайдера "жертви" або вузли віддаленого офісу компанії, що атакується. На цьому етапі зловмисник може намагатися визначити адреси "довірених" систем і вузлів, які безпосередньо сполучені з ціллю атаки і так далі. Такі дії досить важко виявити, оскільки вони виконуються протягом достатньо

тривалого періоду часу і зовні області, що контролюється засобами захисту (міжмережевими екранами, системами виявлення атак і тому подібне).

Ідентифікація топології мережі

Існує два основні методи визначення топології мережі, що використовуються зловмисниками:

– зміна TTL (TTL – граничний період часу або число ітерацій або переходів, за який набір даних (пакет) може існувати до свого зникнення);

– запис маршруту (record route).

По першому методу працюють програми traceroute для Unix і tracert для Windows. Вони використовують поле Time to Live ("час життя") в заголовку ip-пакету, яке змінюється залежно від числа пройденних мережевим пакетом маршрутизаторів. Для запису маршруту icmp-пакету може бути використана утиліта ping.

Дуже часто мережеву топологію можна з'ясувати за допомогою протоколу SNMP (протокол управління мережами зв'язку на основі архітектури UDP; це технологія, покликана забезпечити управління і контроль за пристроями і додатками в мережі зв'язку шляхом обміну інформацією, що управляє, між агентами, розташованими на мережевих пристроях, і менеджерами, розташованими на станціях управління; SNMP визначає мережу як сукупність мережевих станцій, що управляють, і елементів мережі (головні машини, шлюзи і маршрутизатори, термінальні сервери), які спільно забезпечують адміністративні зв'язки між мережевими керівниками станціями і мережевими агентами), встановленого на багатьох мережевих пристроях, захист яких невірно конфігурований. За допомогою протоколу RIP (дистанційно-векторний протокол, який оперує транзитними ділянками в якості метрики маршрутизації) можна спробувати отримати інформацію про таблицю маршрутизації в мережі і так далі.

Майже всі з цих методів використовуються сучасними системами управління (наприклад, HP Openview, Cabletron SPECTRUM, MS Visio і так далі)

для побудови карт мережі. І ці ж методи можуть бути з успіхом застосовані зловмисниками для побудови карти мережі, що атакується.

Ідентифікація вузлів

Ідентифікація вузла, як правило, здійснюється шляхом відправки за допомогою утиліти ping команди Echo_request протоколу ICMP (мережевий протокол, що входить в стек протоколів TCP/IP. В основному ICMP використовується для передачі повідомлень про помилки і інші виняткові ситуації, що виникли в момент передачі даних, наприклад, послуга недоступна, або хост, або маршрутизатор не відповідають. Також на ICMP покладаються деякі сервісні функції). У відповідь повідомлення Echo_reply говорить про те, що вузол доступний. Існують вільно поширювані програми, які автоматизують і прискорюють процес паралельної ідентифікації великої кількості вузлів, наприклад, fping або nmap.

Небезпека даного методу в тому, що стандартними засобами вузла запити Echo_request не фіксуються. Для цього необхідно застосовувати засоби аналізу трафіку, міжмережеві екрани або системи виявлення атак. Це найпростіший метод ідентифікації вузлів. Проте він має два недоліки. Багато мережевих пристроїв і програми блокують icmp-пакеті і не пропускають їх у внутрішню мережу (або навпаки не пропускають їх назовні). Наприклад, MS Proxy Server не дозволяє проходження пакетів по протоколу ICMP. В результаті виникає неповна картина. З іншого боку, блокування icmp-пакету говорить зловмисникові про наявність "першої лінії оборони" – маршрутизаторів, міжмережевих екранів і т.д. Використання icmp-запітів дозволяє з легкістю виявити їх джерело, що, зрозуміло, не може входити в завдання зловмисника.

Існує ще один метод ідентифікації вузлів – використання "змішаного" режиму мережевої карти, який дозволяє визначити різні вузли в сегменті мережі. Але він не може застосовуватися в тих випадках, коли трафік сегменту мережі недоступний зловмиснику зі свого вузла, тобто цей метод застосовний лише в локальних мережах. Іншим способом ідентифікації вузлів мережі є так звана

розвідка DNS (комп'ютерна розподілена система для отримання інформації про домени. Найчастіше використовується для отримання IP-адреси на ім'я хоста (комп'ютера або пристрою), отримання інформації про маршрутизацію пошти та про обслуговуючі вузли для протоколів в домені), яка дозволяє ідентифікувати вузли корпоративної мережі за допомогою звернення до сервера служби імен.

Ідентифікація сервісів або сканування портів

Ідентифікація сервісів, як правило, здійснюється шляхом виявлення відкритих портів (port scanning). Такі порти дуже часто пов'язані з сервісами, заснованими на протоколах TCP (один з основних мережевих протоколів Інтернету, призначений для управління передачею даних в мережах і підмережах TCP/IP) або UDP (транспортний протокол для передачі даних в мережах IP без встановлення з'єднання). Наприклад:

– відкритий 80-й порт має на увазі наявність web-сервера;

– 25-й порт – поштового smtp-сервера;

– 31337-й – серверної частини троянського коня Backorifice;

– 12345-й або 12346-й – серверної частини троянського коня Netbus і так далі.

Для ідентифікації сервісів і сканування портів можуть бути використані різні програми, в тому числі і вільно поширювані. Наприклад, nmap або netcat.

Ідентифікація операційної системи

Основний механізм віддаленого визначення ОС – аналіз відповідей на запити, що враховують різні реалізації TCP/IP-стека в різних операційних системах. У кожній ОС по-своєму реалізований стек протоколів TCP/IP, що дозволяє за допомогою спеціальних запитів і відповідей на них визначити, яка ОС встановлена на віддаленому вузлі.[2]

Інший, менш ефективний і вкрай обмежений, спосіб ідентифікації ОС вузлів – аналіз мережевих сервісів, виявлених на попередньому етапі. Наприклад, відкритий 139-й порт дозволяє зробити висновок, що віддалений вузол, найімовірніше, працює під управлінням ОС сімейства Windows. Для

визначення ОС можуть бути використані різні програми. Наприклад, nmap або queso.

Визначення ролі вузла

Передостаннім кроком на етапі збору інформації про вузол, що атакується, є визначення його ролі, наприклад, виконанні функцій міжмережевого екрану або web-сервера. Виконується цей крок на основі вже зібраної інформації про активні сервіси, імена вузлів, топології мережі і тому подібне Наприклад, відкритий 80-й порт може вказувати на наявність web-сервера, блокування icmp-пакету вказує на потенційну наявність міжмережевого екрану, а DNS-ім'я вузла proxy.domain.ua або fw.domain.ua говорить само за себе.

Визначення вразливостей вузла

Останній крок – пошук вразливостей. На цьому кроці зловмисник за допомогою різних автоматизованих засобів або вручну визначає вразливості, які можуть бути використані для реалізації атаки. Як такі автоматизовані засоби можуть бути використані Shadowsecurityscanner, nmap, Retina і так далі.

1.2.2 Соціальна інженерія

Соціальна інженерія – це метод керування діями людини без використання технічних засобів. Метод заснований на використанні слабкостей людського чинника і вважається дуже руйнівним. Часто соціальну інженерію розглядають як незаконний метод отримання інформації, проте це не зовсім так. Соціальну інженерію можна також використати і в законних цілях, і не лише для отримання інформації, а і для здійснення дій конкретною людиною. Сьогодні соціальну інженерію часто використовують в інтернеті для отримання закритої інформації або інформації, яка представляє велику цінність. Зловмисник отримує інформацію, наприклад, шляхом збору інформації про службовців об'єкту атаки, за допомогою звичайного телефонного дзвінка або шляхом проникнення в організацію під виглядом її службовця.

Зловмисник може подзвонити працівникові компанії (під виглядом технічної служби) і вивідати пароль, зіславшись на необхідність вирішення невеликої проблеми в комп'ютерній системі. Дуже часто цей трюк проходить.

Імена службовців вдається дізнатися після черги дзвінків і вивчення імен керівників на сайті компанії і інших джерел відкритої інформації (звітів, реклами і т. п.).

Використовуючи реальні імена в розмові із службою технічної підтримки, зловмисник розповідає придуману історію, що не може потрапити на важливу нараду на сайті зі своїм обліковим записом віддаленого доступу.

Іншою підмогою в цьому методі є дослідження сміття організацій, віртуальних сміттєвих кошиків, крадіжка портативного комп'ютера або носіїв інформації.

Цей метод використовується, коли зловмисник намітив в якості жертви конкретну компанію.

Уся техніка соціальної інженерії заснована на особливостях ухвалення рішень людьми, що називаються когнітивним базисом. Вони також можуть бути названі особливістю ухвалення рішення людської і соціальної психологий, заснованою на тому, що людина повинна кому-небудь довіряти в соціальному середовищі виховання.

Претекстинг

Претекстинг – ця дія, відпрацьована заздалегідь складеним сценарієм (претекстом). В результаті ціль повинна видати певну інформацію або вчинити певну дію. Цей вид атак застосовується зазвичай по телефону. Частіше ця техніка включає більше, ніж просто брехня, і вимагає яких-небудь попередніх досліджень (наприклад, персоналізації: дата народження, сума останнього рахунку та ін.), з тим, щоб забезпечити довіру цілі. До цього ж виду відносяться атаки і по онлайн-мессенджерам.

Фішинг

Фішинг – техніка, спрямована на шахрайське отримання конфіденційної інформації. Зазвичай, зловмисник посилає e-mail, підроблений під офіційний лист від банку або платіжної системи, що вимагає "перевірки" певної інформації або здійснення певних дій. Цей лист, зазвичай, містить посилання на фальшиву вебсторінку, імітуючу офіційну, з корпоративним логотипом і контентом, і форму, що містить вимогу ввести конфіденційну інформацію, від домашньої адреси до пін-кода банківської картки.

Троянський кінь

Ця техніка експлуатує цікавість або жадібність цілі. Зловмисник відправляє e-mail, що містить у вкладенні "цікавий" скрин-сейвер, важливий апгрейд антивіруса або навіть свіжий компромат на співробітника. Така техніка залишається ефективною, поки користувачі сліпо клікати по будь-яких вкладеннях.

Зворотня соціальна інженерія

Метою зворотньої соціальної інженерії (reverse social engineering) є бажання змусити ціль самостійно звернутися до зловмисника по "допомогу". З цією метою зловмисник може застосувати наступну техніку:

– Диверсія. Створення оборотної неполадки на комп'ютері жертви.

– Реклама. Зловмисник підкидає жертві оголошення виду "Якщо виникли неполадки з комп'ютером, подзвоніть по такому-то номеру".[3]

1.2.3 Реалізація атаки

З цієї миті починається спроба отримання доступу до вузла, що атакується. При цьому доступ може бути як безпосередній, тобто проникнення на вузол, так і опосередкований, наприклад, при реалізації атаки типа "відмова в обслуговуванні". Реалізація атак в разі безпосереднього доступу також може бути розділена на два етапи:

– проникнення;

– встановлення контролю.

Проникнення

Проникнення має на увазі під собою подолання засобів захисту периметра (наприклад, міжмережевого екрану). Реалізовуватися це може різними шляхами. Наприклад, використання вразливості сервісу комп'ютера, що "дивиться" назовні або шляхом передачі ворожого вмісту по електронній пошті або через аплети Java. Такий вміст може використовувати так звані "тунелі" в міжмережевому екрані (не плутати з тунелями VPN), через які потім і проникає зловмисник. До цього ж етапу можна віднести підбір пароля адміністратора або іншого користувача за допомогою спеціалізованої утиліти (наприклад, L0phtcrack або Crack).

Встановлення контролю

Після проникнення зловмисник встановлює контроль над вузлом, що атакується. Це може бути здійснено шляхом впровадження програми типа "троянський кінь" (наприклад, Netbus або Backorifice). Після встановлення контролю над потрібним вузлом і "замітання" слідів, зловмисник може здійснювати всі необхідні несанкціоновані дії дистанційно без відома власника атакованого комп'ютера. При цьому встановлення контролю над вузлом корпоративної мережі повинне зберігатися і після перезавантаження операційної системи. Це може бути реалізовано шляхом заміни одного із завантажувальних файлів або вставки посилання на ворожий код у файли автозавантаження або системний реєстр. Відомий випадок, коли зловмисник зміг перепрограмувати EEPROM (електрично видаляємий з можливістю перепрограмування ПЗП, один з видів енергонезалежної пам'яті) мережевої карти і навіть після переустановлення ОС він зміг повторно реалізувати несанкціоновані дії. Простішою модифікацією цього прикладу є впровадження необхідного коду або фрагмента в сценарій мережевого завантаження (наприклад, для ОС Novell Netware).

Цілі реалізації атаки

Необхідно зазначити, що зловмисник на другому етапі може переслідувати дві мети. По-перше, отримання несанкціонованого доступу до самого вузла і інформації, що міститься на ньому. По-друге, отримання

несанкціонованого доступу до вузла для здійснення подальших атак на інші вузли. Перша мета, як правило, здійснюється лише після реалізації другої.

Тобто, спочатку зловмисник створює собі базу для подальших атак і лише після цього проникає на інші вузли. Це необхідно для того, щоб приховати або істотно ускладнити знаходження джерела атаки.

1.2.4 Завершення атаки

Етапом завершення атаки є "замітання слідів" з боку зловмисника. Зазвичай це реалізується шляхом видалення відповідних записів з журналів реєстрації вузла і інших дій, що повертають атаковану систему у вихідний, "передатакований" стан.[4]

1.3 Класифікація атак

Існують різні типи класифікації атак. Наприклад, ділення на пасивні і активні, зовнішні і внутрішні, умисні і ненавмисні. Проте щоб не заплутатися у великій різноманітності класифікацій, які мало застосовуються на практиці, далі буде запропонована більш "життєва" класифікація:

1) віддалене проникнення (remote penetration). Атаки, які дозволяють реалізувати віддалене управління комп'ютером через мережу. Наприклад, Netbus або Backorifice.

2) локальне проникнення (local penetration). Атака, яка призводить до отримання несанкціонованого доступу до вузла, на якому вона запущена. Наприклад, Getadmin.

3) віддалена відмова в обслуговуванні (remote denial of service). Атаки, які дозволяють порушити функціонування або перенавантажувати комп'ютер через Internet. Наприклад Teardrop або trin00.

4) локальна відмова в обслуговуванні (local denial of service). Атаки, які дозволяють порушити функціонування або перенавантажити комп'ютер, на якому вони реалізуються. Прикладом такої атаки є "ворожий" аплет, який

завантажує центральний процесор безперервним циклом, що призводить до неможливості обробки запитів іншого програмного забезпечення.

5) мережеві сканери (network scanners). Програми, які аналізують топологію мережі і виявляють сервіси, доступні для атаки. Наприклад, система nmap.

6) сканери вразливостей (vulnerability scanners). Програми, які шукають вразливості на вузлах мережі і які можуть бути використані для реалізації атак. Наприклад, система SATAN або Shadowsecurityscanner.

7) зломщики паролів (password crackers). Програми, які "підбирають" паролі користувачів. Наприклад, L0phtcrack для Windows або Crack для Unix.

8) аналізатори протоколів (sniffers). Програми, які "прослуховують" мережевий трафік. За допомогою цих програм можна автоматично шукати таку інформацію, як ідентифікатори і паролі користувачів, інформацію про кредитні карти і так далі. Наприклад, Microsoft Network Monitor, Netxray компанії Network Associates або Lanexplorer.

Компанія Internet Security Systems, Inc. ще більше скоротила число можливих категорій, довівши їх до 5:

– збір інформації (Information gathering);
– спроби несанкціонованого доступу (Unauthorized access attempts);
– відмова в обслуговуванні (Denial of service);
– підозріла активність (Suspicious activity);
– системні атаки (System attack).

Перші 4 категорії відносяться до віддалених атак, а остання – до локальних, що реалізовується на вузлі, що атакується. Можна відмітити, що в дану класифікацію не попав цілий клас так званих "пасивних" атак ("прослуховування" трафіку, "помилковий dns-сервер", "підміна arp-сервера" і тому подібне).

Класифікація атак, реалізована в багатьох системах виявлення атак, не може бути категоричною. Наприклад, атака, реалізація якої для ОС Unix

(наприклад, переповнювання буфера statd) може мати найтяжкіші наслідки (найвищий пріоритет), для ОС Windows може бути взагалі не застосовна або мати дуже низьку міру ризику. Крім того, існує неясність і в самих назвах атак і вразливостей. Одна і та ж атака, може мати різні найменування у різних виробників систем.[5]

1.4 Виявлення вторгнень

Виявлення вторгнень – це активний процес, при якому відбувається виявлення зловмисника при його спробах проникнути в систему. Виявлення вторгнень допомагає при превентивній ідентифікації активних загроз за допомогою сповіщень і запобігань про те, що зловмисник здійснює збір інформації, необхідної для проведення атаки. Система виявлення вторгнень IDS призначена для розмежування авторизованого входу і несанкціонованого проникнення. Систему IDS можна порівняти з охоронцем, що стежить за всім, що відбувається, і що виявляє несанкціоновані дії.

Типи систем виявлення вторгнень IDS:

1) вузлові системи виявлення вторгнень (HIDS). Розташовується на окремому вузлі і відстежує ознаки атак на даний вузол.

2) мережеві системи виявлення вторгнень (NIDS). Знаходиться на окремій системі, що відстежує мережевий трафік на наявність ознак атак, що проводяться в підконтрольному сегменті мережі.

1.4.1 Вузлові IDS

Вузлові IDS (HIDS), є системою датчиків, що завантажуються на різних серверах організації і керуються центральним диспетчером. Датчики відстежують різні типи подій і роблять певні дії на сервері або передають повідомлення. Датчики HIDS відстежують події, пов'язані з сервером, на якому вони завантажені. Сенсор HIDS дозволяє визначити, чи була атака успішною, якщо атака мала місце на тій же платформі, на якій встановлений датчик.

Процес датчика на сервері може займати від 5 до 15 % загального процесорного часу. Тому доведеться знаходити продуктивнішу систему, щоб присутність датчика негативно не позначилася на продуктивності використовуваної системи.

Існує п'ять основних типів датчиків HIDS:

– аналізатори журналів;

– датчики ознак;

– аналізатори системних викликів;

– аналізатори поведінки додатків;

– контролери цілісності файлів.

Аналізатори журналів

Аналізатор журналу – це те, що відображає сама назва датчика. Процес виконується на сервері і відстежує відповідні файли журналів в системі. При відповідності запису в журналі і критерію в процесі датчика HIDS, робиться встановлена дія. Адміністратор системи, за бажанням, може визначити інші записи журналу, що представляють певний інтерес. Аналізатори журналів не запобігають атаці на систему, а реагують на подію вже після того, як вона сталася. Його можна використовувати для відстежування активності і переміщення запису про активність персоналу в область, недосяжну для адміністратора або користувача.

Датчики ознак

Це набори певних ознак подій безпеки, що зіставляються з вхідним трафіком або записами журналу. Можливість аналізу вхідного трафіку є відмінністю даних датчиків від аналізаторів журналів. Датчик ознак HIDS є корисним при відстежуванні авторизованих користувачів усередині інформаційних систем.

Аналізатори системних викликів

Дані аналізатори здійснюють аналіз викликів між додатками і операційною системою для ідентифікації подій, пов'язаних з безпекою. Датчики HIDS даного

типа розміщують програмну спайку між операційною системою і додатками. При виконанні додатком дій, його виклик операційної системи аналізується і зіставляється з базою даних ознак, які є прикладами різних типів поведінки, що являються атакуючими діями, або об'єктом інтересу для адміністратора IDS. Аналізатори системних викликів відрізняються від вище перерахованих датчиків тим, що вони можуть запобігати діям. Забезпечення неправильної конфігурації датчиків цього типа або їх некоректне налаштування спричиняє за собою помилки в додатках або відмови в їх роботі.

Аналізатори поведінки додатків

Застосовуються у вигляді програмної спайки між додатками і операційною системою, і перевіряє виклик на предмет того, чи дозволено додатку виконувати дану дію, замість визначення відповідності виклику ознакам атак. При конфігурації таких датчиків необхідно створювати список дій, дозволених для виконання кожною програмою. Постачальники датчиків даного типа надають шаблони для найширшого використовуваних застосувань.

Контролери цілісності файлів

Відстежують зміни у файлах за допомогою використання криптографічної контрольної суми або цифрового підпису файлу (шифрування). При зміні хоч би малої частини вихідного файлу (це можуть бути атрибути файлу, такі як час і дата створення), кінцевий цифровий підпис файлу буде змінений. Мета даного алгоритму – максимальне зниження можливості для внесення змін до файлу із збереженням колишнього підпису. Обробці даного алгоритму для створення початкового підпису піддається при початковій конфігурації датчика кожен файл. Отримане число є доповненням до підпису і при необхідності зіставляється з оригіналом. Невідповідність показує, що до файлу були внесені зміни. Контролер цілісності файлів не здійснює ідентифікацію атаки, а деталізує результати проведеної атаки. [6]

1.4.2 Мережеві IDS

NIDS – це програмний процес, що працює на спеціально виділеній системі, що відповідає за перемикання мережевої карти в системі в нерозбірливий режим роботи, при якому мережевий адаптер пропускає весь мережевий трафік в програмне забезпечення NIDS. Аналізує трафік, використовуючи набір правил і ознак атак для визначення того, чи представляє цей трафік який-небудь інтерес, після чого генерується відповідна подія. На даний момент в більшість систем NIDS вбудований набір ознак атак, з якими зіставляється трафік в каналі зв'язку. За відсутності якихось ознак атаки в системі виявлення вторгнень, система NIDS не помічає цю атаку. Дані системи дозволяють вказувати трафік, яким зацікавилися, за адресою джерела, кінцевою адресою, портом джерела або кінцевим портом. Це дає можливість відстежування трафіку, що не відповідає ознакам атак.

Переваги використання NIDS:

– NIDS можна повністю приховати в мережі таким чином, що зловмисник не знатиме про те, що за ним ведеться спостереження;

– одна система NIDS може використовуватися для моніторингу трафіку з великим числом потенційних систем-цілей;

– NIDS може здійснювати перехоплення вмісту всіх пакетів, що прямують на систему-ціль.

Недоліки використання NIDS:

– система NIDS може лише видавати сигнал тривоги, якщо трафік відповідає передвстановленим правилам або ознакам;

– NIDS може упустити потрібний трафік через використання широкої смуги пропускання або альтернативних маршрутів;

– система NIDS не може визначити, чи була атака успішною;

– у комутованих мережах (на відміну від мереж із загальними носіями) потрібні спеціальні конфігурації, без яких NIDS перевірятиме не весь трафік.

1.5 Розгляд потенційно небезпечних атак

Сніфінг

Аналізатор трафіку, або сніфер (sniffer) – це деякий прослуховуючий пристрій, впроваджений в мережу для перехоплення даних, що передаються.

Сніфер – це програма, яка перехоплює мережеві пакети. Сніфер, встановлений на проміжному комп'ютері, через який проходитимуть пакети, – здатний захоплювати їх, поки вони ще не досягли цілі. У різних сніферів процес захоплення інформації реалізований по різному, наприклад (комп'ютер користувача) - (сусідній комп'ютер) - (комп'ютер зі сніфером) - (віддалений комп'ютер). Стандартний пакет мандрує з "комп'ютера користувача" через мережу. Він пройде через кожен комп'ютер в мережі, починаючи з "сусіднього комп'ютера", через "комп'ютер з сниффером" і закінчуючи "віддаленим комп'ютером". Кожна машина повинна ігнорувати пакет, якщо він не призначений для її IP адреси. Проте, машина зі сніфером дозволяє приймати будь-який пакет, який через неї проходить. Сніфінг – це сукупність заходів по перехопленню трафіку. У рамках конкретного продукту може бути реалізована функція по захопленню пакетів (packet capturing). На цьому етапі можна отримати деякий сирий (mashine readable) дамп (відрізок) даних, зазвичай розділений на частини по межах кадрів (пакетів).

Перехоплення трафіку може здійснюватися:

– звичайним "прослуховуванням" мережевого інтерфейсу (метод ефективний при використанні в сегменті концентраторів замість комутаторів, інакше метод малоефективний, оскільки на сніфер потрапляють лише окремі фрейми);

– підключенням сніфера в розрив каналу;

– відгалуженням (програмним або апаратним) трафіку і направленням його копії на сніфер;

– через аналіз побічних електромагнітних випромінювань і відновлення трафіку, що таким чином прослуховується;

– через атаку на канальному (MAC-spoofing) або мережевому рівні (IP-spoofing), що призводить до перенаправлення трафіку жертви або усього трафіку сегменту на сніфер з подальшим поверненням трафіку в належну адресу.

Сніфери застосовуються як у благих, так і в деструктивних цілях. Аналіз трафіку, що пройшов через сниффер, дозволяє:

– виявити паразитний, вірусний і закольцований трафік, наявність якого збільшує завантаження мережевого устаткування і каналів зв'язку (сніфери тут малоефективні; як правило, для цих цілей використовують збір різноманітної статистики серверами і активним мережевим устаткуванням і її подальший аналіз);

– виявити в мережі шкідливе і несанкціоноване ПО, наприклад, мережеві сканери, флудери, троянські програми, клієнти піринговых мереж й інші (це зазвичай роблять за допомогою спеціалізованих сніферів - моніторів мережевої активності);

– перехопити будь-який незашифрований (а іноді і зашифрований) призначений для користувача трафік з метою отримання паролів й іншої інформації;

– локалізувати несправність мережі або помилку конфігурації мережевих агентів (для цієї мети сніферы часто застосовуються системними адміністраторами).

Оскільки в "класичному" сніфері аналіз трафіку відбувається в ручному режимі, із застосуванням лише простих засобів автоматизації (аналіз протоколів, відновлення TCP-потоку), то він підходить для аналізу лише невеликих об'ємів трафіку.

По "місцезнаходженню" сніфер може працювати:

1) на маршрутизаторі (шлюзі) – при такому розкладі можливо перехоплювати трафік, що проходить через інтерфейси цього шлюзу. Наприклад, з однієї локальної мережі в іншу мережу і у зворотний бік. Якщо встановити

сніфер на маршрутизаторі провайдера Інтернет, то можна відстежувати трафік його користувачів.

2) на кінцевому вузлі мережі – стосовно Ethernet маються на увазі два основних можливих варіанти прослуховування. Класичний, не комутований Ethernet припускає, що кожен мережевий інтерфейс в принципі "чує" увесь трафік свого сегменту. Проте в нормальному режимі роботи мережевої карти, прочитавши перші 48 біт заголовка фрейма, станція порівнює свою MAC-адресу з адресою одержувача, вказаного у фреймі. Якщо адреса чужа, станція перестає читати чужий фрейм. Таким чином, в нормальному режимі можна перехоплювати і аналізувати тільки свій трафік. Для перехоплення пакетів усіх станцій сегменту треба перевести мережеву карту в режим під назвою promiscuous mode, щоб вона продовжувала читати не призначені їй пакети. Практично усі реалізації сніферів дозволяють перехід карти в promiscuous mode. Використання комутованого Ethernet створює ситуацію, коли навіть перехід карти в promiscuous mode робить прослуховування не призначеного даній станції трафіку практично неможливим. Проте існує технологія організації такого прослуховування шляхом так званого ARP-спуфінгу. Суть в наступному: комутатор створює так званий "broadcast domain", і хост зі встановленим сніфером за допомогою підробки ARP-повідомлень може прикинутися, наприклад, пограничним маршрутизатором (розсилаючи постійно ARP-повідомлення, де мережевій адресі маршрутизатора відповідає MAC-адреса прослуховуючої станції). Таким чином, трафік сусідів примусово завернеться у бік зловмисника. В іншому ж сніфери можуть відрізнятися один від одного головним чином функціональними можливостями, як то:

– фізичні інтерфейси і протоколи канального рівня, що підтримуютсья;

– якість декодування і кількість відомих протоколів;

– призначений для користувача інтерфейс і зручність відображення;

– додаткові можливості: статистика, перегляд в реальному часі, генерування або модифікація пакетів і інше.

На рисунку 1.1 приведена схема реалізації сніфінгу шляхом перехоплення інформації, що циркулює між комп'ютером користувача та сервером, комп'ютером зі встановленим аналізатором трафіку.

Рисунок 1.1 - Схема проведення сніфінгу

ARP-спуфінг

Протокол ARP призначений для перетворення IP-адрес в MAC-адреси. Найчастіше йдеться про перетворення в адреси Ethernet, але ARP використовується і в мережах інших технологій: Token Ring, FDDI та інших.

Протокол може використовуватися в наступних випадках:

– хост A хоче передати IP-пакет вузлу B, що знаходиться з ним в одній мережі;

– хост A хоче передати IP-пакет вузлу B, що знаходиться з ним в різних мережах, і користується для цього послугами маршрутизатора R.

У будь-якому з цих випадків вузлом A використовуватиметься протокол ARP, тільки в першому випадку для визначення MAC-адреси вузла B, а в другому – для визначення MAC-адреси маршрутизатора R. У останньому випадку пакет буде переданий маршрутизатору для подальшої ретрансляції.

Далі для простоти розглядається перший випадок, коли інформацією обмінюються вузли, що знаходяться безпосередньо в одній мережі. (Випадок коли пакет адресований вузлу що знаходиться за маршрутизатором відрізняється

тільки тим, що в пакетах передаваних після того, як ARP-перетворення завершене, використовується IP-адреса одержувача, але MAC-адреса маршрутизатора, а не одержувача.)

Протокол ARP є абсолютно незахищеним. Він не має ніяких способів перевірки достовірності пакетів: як запитів, так і відповідей. Ситуація стає ще складнішою, коли може використовуватися свавільний ARP (gratuitous ARP).

Свавільний ARP – така поведінка ARP, коли ARP-відгук присилається, коли в цьому (з точки зору одержувача) немає особливої необхідності. Свавільний ARP-відгук це пакет-відповідь ARP, прислана без запиту. Він застосовується для визначення конфліктів IP-адрес в мережі: як тільки станція отримує адресу по DHCP або адреса привласнюється вручну, розсилається ARP-відгук gratuitous ARP.

Свавільний ARP може бути корисний в наступних випадках:

– оновлення ARP-таблиць, зокрема, в кластерних системах;

– інформування комутаторів;

– сповіщення про включення мережевого інтерфейсу.

Незважаючи на ефективність свавільного ARP, він є особливо небезпечним, оскільки з його допомогою можна запевнити віддалений вузол в тому, що MAC-адреса якої-небудь системи, що знаходиться з нею в одній мережі, змінилася і вказати, яка адреса використовується тепер.

До виконання ARP-спуфінга в ARP-таблиці вузлів A і B існують записи з IP та MAC-адресами один одного. Обмін інформацією здійснюється безпосередньо між вузлами A і B (зелена стрілка).

В ході виконання ARP-спуфінгу комп'ютер C, що виконує атаку, відправляє ARP-відгуки (без отримання запитів):

– вузлу A: з IP-адресою вузла B і MAC-адресою вузла C;

– вузлу B: з IP-адресою вузла A і MAC-адресою вузла C.

Через те що комп'ютери підтримують свавільний ARP (gratuitous ARP), вони модифікують власні ARP-таблиці і вносять туди записи, де замість

справжніх MAC-адрес комп'ютерів A і B стоїть MAC-адреса комп'ютера C (червоні стрілки).

Після того, як атака виконана, коли комп'ютер A хоче передати пакет комп'ютеру B, він знаходить в ARP-таблиці запис (він відповідає комп'ютеру C) і визначає з нього MAC-адресу одержувача. Відправлений по цій MAC-адресі пакет приходить комп'ютеру C замість одержувача. Комп'ютер C потім ретранслює пакет тому, кому він дійсно адресований – тобто комп'ютеру B (сині стрілки).

На рисунку 1.2 приведена схема реалізації ARP-спуфінгу комп'ютером C з метою перехоплення інформації, що циркулює між комп'ютерами A і B.

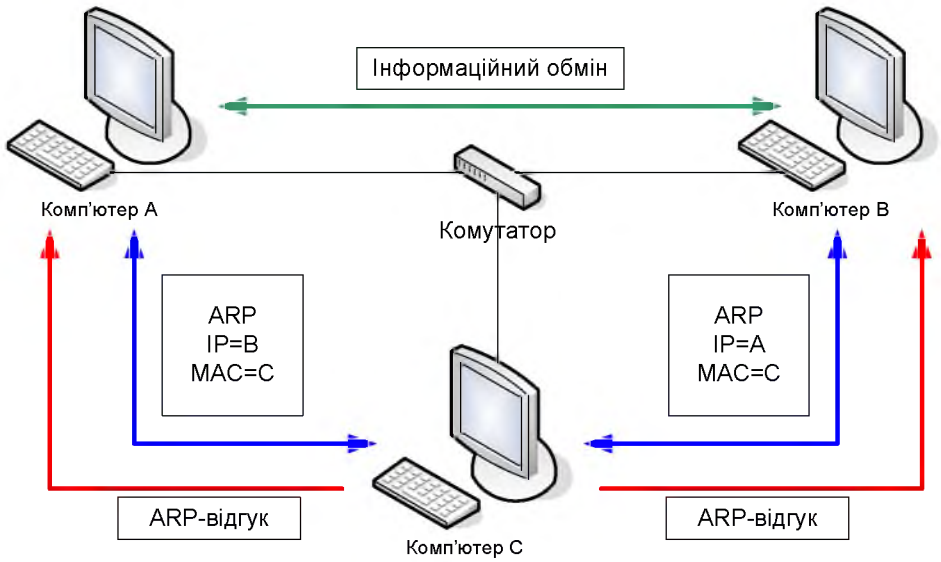

Рисунок 1.2 – Схема проведення ARP-спуфінгу

Віруси й троянські програми

Комп'ютерний вірус – різновид комп'ютерних програм, відмінною особливістю яких є здатність до розмноження (самореплікація). На додаток до цього віруси можуть без відома користувача виконувати інші довільні дії, що у тому числі завдають шкоди користувачеві і/або комп'ютеру. З цієї причини віруси відносять до шкідливих програм. Неспеціалісти помилково відносять до комп'ютерних вірусів й інші види шкідливих програм – програми-шпигуни і

навіть спам. Відомі десятки тисяч комп'ютерних вірусів, які поширюються через інтернет у всьому світі.

Віруси поширюються, копіюючи своє тіло і забезпечуючи його подальше виконання: впроваджуючи себе у виконуваний код інших програм, замінюючи собою інші програми, прописуючись в автозапуск й інше. Вірусом або його носієм можуть бути не лише програми, що містять машинний код, але і будь-яка інформація, що містить автоматично виконувані команди – наприклад, пакетні файли і документи Microsoft Word і Excel, що містять макроси. Крім того, для проникнення на комп'ютер вірус може використати вразливості в популярному програмному забезпеченні (наприклад, Adobe Flash, Internet Explorer, Outlook), для чого розповсюджувачі впроваджують його в звичайні дані (картинки, тексти, і т. д.) разом з експлоітом, що використовує вразливість.

Канали розповсюдження комп'ютерних вірусів:

Флеш-носії

Нині USB-накопичувачі замінюють дискети і повторюють їх долю – велика кількість вірусів поширюється через знімні накопичувачі, включаючи цифрові фотоапарати, цифрові відеокамери, MP3-програвачі, стільникові телефони. Використання цього каналу раніше було переважно обумовлене можливістю створення на накопичувачі спеціального файлу autorun.inf, в якому можна вказати програму, що запускається Провідником Windows при відкритті такого накопичувача. У версії MS Windows під торговою назвою Windows 7 можливість автозапуску файлів з переносних носіїв була усунена. Флеш-накопичувач – основне джерело зараження для комп'ютерів, не підключених до Інтернету.

Електронна пошта

Зазвичай віруси в листах електронної пошти маскуються під нешкідливі вкладення: картинки, документи, музику, посилання на сайти. У деяких листах можуть міститися дійсно тільки посилання, тобто в самих листах може і не бути шкідливого коду, але якщо відкрити таке посилання, то можна потрапити на спеціально створений вебсайт, що містить вірусний код. Багато поштових

вірусів, потрапивши на комп'ютер користувача, потім використовують адресну книгу зі встановлених поштових клієнтів типу Outlook для розсилання самого себе далі.

Системи обміну миттєвими повідомленнями

Також поширена розсилка посилань на нібито фото, музику або програми, що насправді є вірусами, через програми миттєвого обміну повідомленнями.

Вебсторінки

Можливо також зараження через сторінки Інтернету зважаючи на наявність на сторінках всесвітньої павутини різного "активного" вмісту : скриптів, ActiveX-компонентів. В цьому випадку використовуються вразливості програмного забезпечення, встановленого на комп'ютері користувача, або вразливості в ПЗ власника сайту (що небезпечніше, оскільки зараженню піддаються доброчесні сайти з великим потоком відвідувачів), а нічого не підозрюючі користувачі, зайшовши на такий сайт, ризикують заразити свій комп'ютер.

Інтернет і локальні мережі («хробаки»)

«Хробаки» – вид вірусів, які проникають на комп'ютер-жертву без участі користувача. Черв'яки використовують так звані "діри" (вразливості) в програмному забезпеченні операційних систем, щоб проникнути на комп'ютер. Вразливості – це помилки і недоробки в програмному забезпеченні, які дозволяють віддалено завантажити і виконати машинний код, внаслідок чого вірус-хробак потрапляє в операційну систему і, як правило, починає дії із зараження інших комп'ютерів через локальну мережу або Інтернет. Зловмисники використовують заражені комп'ютери користувачів для розсилки спаму або для DDoS-атак.

Троянська програма (також – троянець, троянський кінь) – шкідлива програма, поширювана людьми на відміну від вірусів і хробаків, які поширюють мимоволі.

Троянські програми – найпростіший вид шкідливих програм, складність яких залежить виключно від складності істинного завдання і засобів маскування. Найпримітивніші екземпляри (наприклад, що стирають вміст диска після запуску) можуть мати початковий код в декілька рядків.

Як і будь-яка шкідлива програма, троян може робити практично усе, що завгодно, наприклад:

– заважати роботі користувача;

– красти або знищувати дані і реєстраційну інформацію (паролі, кредитні карти);

– вимагати гроші (за можливість роботи або збереження даних);

– шпигувати за користувачем;

– використовувати ресурси комп'ютера (у тому числі мережеві з'єднання), у тому числі для протизаконної діяльності і так далі і тому подібне.

Троянські програми поширюються людьми – як безпосередньо завантажуються в комп'ютерні системи зловмисниками-інсайдерами, так і спонукають користувачів завантажувати і/або запускати їх на своїх системах.

Для досягнення останнього троянські програми поміщаються зловмисниками на відкриті або індексовані ресурси (файл-сервери і системи файлообміну), носії інформації, надсилаються за допомогою служб обміну повідомленнями (наприклад, електронною поштою), потрапляють на комп'ютер через проломи безпеки або завантажуються самим користувачем з адрес, отриманих одним з перерахованих способів.

Іноді використання троянів є лише частиною спланованої багатоступінчастої атаки на певні комп'ютери, мережі або ресурси (у тому числі, треті).

Троянська програма може імітувати ім'я і іконку існуючої, неіснуючої, або просто привабливої програми, компонента, або файлу даних (наприклад картинки), як для запуску користувачем, так і для маскування в системі своєї присутності.

Троянська програма може в тій чи іншій мірі імітувати або навіть повноцінно виконувати завдання, під яке вона маскується (у останньому випадку шкідливий код вбудовується зловмисником в існуючу програму).

Вцілому, троянські програми виявляються і видаляються антивірусним і антишпигунським ПЗ так само як й інші шкідливі програми.

Троянські програми гірше виявляються контекстними методами антивірусів (заснованих на пошуку відомих програм), тому що їх поширення краще контролюється, і екземпляри програм потрапляють до фахівців антивірусної індустрії з більшою затримкою, ніж мимоволі поширювані шкідливі програми. Проте евристичні (пошук алгоритмів) і проактивні (стеження) методи для них такі ж ефективні.

На рисунку 1.3 наведена схема роботи вірусів й троянських програм, а точніше – методи проникнення шкідливого коду на комп'ютер жертви та наслідки, які можуть виникнути в результаті діяльності вірусів й троянських програм.

Рисунок 1.3 – Схема роботи вірусів й троянських програм

Dos-атаки

DoS-атака (від англ. Denial of Service, відмова в обслуговуванні) – атака на обчислювальну систему з метою довести її до відмови, тобто створення таких умов, при яких легітимні (правомірні) користувачі системи не можуть отримати

доступ до ресурсів (серверів), що надаються системою, або цей доступ ускладнений.

Відмова "ворожої" системи може бути як самоціллю (наприклад, зробити недоступним популярний сайт), так і одним з кроків до оволодіння системою (якщо в позаштатній ситуації ПЗ видає яку-небудь критичну інформацію – наприклад, версію, частину програмного коду тощо).

Якщо атака виконується одночасно з великого числа комп'ютерів, говорять про DDoS-атаку (від англ. Distributed Denial of Service, розподілена атака типу "відмова в обслуговуванні"). В деяких випадках до DDoS-атаки призводить легітимна дія, наприклад, розміщення на популярному інтернет-ресурсі посилання на сайт, розміщений на не дуже продуктивному сервері (слешдот-ефект). Великий наплив користувачів призводить до перевищення допустимого навантаження на сервер і відмови в обслуговуванні частини з них.

Існують різні види виконання DoS-атак:

- помилка в програмному коді, що призводить до звернення до невживаного фрагмента адресного простору, виконання неприпустимої інструкції або іншої необроблюваної виняткової ситуації, коли відбувається аварійне завершення серверного застосування;

– недостатня перевірка даних користувача, що призводить до нескінченного або тривалого циклу або підвищеного тривалого споживання процесорних ресурсів (вичерпання процесорних ресурсів) або виділення великого об'єму оперативній пам'яті (вичерпання пам'яті);

– флуд (англ. flood) – атака, пов'язана з великою кількістю зазвичай безглуздих або сформованих в неправильному форматі запитів до комп'ютерної системи або мережевого устаткування, що має на меті або що привела до відмови в роботі системи через вичерпання ресурсів ресурсів системи – процесора, пам'яті або каналів зв'язку;

– атака другого роду – атака, яка прагне викликати неправдиве спрацьовування системи захисту і таким чином привести до недоступності ресурсу.

Якщо атака (зазвичай флуд) робиться одночасно з великої кількості IP-адрес, то в цьому випадку вона називається розподіленою атакою на відмову в обслуговуванні (DDoS).

Флудом називають величезний потік безглуздих запитів з різних комп'ютерів з метою зайняти "ворожу" систему (процесор, ОЗП або канал зв'язку) роботою і цим тимчасово вивести її з ладу. Поняття "DDoS-атака" практично рівносильно поняттю "флуд", і в ужитку те і інше часто взаємозамінні.

Будь-який комп'ютер, що має зв'язок із зовнішнім світом по протоколу TCP/IP, схильний до таких типів флуда:

1) SYN-флуд: при цьому виді атаки на вузол, що атакується, спрямовується велика кількість SYN-пакетов по протоколу TCP (запитів на відкриття з'єднання). При цьому на сервері, що атакується, через короткий час вичерпується кількість відкритих сокетів і сервер перестає відповідати.

2) UDP-флуд: цей тип флуда атакує не комп'ютер-ціль, а його канал зв'язку. Провайдери припускають, що UDP пріоритетніший, ніж TCP. Великою кількістю UDP-пакетів різного розміру викликається перевантаження каналу зв'язку, і сервер, працюючий по протоколу TCP, перестає відповідати.

3) ICMP-флуд: те ж саме, але за допомогою ICMP-пакетов.

Багато служб влаштовані так, що невеликим запитом можна викликати велику витрату обчислювальних потужностей на сервері. У такому разі атакується не канал зв'язку або TCP-подсистема, а безпосередньо служба – флудом подібних "хворих" запитів. Наприклад, вебсервери схильні до HTTP-флуду: для виведення сервера з ладу може застосовуватися як просте GET /, так і складний запит у базу даних на кшталт GET /index.php?search=<випадковий рядок>.

Існує думка, що спеціальних засобів для виявлення DoS-атак не потрібно, оскільки факт DoS-атаки неможливо не помітити. У багатьох випадках це дійсно так. Проте досить часто відзначалися успішні атаки, які були помічені жертвами лише через 2-3 доби. Бувало, що негативні наслідки атаки (типу флуд) полягали в зайвих витратах по оплаті трафіку, що з'ясовувалося лише при отриманні рахунку. Крім того, багато методів виявлення атак неефективні поблизу ціді атаки, але ефективні на магістральній мережі. У такому разі доцільно ставити системи виявлення саме там, а не чекати, поки користувач, що піддався атаці, сам її помітить і звернеться по допомогу. До того ж, для ефективної протидії необхідно знати тип, характер і інші показники DoS-атаки, а оперативно отримати ці відомості якраз і дозволяють системи виявлення. [3]

Методи виявлення можна розділити на декілька великих груп:

– сигнатурні: засновані на якісному аналізі трафіку;

– статистичні: засновані на кількісному аналізі трафіку;

– гібридні: поєднані переваги двох попередніх методів.

На рисунку 1.4 приведена схема проведення DDoS-атаки, що розглянута у двох випадках: атака на канал зв’язку та атака безпосередньо на сам комп’ютер-жертву.

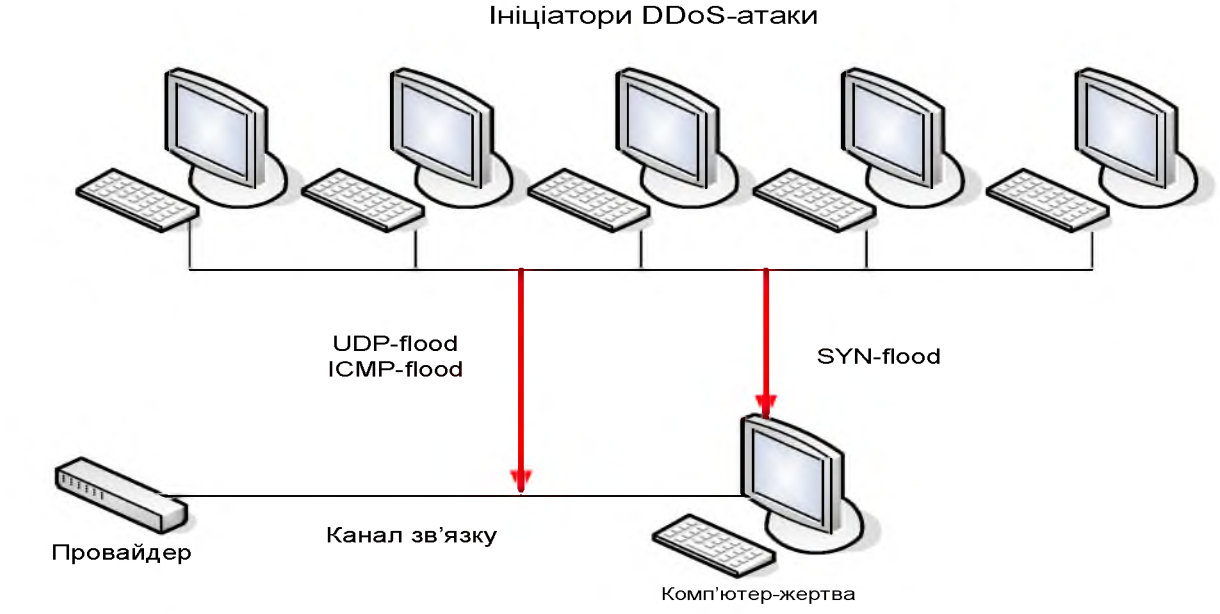

Рисунок 1.4 - Схема проведення DDoS-атаки

Nuke-атаки

Атака типу Nuke є найпоширенішою атакою в мережі з причини її простоти проведення. Атака робиться з використанням спеціальних програм, створених для цих цілей, і називаються вони нюкерами. Словом Nuke називають сьогодні будь-яку атаку на порти віддалених комп'ютерів, що призводить до "краху" операційної системи або до розриву інтернет-з'єднання. Nuke-атаки не порушують стан файлової системи і окрім невеликих незручностей не доставляють ніякої шкоди комп'ютеру. Перезавантаживши, користувач може продовжити роботу в Мережі. Способів атаки дуже багато, тому розглянуті будуть лише декілька. Ідея nuke полягає в тому, що він посилає неправильні (invalid) ICMP пакети, і машина-хост втрачає зв'язок із з'єднанням інтернету.

1) атака під назвою Ping o' Death (багато розповсюджених операційних систем виявилися до неї дуже чутливі, у багатьох системах діру вже дуже давно прикрили, але все-таки досі трапляються успішні проведення цієї атаки). Ping o' Death заснована на відсиланні більше 65 527 байтів даних + 20 байт заголовка IP + 8 байт ICMP-заголовка в одному IP-пакеті. Оскільки більшість систем не збирають усі пакети до того, як отримають їх повністю, то отримані дані просто не уміщатимуться в 16-бітовій внутрішній змінній і виникає неусувна помилка.

2) атака Out Of Band (WinNuke або OOB). Заснована на тому, що 139 порт Windows не закритий для стороннього вторгнення, тому, відіславши на цей порт пакет даних з прапором заголовка OOB, можна спровокувати зависання операційної системи і появу синього екрану. З'ясувалося експериментально, якщо під'єднатися до Windows машини по будь-якому слухаючому порту і послати туди декілька байт OutOfBand даних, реалізація стека TCP/IP не знає, що робити з цими даними і попросту підвішує або перезавантажує машину.

Існує два типи нюків: простий Nuke і Winnuke, і вони трохи відрізняються за типом дії, але ефект у них один. Простий, або класичний, Nuke побудований на стандартах протоколу TCP/IP. Сенс його в тому, що зловмисник, використовуючи службовий протокол TCP/IP ICMP (Internet Control Message

Protokol), відправляє запит на перевірку певної адреси в мережі і підсовує відповідь типу "помилка доступу", "адреса недоступна", "мережа недоступна" і так далі і тому подібне. Відповідно, сервер починає переналагоджування таблиць маршрутизації і обриває з'єднання з "недоступною" адресою. WinNuke - чистий DoS (Denial of Service, не плутати з MS-DOS). Є такий термін в стандарті TCP/IP - OOB (Out Of Band), і означає він пересилку термінових службових даних. Дуже багато машин під Windows використовують у своїй роботі протокол NetBIOS. Цьому NetBIOS закидається декілька байт даних по OOB, і цей NetBIOS починає гальмувати і комп'ютер зависає. При цьому на екрані з'являється текст помилки на синьому фоні (blue screen of death).[6]

На рисунку 1.5 приведена схема проведення Nuke-атаки зловмисником на комп'ютер-жертву шляхом маніпуляцій з NetBios-протоколом та ICMP-протоколом.

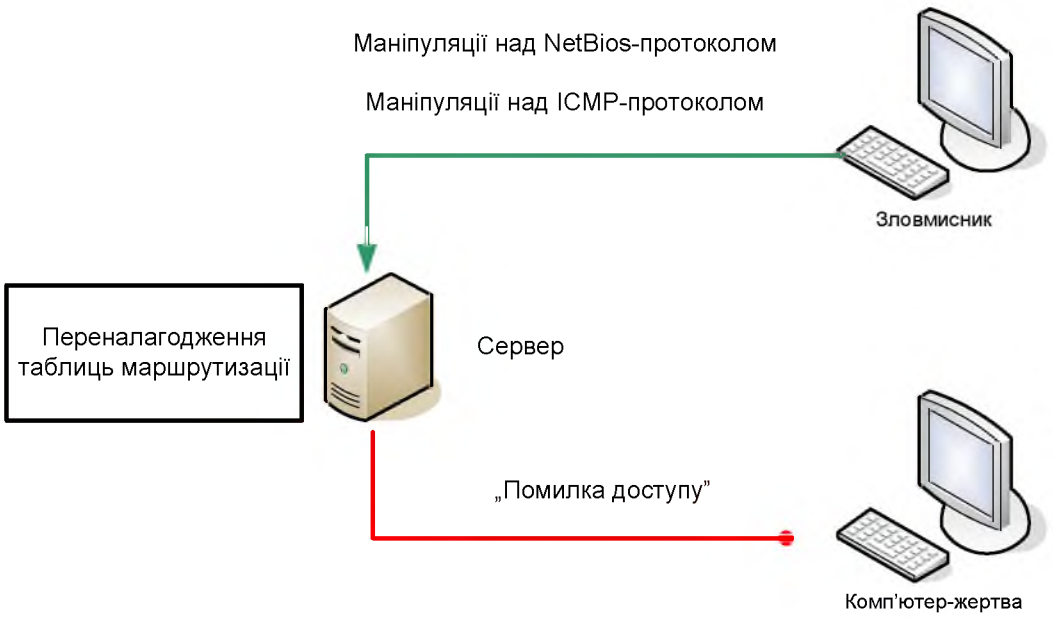

Рисунок 1.5 – Схема проведення Nuke-атаки

NetBios-атаки

NetBios-атаки – спроба отримати доступ до файлів, використовуючи мережевий протокол NetBios (протокол для роботи в локальних мережах на

персональних ЕОМ типу IBM/PC) і ресурси, зроблені загальнодоступними без пароля. Один з найпоширеніших типів атак, оскільки найпростіше реалізується.

Перша стадія – збір даних.

1) посилається запит ping. Якщо немає відповіді, то ця адреса пропускається (машина не атакується на другій стадії).

2) визначається ім'я DNS (про всяк випадок, воно ніде не використовується).

3) виконується спроба встановити null session (анонімну сесію -анонімне з'єднання (з порожнім ім'ям і паролем). Те ж, що і net use\\1.2.3.4\

\ipc$/user:"").

4) виконується спроба отримати перелік shared resources (ресурсів загального користування).

5) якщо null session встановити не вдалося, то перша стадія для даної адреси завершується.

6) виконується спроба отримати перелік імен користувачів.

7) виконується спроба отримати інформацію про операційну систему, а саме:

– платформу;

– версію LAN Manager;

– роль системи в мережі: прапори Master Browser, Backup Browser, PDC (центральний (головний) комп'ютер локальної мережі (сервер), на якому працюють служби каталогів і розташовується сховище цих каталогів), BDC (служба підключення до бізнес-даних) і т.д.

Друга стадія – підбір паролів. Всі імена користувачів, отримані на першій стадії, об'єднуються в один список, що дозволяє врахувати доменну архітектуру систем на базі Windows. На цій стадії перевіряються з'єднання типа net use \\1.2.3.4\ipc$ password /user:username. Якщо з'єднання успішне, то перевіряються всі можливість підключення до всіх shared resources, і, коли підключення успішне, можливість запису.

1) перевіряється підключення з випадковим ім'ям і порожнім паролем. Якщо підключення успішне, то вважати, що дозволене підключення Guest, і подальші перевірки паролів будуть безглузді. Перевіряються shared resources і більше на цю машину паролі не підбираються.

2) перевіряється підключення зі всіма іменами, зібраними на першій стадії. Перевіряється порожній пароль і пароль, співпадаючий з ім'ям користувача. Якщо підключення успішне, то перевіряються shared resources.

Вище було приведено загальний опис реалізації NetBios-атаки. Неправильне адміністрування та налагодження робочих станцій може призвести до крадіжки або знищення інформації, тому про цей тип атак не слід забувати.

На рисунку 1.6 приведена схема проведення NetBios-атаки зловмисником на комп'ютер-жертву шляхом маніпуляцій з NetBios-протоколом з метою отримання доступу до ресурсів загального використання, а звідти – і до самого комп'ютеру.

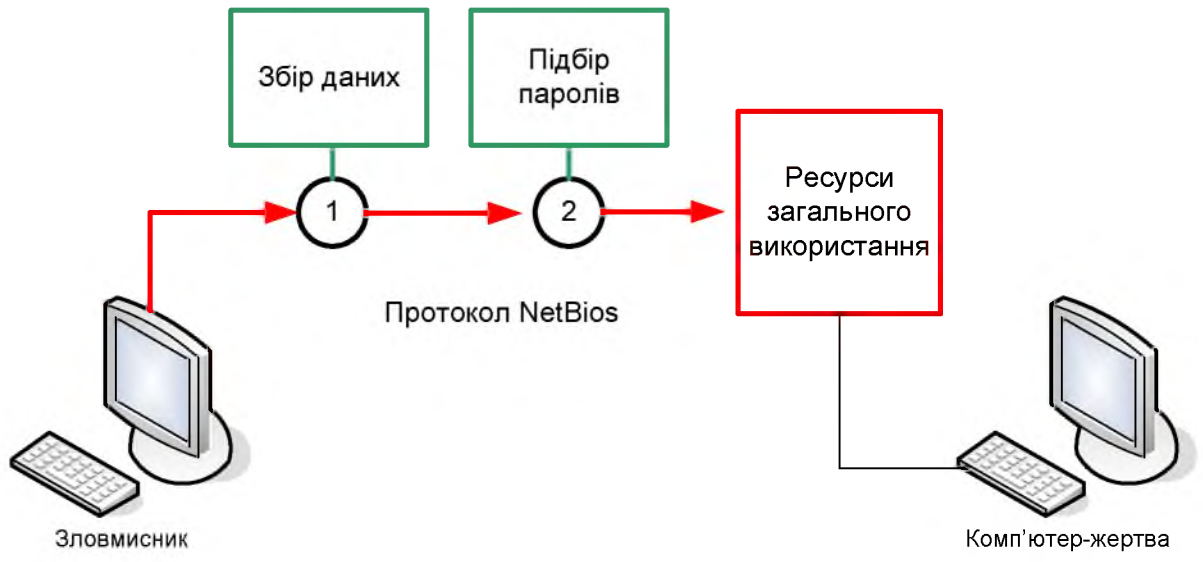

Рисунок 1.6 – Схема проведення NetBios-атаки

1.6 Висновок

Виявлення вторгнень – це активний процес, при якому відбувається виявлення зловмисника при його спробах проникнути в систему. Виявлення вторгнень допомагає при превентивній ідентифікації активних загроз за допомогою сповіщень і запобігань про те, що зловмисник здійснює збір інформації, необхідної для проведення атаки.

Актуальність теми магістерської дипломної роботи полягає у тому, що впровадження в ІКС системи виявлення вторгнень, є одним з головних заходів щодо мінімізування ймовірності проведення атак, що дозволяє при якісному обслуговуванні значно знизити загрозу непомітної вдалої реалізації мережевої атаки.

Були розглянуті основні етапи реалізації атаки на мережу. Розглянуті потенційно небезпечні атаки, та їх класифікація.

РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА

2.1 Аналіз роботи локальної обчислювальної мережі «АрдКом»

2.1.1 Загальні відомості про роботу підприємства

ПП «АрдКом» займається оптовою торгівлею канцтоварами у Дніпропетровської області.

В локальній обчислювальної мережі підприємства знаходиться 31 робоча станція і 1 файл-сервер, а також мережевий принтер.

Штат співробітників підприємства складається з 32-ох осіб:

– директор – 1 чол.;

– секретар – 2 чол.;

– заступник директора з постачання – 1 чол.;

– заступник директора з економічних питань – 1 чол.;

– заступник директора з експлуатації – 1 чол.;

– головний менеджер – 1 чол.;

– відділ головного менеджера – 3 чол.;

– бухгалтерія – 4 чол.;

– планово-економічний відділ – 4 чол.;

– профспілковий комітет – 1 чол.;

– юридичний відділ – 3 чол.;

– відділ кадрів – 2 чол.;

– відділ охорони праці – 3 чол.;

– головний бухгалтер – 1 чол.;

– заступник головного бухгалтера – 1 чол.;

– адміністратор комп’ютерної мережі – 2 чол.;

– прибиральниця – 1 чол.

Графік роботи підприємства і його працівників:

На підприємстві встановлено п'ятиденний робочий тиждень.

Режим роботи: з 8.00 до 17.00, перерва – з 12.00 до 13.00, вихідні: субота, неділя.

Прибирання приміщень здійснюється з 7.00 до 8.00 щодня, крім вихідних.

Загальна характеристика будівлі:

ПП «АрдКом» розташоване на шостому поверсі шестиповерхового будинку за адресою м. Дніпро, пр. Б.Хмельницького, 227. Займана площа: 789 м^2.

На півночі від будівлі, в якій розташоване підприємство, знаходиться шестиповерховий житловий будинок (відстань – 20 м).

На півдні від будівлі, в якій розташоване підприємство, знаходиться проїзна частина (відстань – 7 м).

На сході від будівлі, в якій розташоване підприємство, знаходиться шестиповерхова будівля готелю (відстань – 20 м).

На заході від будівлі, в якій розташоване підприємство, знаходиться шестиповерховий офісний будинок (відстань – 25 м).

Телефон: Використовується постачальник послуг «Укртелеком».

Интернет: Використовується постачальник послуг "Fregat".

В офісі спроєктоване централізоване опалення, каналізація, електрика, водопровід, вентиляція.

Матеріал зовнішніх стін - шлакобетон, товщина – 500 мм.

Матеріал внутрішніх стін - гіпсокартон, товщина – 200 мм.

Вікна - металопластикові склопакети, ширина від 600 до 1000 мм, висота 1500 мм., Товщина скла 150 мм.

Двері – дерев'яні соснові, ширина 950 мм, висота 2100 мм.

Опис кімнат

– кабінет директора (площа 27 м^2);
– кабінет секретаря директора (площа 21 м^2);
– кабінет заступника директора з постачання (площа 27 м^2);
– кабінет головного менеджера (площа 27 м^2);
– кабінет секретаря заступника директора з постачання та головного менеджера (площа 21 м^2);
– кабінет заступника директора з експлуатації (площа 21 м^2);
– відділ головного менеджера (площа 48 м^2);

– бухгалтерія (площа 36 м^2);
– планово-економічний відділ (площа 57 м^2);
– кімната профспілкового комітету (площа 27 м^2);
– юридичний відділ (площа 21 м^2);
– кімната відділу кадрів (площа 48 м^2);
– кімната відділу охорони праці (площа 36 м^2);
– кабінет заступника директора з економічних питань (площа 48 м^2);
– кабінет головного бухгалтера (площа 27 м^2);
– кабінет заступника головного бухгалтера (площа 21 м^2);
– відділ інформаційних технологій (площа 27 м^2);
– туалет (площа 21 м^2);
– коридор (площа 192 м^2).

Схема локальної обчислювальної мережі ПП «АрдКом» наведена на рисунку 2.1.

Рисунок 2.1 – Схема локальної обчислювальної мережі ПП «АрдКом»

2.1.2 Характеристика обчислювальної підсистеми КМ

Метою створення комп'ютерних систем підприємства є надання будь-якому користувачеві, у відповідності з захищеною технологією обробки

інформації, потенційної можливості доступу до інформаційних ресурсів всіх комп'ютерів, які об'єднані в обчислювальну мережу.

Узагальнена функціонально-логічна структура обчислювальної системи КМ підприємства включає:

– підсистему обробки інформації;

– підсистему взаємодії користувачів з КМ;

– підсистему обміну даними.

Підсистема обробки інформації реалізує головну цільову функцію КМ і складається із засобів обробки інформації, які утворюють основу інформаційно-обчислювальних ресурсів КМ, що надаються користувачам (обчислення, пошук, зберігання та обробка інформації). Принциповими її особливостями є багатофункціональність і можливість доступу до неї для будь-яких робочих станцій КМ.

Підсистема взаємодії користувачів з КМ забезпечує користувачам доступ до засобів підсистеми обробки інформації і подання отриманого від них ресурсу у вигляді результату обчислення, інформаційного масиву або графічного зображення у зручній і зрозумілій для користувача формі.

Підсистема обміну даними складається з пасивної мережі для обміну даними (кабельна мережа), активного мережного обладнання (комутатора), яке об'єднує в єдине ціле пасивну мережу з обладнанням інших підсистем для забезпечення інформаційної взаємодії.

2.1.3 Типові адміністративні та організаційні вимоги до КМ підприємства, що стосуються питань ТЗІ

Сервери, робочі станції, периферійні пристрої, інші технічні засоби обробки інформації з обмеженим доступом до інформаційно-комунікаційної мережі ПП «АрдКом» категоровані згідно з вимогами нормативних документів з технічного захисту інформації.

Програмне забезпечення КМ, яке задіяне в ІКС, має підтвердження відповідності нормативним документам із захисту інформації (сертифікат

відповідності) і використовуються відповідно до вимог, визначених цими документами.

2.1.4 Характеристика фізичного середовища КМ

У загальному випадку КМ є територіально розосередженою системою, фізичне розташування компонентів якої можна представити як ієрархію, яка включає:

– територію, на якій вона знаходиться;

– будинок, який знаходиться на території;

– окреме приміщення в межах будівлі.

Будівля, в якій знаходиться ПП «АрдКом» представляє собою окремий 6-поверховий будинок, що знаходиться в помірній зоні. Контроль доступу людей в КЗ забезпечується на першому поверсі при вході (турнікети, кімната охорони). Об'єкт укомплектований необхідними засобами резервного енергозабезпечення та сигналізації. Поблизу будівлі немає представництв іноземних держав.

2.1.5 Характеристика користувачів КМ

За рівнем повноважень щодо доступу до інформації, характером та складом робіт, які виконуються в процесі функціонування комп'ютерних систем, особи, які мають доступ до КМ, поділяються на наступні категорії:

– користувачі, яким надано повноваження забезпечувати управління КМ та розробляти й супроводжувати СЗІ (адміністратор комп'ютерної мережі);

– користувачі, яким надано право доступу до інформації з обмеженим доступом одного або декількох класифікаційних рівнів (директор, заступники директора, головний інженер, головний бухгалтер та його заступник, бухгалтерія, планово-економічний відділ, юридичний відділ, відділ головного менеджера, відділ кадрів);

– користувачі, яким надано право доступу тільки до відкритої інформації (секретарі, профспілковий комітет, відділ охорони праці);

– технічний персонал, який здійснює повсякденне підтримання життєдіяльності фізичного середовища КМ (менеджери, технічний персонал з обслуговування будівель, ліній зв'язку і т.д.).

2.1.6 Характеристика інформації, що обробляється в КМ

В КМ ПП «АрдКом» обробляється інформація з обмеженим доступом, володіти, користуватися чи розпоряджатися якою можуть окремі фізичні і (або) юридичні особи, які мають доступ до неї у відповідності з правилами, встановленими власником цієї інформації. До цього виду інформації належать:

– база даних підприємства (клієнти, співробітники, товарно-матеріальні цінності, фінансові операції);

– бухгалтерська документація;

– внутрішня службова документація (акти, накази, інструкції).

В КМ також зберігається і циркулює відкрита інформація, яка не потребує захисту, або захист якої забезпечувати недоцільно. До цього виду інформації належить прайс лист продукції яка реалізується, особиста інформація співробітників підприємства, графік роботи працівників.

2.1.7 Матриця доступу користувачів мережі до інформації з обмеженим доступом

Таблиця 2.2 – Матриця доступу

Користувачі	База даних підприємства	Бухгалтерська документація	Внутрішня службова документація
Директор	RWD	RWD	RWD
Заступники директора	RWD	RWD	RWD
Головний менеджер	RW	R	RW
Головний бухгалтер	RW	RW	R
Відділ головного менеджера	RW	R	R
Бухгалтерія	R	RW	R
Економічний відділ	RW	R	R

Продовження таблиці 2.2

1	2	3	4
Профспілковий комітет	R	R	R
Юридичний відділ	R	R	RW
Відділ кадрів	R	R	R
Відділ безпеки руху та охорони праці	R	R	R
Заступник головного бухгалтера	RW	RW	R
Адміністратор комп'ютерної мережі	R	R	R

R – користувач має право читати інформацію даного ресурсу;

W – користувач має право змінювати інформацію даного ресурсу;

D – користувач має право видаляти інформацію з даного ресурсу.

В таблиці 2.2 наведена матриця доступу користувачів інформаційно-комунікаційної мережі, які мають доступ до інформації з обмеженим доступом.

2.1.8 Характеристика програмного забезпечення, що використовується в процесі роботи КМ

В інформаційно-комунікаційній мережі підприємства використовується наступне програмне забезпечення:

– ОС серверних станцій: Ubuntu Server;

– ОС робочих станцій: Microsoft Windows 7;

– 1С Бухгалтерия 8;

– Microsoft Office 365;

– Avast.

2.1.9 Характеристика захищеності КМ підприємства від атак

На даний час на підприємстві майже відсутні засоби захисту ІКС від атак. Антивірус не може забезпечити потрібний захист інфомрації у разі

цілеспрямованої атаки зловмисника. Це тягне за собою ризики, пов'язані з можливою втратою та (або) спотворенням інформації, що в свою чергу призведе до матеріальних збитків підприємства.

2.2 Вибір профілю захищеності

Відповідно до НД ТЗІ 2.5-005-99 та НД ТЗІ 2.5-004-99 для ПП «АрдКом» було обрано наступний профіль захищеності:

3.КЦД.1 = { КД-2, КО-1, КВ-1,
ЦД-1, ЦО-1, ЦВ-1,
ДР-1, ДВ-1,
НР-2, НИ-2, НК-1, НО-2, НЦ-2, НТ-2, НВ-1 }

Це стандартний функціональний профіль захищеності інформації в КМ, який входить в склад АС класу 3 з підвищеними вимогами до забезпечення конфіденційності, цілісності і доступності інформації, що обробляється в КМ.

Довірча конфіденційність (КД-2)

Ця послуга застосовується для розмежування доступу користувачів до захищених об'єктів і дозволяє користувачу керувати потоками інформації в АС від захищених об'єктів, що належать його домену, до інших користувачів. Ця послуга може реалізовуватися лише у разі необхідності та у доповнення до послуги КА-2, яка визначає основний механізм розмежування доступу до конфіденційної інформації в АС класу 3.

Конфіденційність при обміні (КВ-1)

Ця послуга дозволяє забезпечити конфіденційність інформації з обмеженим доступом, що передається по незахищеним каналам за межами інформаційної системи. Забезпечується завдяки створенню VPN-каналу.

Повторне використання об'єктів (КО-1)

Ця послуга дозволяє забезпечити коректність повторного використання розділюваних об'єктів, гарантуючи, що в разі, якщо розділюваний об'єкт виділяється новому користувачу або процесу, він не містить інформації, яка залишилась від використання його попереднім користувачем або процесом.

Політика повторного використання об'єктів, що реалізується КЗЗ, стосується тільки тих об'єктів АС, які містять конфіденційну інформацію і ресурси яких поділяються між користувачами АС та прикладними процесами, що виконуються в АС.

Довірча цілісність (ЦД-1)

Ця послуга застосовується для захисту оброблюваної інформації від несанкціонованої модифікації і дозволяє користувачу будь-якої категорії керувати потоками інформації в АС від інших користувачів до захищених об'єктів, що належать його домену. Ця послуга може реалізовуватися лише у разі необхідності та у доповнення до послуги ЦА-2 (ЦА-1), яка визначає в АС класу 2 основний механізм захисту від несанкціонованої модифікації об’єктів, які містять конфіденційну інформацію.

Цілісність при обміні (ЦВ-1)

Ця послуга дозволяє забезпечити цілісність інформації з обмеженим доступом, що передається по незахищеним каналам за межами інформаційної системи. Забезпечується завдяки створенню VPN-каналу.

Відкат (ЦО-1)

Ця послуга забезпечує можливість відмінити окрему операцію або послідовність операцій й повернути захищений об'єкт, з яким маніпулював користувач, до попереднього наперед визначеного стану.

Використання ресурсів (ДР-1)

Ця послуга дозволяє керувати використанням користувачами послуг та ресурсів. Обмеження щодо використання окремим користувачем та/або процесом обсягів обчислювальних ресурсів АС або кількості об’єктів встановлюються адміністратором безпеки або користувачами, яким надано повноваження інших адміністраторів. Запити на зміну встановлених обмежень повинні оброблятися КЗЗ тільки в тому випадку, якщо вони надходять від адміністраторів.

Відновлення після збоїв (ДВ-1)

Політикою відновлення після збоїв повинна бути визначена й задокументована множина типів відмов і переривань обслуговування АС або окремих її компонентів, після яких можливе повернення у відомий захищений стан без порушення політики безпеки. Для кожної з відмов повинні бути чітко вказані рівні відмов, у разі перевищення яких необхідна повторна інсталяція АС.

Реєстрація/аудит (НР-2)

Послуга реєстрації рівня НР-2 дозволяє контролювати небезпечні для АС дії зі сторони користувачів будь-яких категорій відносно процесів і об'єктів, що існують в АС і стосуються захищених об’єктів. Адміністратор безпеки і користувачі, яким надано повноваження інших адміністраторів, повинні мати в своєму розпорядженні засоби перегляду і аналізу журналу реєстрації, а КЗЗ повинен забезпечувати захист журналу реєстрації від несанкціонованого доступу, модифікації або руйнування.

Ідентифікація та автентифікація (НИ-2)

Ідентифікація та автентифікація дозволяють визначити й перевірити особу користувача будь-якої категорії, що намагається одержати доступ до АС або до захищених об’єктів, та повинні гарантувати, що доступ може бути надано тільки авторизованному користувачу. Дозвіл на виконання будь-яких дій, що контролюються КЗЗ, користувач отримує тільки після автентифікації його КЗЗ на підставі введеного ним пароля.

Достовірний канал (НК-1)

Ця послуга повинна гарантувати користувачу будь-якої категорії можливість безпосередньої взаємодії з КЗЗ, а також те, що ніяка взаємодія користувача з АС не може бути модифікованою іншим користувачем або процесом. Послуга повинна визначати вимоги до механізму встановлення достовірного зв'язку між користувачем і КЗЗ. Достовірний канал повинен використовуватися для початкової ідентифікації та автентифікації. Зв'язок із використанням даного каналу повинен ініціюватися виключно користувачем.

Розподіл обов’язків (НО-2)

Ця послуга дозволяє розмежувати повноваження користувачів, визначивши категорії користувачів із певними й притаманними для кожної з категорій функціями (ролі). Послуга призначена для зменшення потенційних збитків від навмисних або помилкових дій користувачів й обмеження авторитарності керування АС.

Цілісність КЗЗ (НЦ-2)

Ця послуга визначає міру здатності КЗЗ захищати себе і гарантувати свою спроможність керувати захищеними об'єктами. Повинен бути визначений механізм контролю цілісності компонентів, що входять до складу КЗЗ. У разі виявлення порушення цілісності будь-якого зі своїх компонентів КЗЗ повинен повідомити щодо цього адміністратора безпеки або уповноваженого співробітника СЗІ і перевести АС до стану, в якому забороняється обробка конфіденційної інформації. Повернути АС до нормального функціонування може тільки адміністратор безпеки або уповноважений співробітник СЗІ після відновлення відповідності цього компонента КЗЗ еталону.

Самотестування (НТ-2)

Самотестування дозволяє КЗЗ перевірити й на підставі цього гарантувати правильність функціонування і цілісність множини функцій АС, що забезпечуються захистом. До складу КЗЗ повинен входити набір тестових процедур, достатній для оцінки правильності виконання в АС всіх критичних для безпеки конфіденційної інформації та технологічної інформації КСЗІ функцій, а сам КЗЗ повинен бути здатним контролювати їх виконання. Тести повинні виконуватися при ініціалізації КЗЗ за запитом адміністратора безпеки або уповноважених співробітників СЗІ. У разі некоректного виконання якогось із тестів КЗЗ повинен перевести АС до стану, в якому забороняється обробка конфіденційної інформації взагалі, або до стану, в якому забороняється обробка конфіденційної інформації з використанням послуг безпеки, для яких тест не було виконано. Повернути АС до нормального функціонування може тільки адміністратор безпеки або уповноважений співробітник СЗІ після відновлення працездатності КЗЗ і повторного виконання повного набору тестів.

Автентифікація при обміні (НВ-1)

Ця послуга забезпечує автентифікацію обох сторін обміну перед передаванням інформації з обмеженим доступом. Реалізовується завдяки програмному забезпеченню, яке відвовідає за створення VPN-каналу.

2.3 Аналіз інформаційних загроз підприємства

Так як в інформаційній системі ПП «АрдКом» обробляється інформація з обмеженим доступом, то, отже, по відношенню до цієї інформації існують загрози.

Класифікація рівнів збитку внаслідок реалізації загрози:

– критичний – інформація може бути знищена, змінена без можливості відновлення (втрати підприємства в даному випадку дуже великі);

– високий – інформація втрачає деякі свої властивості, може бути відновлена, втрати підприємства в цьому випадку менше тих, які були б внаслідок повної втрати і неможливості відновити інформацію;

– середній – інформація втрачає деякі властивості, але може бути відновлена в прийнятні терміни і з мінімальними втратами;

– незначний – частина інформаційного ресурсу втрачає деякі зі своїх властивостей, які можливо відновити в найкоротший термін (крім конфіденційності).

Класифікація ступенів ймовірності виконання загрози:

– висока – прогнозується ймовірне виконання загрози декілька разів на місяць;

– середня – виконання загрози ймовірне декілька разів на рік;

– низька – ймовірність виконання загрози – до одного разу на рік.

Проаналізуємо перелік загроз, який наведено в таблиці 2.3.

Таблиця 2.3 – Аналіз інформаційних загроз

Вид загрози	Ймовірність	Що порушує	Рівень збитків	Вразливості
Крадіжка а) інформації; б) засобів доступу (ключі, паролі)	висока	К	високий	Відсутність засобів захисту інформації від несанкціонованого доступу
Підміна (модифікація) а) операційних систем; б) систем управління базами даних; в) програмного забезпечення; г) інформації (даних); д) паролей та правил доступу	середня	Ц	незначний	Відсутність засобів виявлення несанціонованого підключення для КМ; недостатній мережевий моніторинг
Знищення а) програмного забезпечення (ОС, СУБД, програмного забезпечення); б) інформації (файлів, даних); в) паролей та ключової інформації	низька	Ц,Д	високий	Відсутність засобів виявлення мережевого вторгнення; відсутність засобів протидії мережевим атакам
Порушення нормальної роботи (аномалії) а) швидкості обробки інформації; б) пропускної здатності каналів зв'язку; в) об'ємів оперативної пам'яті; г) об'ємів вільного дискового простору	середня	Д	середній	Недостатня захищеність мережі проти атак зловмисників; неправильне налаштування програмного забезпечення.
Порушення встановлених правил доступу	середня	К,Ц,Д	середній	Вразливості робочіх станцій та серверного ПО внаслідок їх недосконалого налаштування.

Приведена модель загроз включає в себе тільки антропогенні загрози, тому що будь-яка атака на мережу виконуєтсья зловмисником, тобто людиною.

Техногенні та стихійні загрози не мають відношення до мережевих вторгнень і тому не в даній моделі загроз не розглядаються.

З урахуванням усього вищевикладеного, впровадження заходів щодо комплексної системи виявлення мережевих вторгнень на інформаційну систему ПП «АрдКом», є виправданим кроком, оскільки це є контрзаходом по відношенню до всіх критичних загроз інформаційної мережі згідно наведеного вище аналізу загроз.

2.4 Вибір програмного забезпечення

Програмне забезпечення, яке буде використане для синтезу системи виявлення мережевих вторгнень на ІКС, можна поділити на декілька категорій:

– програмне забезпечення, необхідне для виявлення та усунення ARP-спуфінгу;

– програмне забезпечення, необхідне для шифрування мережевого трафіку;

– програмне забезпечення, необхідне для виявлення мережевих вторгнень.

Далі буде розглянуто кожну категорію окремо та вказано програмне забезпечення, яке буде застосоване у проєктованій системі захисту.

2.4.1 Програмне забезпечення, необхідне для виявлення та усунення ARP-спуфінгу

Arpwatch

Arpwatch – програма, яка відстежує відповідність між IP і MAC-адресами, і при виявленні аномалій повідомляє про це в Syslog. Використовується як один з інструментів для боротьби з Arp-спуфінгом.

Програма аналізує arp-відповіді на мережевому інтерфейсі, до якого вона прив'язана, та запам'ятовує відповідність ip-адрес і mac-адрес. Як тільки програма бачить, що відповідність порушена, або виявляє появу нових адрес в мережі, вона повідомляє про це в системний журнал (syslog).

Ім'я інтерфейсу, на якому виконується прослухування, задається як аргумент командного рядка при запуску програми. Якщо вона запускається

стартовим скриптом, то інтерфейс вказується там, де в операційній системі прийнято вказувати ключі для програм, що запускаються.

Основний недолік arpwatch полягає в тому, що програма повинна працювати на хостах, які вона захищає або хоча б на маршрутизаторі, який веде в мережу, що захищається (у останньому випадку машини, які працюють в тій же мережі, де і зловмисник, не захищаються; захищаються лише машини, що знаходяться за маршрутизатором).

2.4.2 Програмне забезпечення, необхідне для шифрування мережевого трафіку

Stunnel

Stunnel це програма, яка встановлюється як на серверній, так і на клієнтській стороні, щоб з'єднати два порти (клієнтський і серверний) шифрованим каналом. Її часто використовують, щоб додати захищений порт вебсерверу, серверу, який не підтримує SSL-шифрування, або поштовому серверу, що працює лише на POP3 і лише на 110-му порті. Серверний скрипт "встановлюється" на порт 443 і передає (усередині сервера) всі запити на порт 80 – в разі вебсервера. Весь трафік від клієнта до сервера йде по надійному каналу, а внутрішній трафік ніхто прослухати вже не зможе, не зламавши сам сервер.

Майже всі інші програми, що шифрують трафік у мережі, використовують технології створеннь VPN-каналів та аналогічні технології. В даному випадку такі програми не потрібні для мінімізування ймовірності проведення атак у мережі, тому що мережа відноситься до АС другого классу, що говорить про те, що вона є локалізованою і налаштування захищеного каналу для з'єднання з іншою мережею не потрібне.

2.4.3 Програмне забезпечення, необхідне для виявлення мережевих вторгнень

Snort

Snort є вільною мережевою системою запобігання вторгненням (IPS) і мережевою системою виявлення вторгнень (IDS) з відкритим вихідним кодом,

здатною виконувати реєстрацію пакетів і в реальному часі здійснювати аналіз трафіку в IP-мережах. Snort виконує протоколювання, аналіз, пошук по вмісту, а також широко використовується для активного блокування або пасивного виявлення цілого ряду нападів і зондувань, таких як переповнювання буфера, стелс-сканування портів, атаки на вебсервера, smb-зондування і спроби визначення ОС. Програмне забезпечення в основному використовується для запобігання проникненню та блокування атак, якщо вони мають місце.

В таблиці 2.4 наведена звідна характеристика проаналізованого програмного забезпечення згідно з обраними критеріями.

Таблиця 2.4 – Звідна таблиця програмного забезпечення

Назва	Комерційне або безкоштовне	Інтерфейс (суб'єктивна оцінка)	Споживання ресурсів	Складність налаштування
Програмне забезпечення, необхідне для виявлення та усунення ARP-спуфінгу				
Arpwatch	безкоштовне	«відмінно»	5%	мала
Програмне забезпечення, необхідне для виявлення мережевих вторгнень				
Snort	безкоштовне	«задовільно»	20%	висока
Програмне забезпечення, необхідне для шифрування мережевого трафіку				
Stunnel	безкоштовне	«добре»	5%	середня

Інтерфейс програми Snort після безпосередньої інсталяції не можна назвати зручним, але після інсталяції декількох необхідних плагінів інтерфейс набагато покращується.

Таким чином в інформаційній мережі ПП «АрдКом» для забезпечення синтезу системи виявлення мережевих вторгнень були обрані наступні програми:

1) для протидії ARP-спуфінгу обрана програма Arpwatch.
2) для шифрування мережевого трафіку обрана програма Stunnel.

3) для протидії мережевим вторгненням обраний програмний комплекс Snort. Хоча на перший погляд здається, що програма дуже складна в налаштуванні й компонуванні, вона має декілька величезних переваг, які будуть освітлені далі.

2.5 Організація заходів щодо забезпечення захищеності робочих станцій для ОС Windows

Одна з основних причин, завдяки якій ОС Windows заслужила репутацію найбільш вразливої системи, полягає в невірному адмініструванні. Більшість проблем виникає завдяки «спрощеному» адмініструванню, яке забезпечується самою ОС Windows і повністю звільняє адміністратора від відстежування внутрішньої роботи середовища, тим самим вириваючи можливість управління з його рук.

2.5.1 Перевірка серверів та оновлень, що використовуються на даний момент

Недолік надійних вбудованих сценаріїв і можливостей віддаленого доступу робить ОС Windows незручною для автоматизації. Проте, перш ніж пробувати відновити систему, спочатку необхідно з'ясувати, які оновлення в ній вже виконані. Інакше можна даремно витратити сили і час, виконуючи оновлення, яке абсолютно не потрібне. Вочевидь, що із збільшенням кількості обслуговуваних систем ця проблема стає усе більш складною. Уникнути зайвих зусиль ручного оновлення систем можна, використовуючи засіб Hfnetchk, який спочатку був автономною програмою, розробленою компанією Shavlik Technologies. Тепер ця програма входить до складу аналізатора Baseline Security Analyzer компанії Microsoft і стає доступною після запуску з командного рядка інтерфейсу mbsacli.exe. Hfnetchk може перевіряти не лише статус Windows Server та Windows, але і те, чи були виконані критичні оновлення IIS (пропріетарний

набір серверів для декількох служб Інтернету від компанії Майкрософт), SQL Server, Exchange Server, Media Player та IE.

Хоча ця програма лише перевіряє статус оновлення системи, а не виконує її оновлення, вона є незамінним інструментом, що зберігає час. Hfnetchk працює за рахунок завантаження підписаного і стислого xml-файлу від компанії Microsoft, що містить відомості про всі оновлення, що існують на даний момент. У файлі містяться контрольні суми і версії файлів, які обробляються в ході кожного оновлення, а також ключі реєстру, що оновлюються в ході кожного оновлення. В ньому є також й інша значуща додаткова інформація. В ході сканування системи Hfnetchk спочатку перевіряє ключі реєстру, пов'язані з більшістю оновлень, доступних для поточної конфігурації системи. Якщо які-небудь з ключів пропущені або не відповідають вмісту xml-файлу, буде встановлений прапор необхідності їх оновлення. Якщо ключ існує і його вміст відповідає xml-файлу, то Hfnetchk перевірить наявність вказаного файлу, а також його версію і контрольну суму.[8]

В разі неспівпадання контрольної суми буде встановлений прапор необхідності оновлення. Всі прапори необхідності оновлення будуть виведені в звіті разом із заслannями на пункт бази Microsoft Knowledge Base, що містить додаткові відомості про необхідне оновлення.

Для встановлення Hfnetchk спочатку необхідно завантажити і встановити Baseline Security Analyzer. Для запуску Hfnetchk в режимі командного рядка необхідно вказати каталог, створений в ході установки (за умовчанням C:\program Flles\Microsoft Baseline Security Analyzer). Для перевірки необхідності оновлення локальної системи потрібно виконати команду:

C:\> Program Files\Microsoft Baseline Security Analyzer> mbsacli /hf

Рисунок 2.2 – Результат роботи програми MBSA

На екран буде виведено інформацію про стан оновлень у вигляді таблиць, що приведені на рисунку 2.2. У першому стовпці буде вказано, яке саме оновлення виконане невдало. У другому стовпці буде вказано результат невдалої перевірки певного оновлення. В третьому – номер пункту в базі знань Microsoft Knowledge Base, в якому міститься додаткова інформація про проблему, що виправляється певним оновленням, а в четвертому – ступінь важливості даного оновлення. Якщо необхідно дізнатися більше про конкретну невдалу перевірку, команду необхідно запускати з ключем –у.

При скануванні локальної системи потрібні повноваження адміністратора. При скануванні віддаленої машини потрібні повноваження адміністратора віддаленої машини. Існує декілька варіантів сканування віддалених комп'ютерів. Для сканування одиночної віддаленої системи як значення ключа -h має бути вказане netbios-ім'я. Так само як значення ключа -i може бути вказана ip-адреса системи.

Перевірку декількох систем можна виконати декількома способами. При використанні параметра -fh можна вказати файл, що містить до 256 netbios-імен (поодинці в рядку) систем, які будуть перевірені. Аналогічно є можливість за допомогою параметра –fip вказати ip-адреси. А за допомогою параметра –r можна вказати діапазон ip-адрес. Для локальної мережі ПП «АрдКом» для сканування систем з ip-адресамі від 192.168.1.1 до 192.168.1.32 використовується наступна команда: mbsacli /hf –r 192.168.1.1 - 192.168.1.32.

Всі параметри дуже зручні, і їх можна використовувати в будь-якому поєднанні. Окрім вказівки віддаленої системи за netbios-ім'ям або ip-адресі, за допомогою параметра -d можна сканувати системи за ім'ям домена, а за допомогою параметра -n можна виконати сканування всього сегменту локальної мережі. При скануванні систем, що виконується з особистої робочої станції, можуть згодитися параметри -u і -p. При зверненні до віддаленої системи вони дозволяють відразу вказувати ім'я користувача і пароль. Ці ключі особливо корисні, якщо вхід в систему здійснюється, не використовуючи обліковий запис адміністратора. Звичайно ж, вказаний в параметрі -u обліковий запис повинен мати привілеї адміністратора віддаленої системи. При скануванні великої кількості систем зручний також параметр -t. Він дозволяє вказувати число потоків, використовуваних сканером. Збільшення кількості потоків зазвичай призводить до підвищення швидкості сканування. Допускаються значення від 1 до 128; за замовчанням – 64. При скануванні більш однієї машини на екран виводиться величезна кількість даних. Параметр -f дозволяє вказати файл, де зберігатимуться результати сканування, які можна пізніше проглянути за допомогою звичайного текстового редактора. Hfnetchk – дуже гнучкий інструмент, який може використовуватися для досить швидкої перевірки необхідності оновлення системи на безлічі машин. Ця програма особливо корисна при появі нового комп'ютерного «хробака», коли необхідно переконатися, що всі системи мають свіже оновлення, що захищає від дії цього «хробака».

2.5.2 Здобуття переліку відкритих файлів і процесів, що володіють ними

Якщо користувач переглядає в менеджеі завдань список процесів, не запускаючи на робочій станції сторонніх завдань, і помічає процес, який раніше не бачив, це вже вказує на те, що захищеність робочої станції порушена. Якщо встановлена не ОС Windows, можна проконтролювати діяльність процесу, проглянувши файли, які він відкриває. На жаль, Windows не дозволяє зробити це. Компанія Sysinternals розробила відмінний інструмент під назвою Handle. Програма Handle дуже схожа на Isof (Unix-система), але вона може виводити перелік оперативних ресурсів різних типів, включаючи потоки, події та семафори. Крім того, ця програма виводить відкриті ключі реєстру. Запуск Handle з командного рядка без параметрів представить перелік всіх відкритих дескрипторів файлів.

Для того, щоб дізнатися, які процеси звертаються до певного файлу в даний момент, можна вказати ім'я конкретного файлу:

C:\> handle ім’я файлу

Крім того, для перегляду всіх типів ресурсів можна скористатися параметром –a.

На рисунку 2.3 наведено результат роботи програми Handle (перевіряється процес firefox.exe).

Handle – дуже потужний засіб, і в командному рядку може одночасно вказуватися декілька різних параметрів для швидкого пошуку саме того файлу, який необхідний. [9]

2.5.3 Перелік запущених служб та відкритих портів

Програма Fport компанії Foundstone дозволяє швидко і просто проглянути відкриті порти. Fport має декілька параметрів, що визначають в основному сортування результатів. Якщо необхідно відсортувати результат на ім'я додатків, необхідно використовувати параметр /a; за номером ідентифікатора процесу – /i. Хоча в Fport не так багато функцій, як в схожої по функціональності програми для unix-систем netstat, ця програма дійсно виконує свою роботу. Для здобуття

переліку всіх портів, відкритих в даний момент в системі, просто потрібно ввести команду fport.

Рисунок 2.3 – Результат виконання програми Handle

На рисунку 2.4 приведено результат роботи програми fport.

При необхідності сортування переліку за номером порту слід використовувати ключ /p. Слід звернути увагу на те, що в результуючій таблиці можуть бути представлені деякі процеси (navapw32, putty і IEXPLORE), які виглядають не як служби. Вони можуть виявитися у вихідному списку через те, що Fport виводить всі відкриті порти, а не лише порти, для яких запущені процеси-«слухачі».

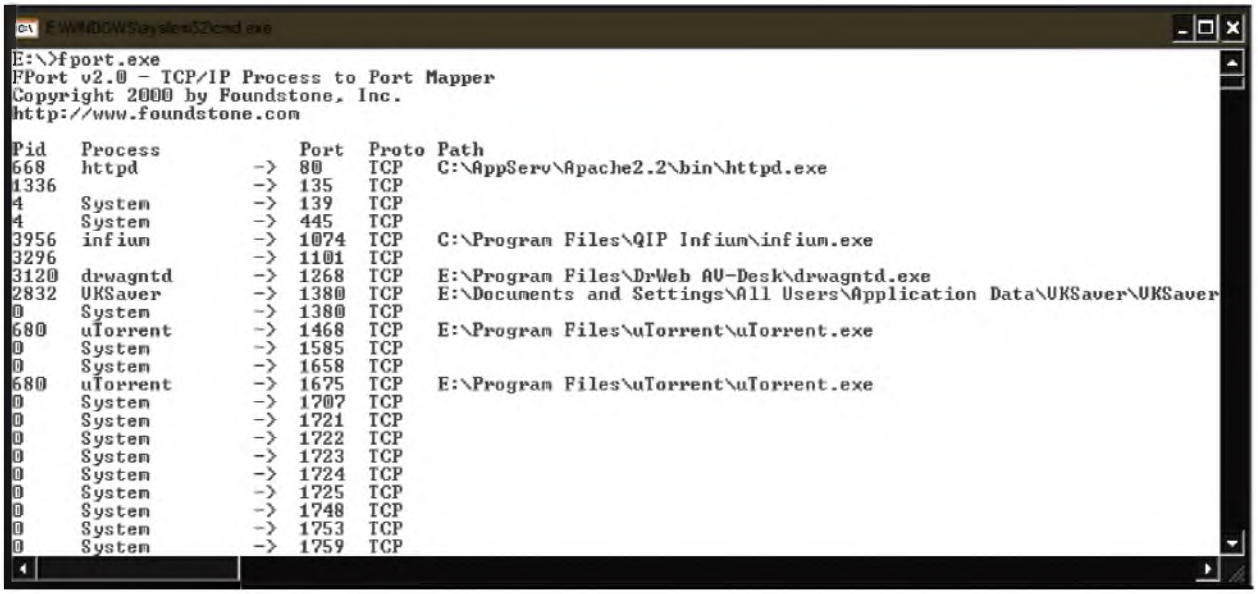

Рисунок 2.4 – Результат виконання програми fport

Не дивлячись на те, що програма Fport не настільки потужна, як деякі команди, доступні в інших операційних системах, все одно вона – надійний, швидкий і простий у використанні засіб, що є хорошим доповненням ОС Windows.

2.5.4 Включення аудиту

Windows володіє дуже потужними функціями протоколювання, але, на жаль, за умовчанням всі вони відключені. У Windows Server це виправлено включенням ряду функцій за умовчанням, але все таки розумно перевірити, що відстежується саме те, що необхідне. За допомогою вбудованих в ОС Windows функцій можна спостерігати за невдалими спробами входу в мережу, відстежувати події управління обліковими записами, доступу до файлів, використання привілей і тому подібне. Можна також протоколювати зміну політик безпеки і всі системні події. Для включення аудиту в будь-якій з перерахованих областей потрібно знайти значок Засобів адміністрування (Administrative Tools) панелі інструментів і двічі клацнути на ньому. Після цього знайти значок Локальна політика безпеки (Local Security Policy) і двічі клацнути на ньому. Розкрити вузол дерева Локальні політики (Local Policies). Тепер можна перейти до будь-якої політики і перевірити включення протоколювання успішної

або невдалої події. Це можна зробити, двічі клацнувши в правій частині вікна на політиці, яку необхідно змінити. Після цього відкриється діалогове вікно, в якому можна включити ибо відключити аудит.

Результат виконання зазначених дій приведено на рисунку 2.5.

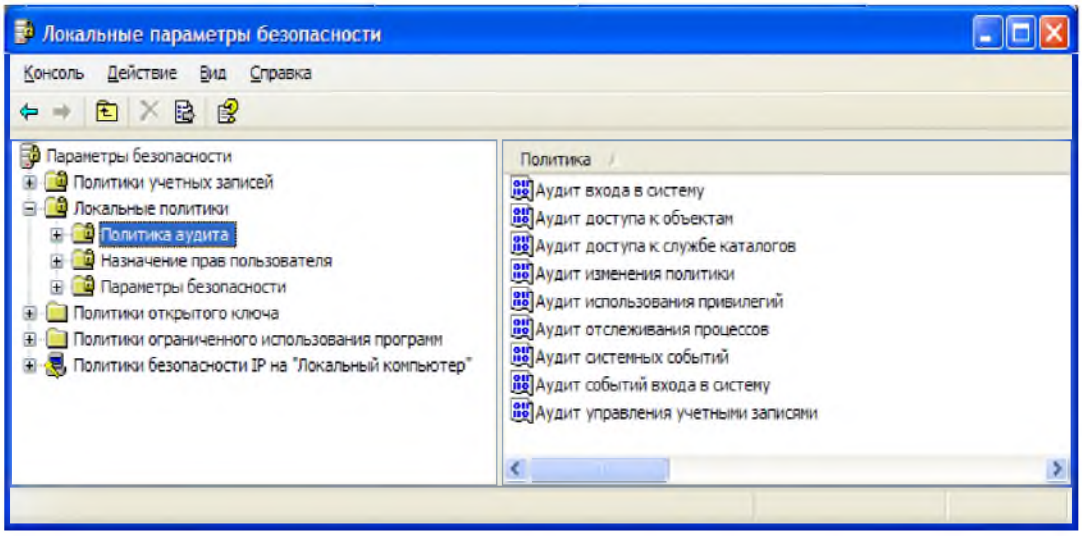

Рисунок 2.5 – Перелік політик аудиту

Якщо залишити аудит відключеним, то протоколювання виконуватися не буде, тому необхідно включити аудит всіх політик. Після включення аудиту для конкретної політики необхідно переглядати пункти журналу подій при виникненні конкретної події аудиту. Після включення аудиту реєстрації в системі необхідно переглядати журнал подій безпеки системи, в якому відбиватимуться успішні і невдалі спроби.

2.5.5 Захист журналів подій

Windows має дуже потужні можливості протоколювання. На жаль, за умовчанням журнал подій не захищений від несанкціонованого доступу або зміни. Не дивлячись на те що події проглядаються за допомогою Event Viewer, журналом подій є звичайний файл, такий же, як інші. Для їх захисту досить лише знайти ці файли і застосувати до них відповідні списки ACL (Access Control List, визначає, хто або що може діставати доступ до конкретного об'єкту, і які саме операції дозволено або заборонено цьому суб'єктові проводити над об'єктом).

Якщо лише розташування файлів не було змінене за допомогою реєстру, то вони повинні знаходитися в каталозі %SystemRoot%\system32\config.[10]

З журналом застосунку, безпеки і системним протоколом зіставлені файли Appevent.Evt, Secevent.Evt і Sysevent.Evt відповідно. Для обмеження доступу до них лише за допомогою облікового запису адміністратора треба застосувати до них ACL. Це можна зробити за допомогою діалогового вікна властивостей файлу. Перейшовши на вкладку Безпека (Security), треба видалити на ній всіх користувачів і групи, за винятком Administrators і SYSTEM.

Результат виконання зазначених дій приведено на рисунку 2.6.

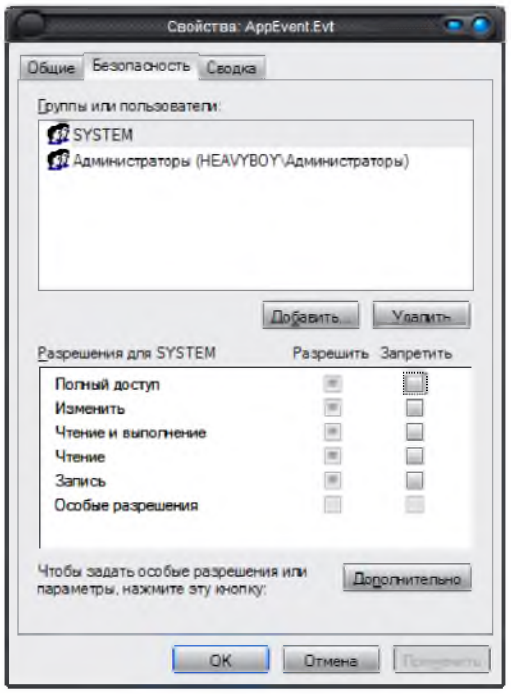

Рисунок 2.6 – Застосування переліків ACL до файлів журналів подій

2.5.6 Зміна максимальних розмірів файлів протоколів

З точки зору безпеки журнали – це найважливіший актив, що зберігається на сервері. Як-не-як, без протоколів ніколи не дізнатися, чи намагався хтось отримати доступ до даної машини. Отже, обов'язковою вимогою є повнота протоколювання. Намагатися виявити джерело інциденту за допомогою протоколу, в якому пропущені записи, все одно що взагалі не мати протоколу. Однією з проблем є те, що за умовчанням максимальний розмір файлу встановлений 512 Кбайт. Для його зміни досить відкрити на панелі управління

Засоби адміністрування (Administrative Tools) і відкрити вікно Перегляд подій (Event Viewer).

Після цього необхідно виділити в панелі, що знаходиться ліворуч цього вікна один з файлів протоколу і клацнути на ньому правою кнопкою миші. У контекстному меню вибрати пункт Властивості (Properties).

Далі треба знайти текстове поле Максимальний розмір протоколу (Maximum log size). Новий розмір можна задати вручну або за допомогою стрілок, розташованих праворуч від поля. Розмір має бути не менше 1 Мбайт і визначається тим, як часто доведеться переглядати і архівувати протоколи. Слід врахувати, що, хоча дуже великий розмір файлів протоколів ніяк не позначається на швидкодії машини, робота утиліти Перегляд подій (Event Viewer) з великими файлами буде сповільненою. Знаходячись на цій вкладці, можна також змінити поведінку протоколу при досягненні ним максимального розміру. За умовчанням записи журналу, зроблені більше семи днів тому, замінюються новими. Рекомендується збільшити це значення до 31 дня. Крім того, можна відключити автоматичний перезапис, але очищення протоколу доведеться виконувати вручну.

Результат виконання зазначених дій приведено на рисунку 2.7.

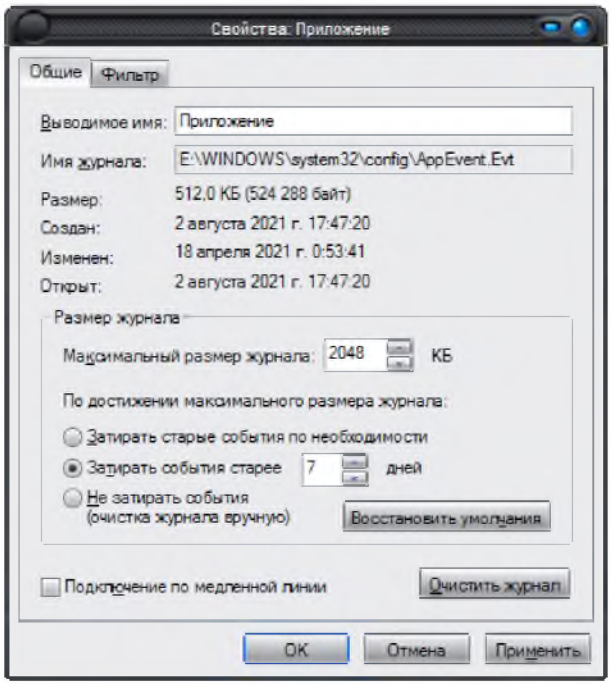

Рисунок 2.7 – Зміна максимального розміру файлів протоколів

2.5.7 Відключення стандартних загальних ресурсів

За умовчанням Windows дозволяє спільний доступ до кожного логічного диска (C$ для диска C:), а також відкриває інший загальний ресурс – Admin$ для каталога %SystemRoot% (тобто C:\winnt). Хоча вони доступні лише адміністраторам, рекомендується відключити ці загальні ресурси, оскільки вони є потенційною «дірою» в системі безпеки. Для відключення цих ресурсів потрібно відкрити реєстр, запустивши файл regedit.exe. Знайти в реєстрі ключ: Hkey_local_machine\system\currentcontrolset\services\lanmanserver\parameters.

Якщо робота здійснюється в ОС Windows – слід звернути увагу на параметр Description. Його наявність говорить про те, що загальний доступ дозволений. Відключення даного параметру заборонить загальний доступ. На рисунку 2.8 показано цей параметр у реєстрі.

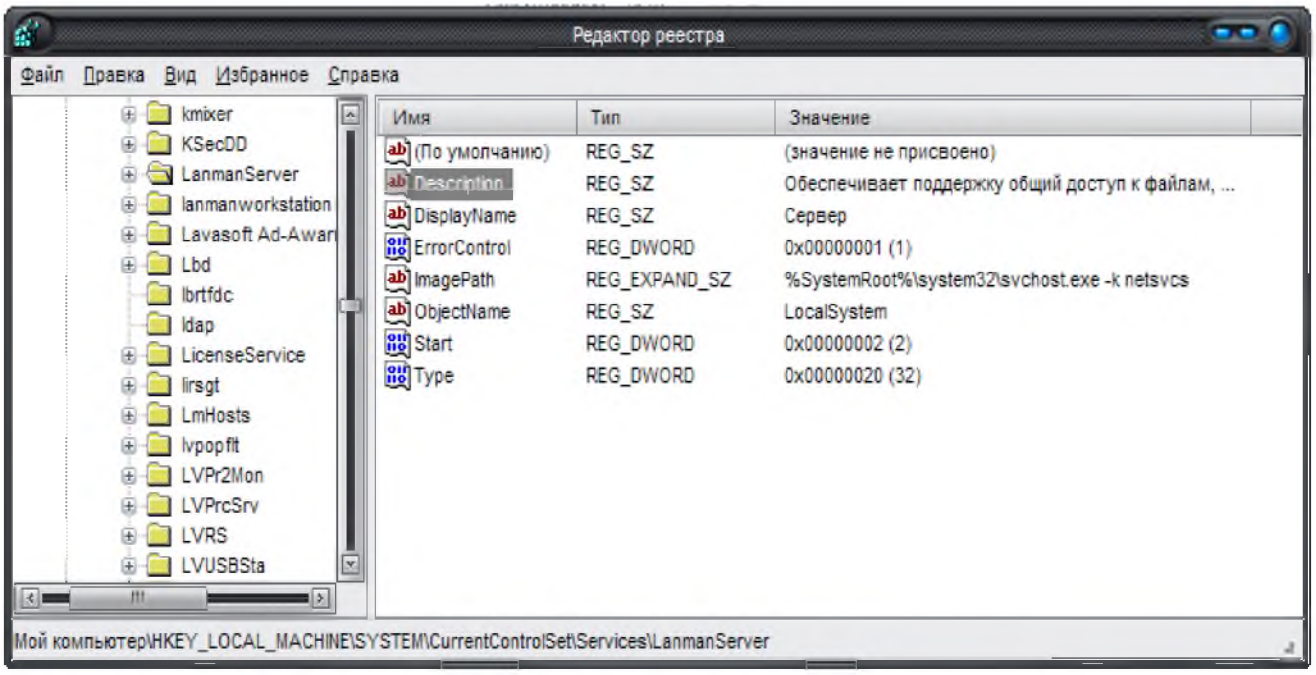

Рисунок 2.8 – відключення стандартних загальних ресурсів

Після завершення редагування реєстру потрібно перезавантажити Windows. Після перезавантаження Windows можна переконатися (за допомогою команди net share), що стандартних загальних ресурсів більше не існує. Перед цим необхідно переконатися, що відключення загальних ресурсів не позначиться негативно на оточенні користувача.

Відсутність цих ресурсів приведе до того, що деякі засоби управління системою, такі як Hfnetchk або System Management Server, працювати не будуть. Це відбувається через те, що подібне програмне забезпечення засноване на віддаленому доступі до відкритих адміністративних ресурсів, які використовуються для звернення до вмісту системних дисків.[11]

2.5.8 Шифрування папки Temp

Багато windows-застосунків в ході роботи створюють проміжні/тимчасові файли. Як правило, ці файли зберігаються в тимчасовому каталозі, розташованому в settings-каталозі поточного користувача. Майже завжди ці файли стають глобально доступними і не завжди видаляються після закінчення роботи застосунка. Одним із способів протистояти даній ситуації є шифрування каталога тимчасових файлів. Для цього потрібно відкрити вікно Провідника Windows й перейти в каталог C:\Documents and Settings\<ім'я користувача>\Local Settings. Тут можна знайти підкаталог з ім'ям Temp. Саме у ньому зберігаються тимчасові файли. Далі потрібно клацнути правою кнопкою на цьому підкаталозі і відкрити діалогове вікно властивостей. Перейти на вкладку Загальні (General) і клацнути на кнопці Додатково (Advanced). Це призведе до відкриття діалогового вікна Додаткові атрибути (Advanced Attributes). Тут можна задати шифрування каталога. Слід встановити прапорець Шифрувати вміст (Encrypt contents to secure data) і клацнути на кнопці ОК. Після цього клацнути на кнопці Застосувати (Apply). З'явиться ще одне діалогове вікно, що уточнює, чи потрібно застосувати рекурсивне шифрування. Для рекурсивного вживання шифрування вибрати параметр Застосувати зміни до каталога і його підкаталогів і файлів (Apply changes to this folder, subfolders and files). Результат виконання зазначених дій приведено на рисунку 2.9.

При цьому, якщо шифрування файлів ніколи раніше не виконувалося, автоматично буде створена пара публічних ключів. Інакше Windows використовуватиме створені раніше ключі.

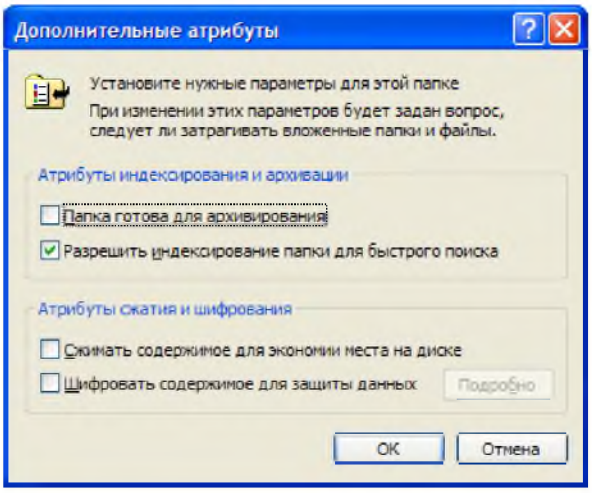

Рисунок 2.9 – Активація шифрування папки Temp

У ході декодування ОС Windows переконується, що приватні ключі зберігаються в пам'яті ядра, що не має посторінкової організації, так що ключ дешифрування ніколи не попаде у файл сторінкової пам'яті (файл підкачування). На жаль, використовуваний алгоритм шифрування (DESX) є лише злегка вдосконаленим DES і ніколи не буде настільки ж криптостійким, як 3DES. Проте для шифрування тимчасових файлів він підходить. [12]

2.5.9 Очищення файлу підкачування при відключенні

Система управління віртуальною пам'яттю (Virtual memory management, VMM) – корисна річ. Вона захищає програми одну від одної і дозволяє їм користуватися об'ємом пам'яті, який перевершує об'єм фізичний пам'яті, встановленій в системі. Для реалізації цього VMM використовує те, що називається сторінковим файлом (своп-файлом, файлом підкачування). У міру запуску все більшого числа програм об'єм фізичної пам'яті вичерпується. При цьому диспетчер пам'яті знаходить рідше використовувану ділянку пам'яті, що відноситься до програм, які в даний момент не проявляють жодної активності, і записує її на жорсткий диск (тобто у віртуальну пам'ять). Цей процес називається підкачуванням. Проте в цього процесу є один істотний недолік: якщо запущена програма, що обробляє конфіденційну інформацію в «своєму» просторі пам'яті, то ця інформація може виявитися на диску. Добре, якщо запущена операційна

система і існують засоби захисту, що запобігають можливості читання файлу підкачування, але як бути у випадку, якщо ця система відключена або виконується перезавантаження комп'ютера в іншу операційну систему? Ось де стає необхідним даний метод. Все, що необхідно зробити, – це попросити операційну систему при виключенні заповнити файл підкачування нулями. Слід врахувати, що це не спрацює, якщо відключення виробляється витяганням мережевого кабелю з розетки або система вимикається невірно; перезапис, про який йде мова, можливий лише при коректному вимиканні комп'ютера. Для включення цієї функції Windows необхідно внести зміни до системного реєстру. Для цього потрібно відкрити реєстр і знайти ключ: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SessionManager\MemoryManagement.

Далі треба знайти параметр Clearpagefileatshutdown в правій панелі і змінити його значення на 1. Результат виконання зазначених дій приведено на рисунку 2.10.

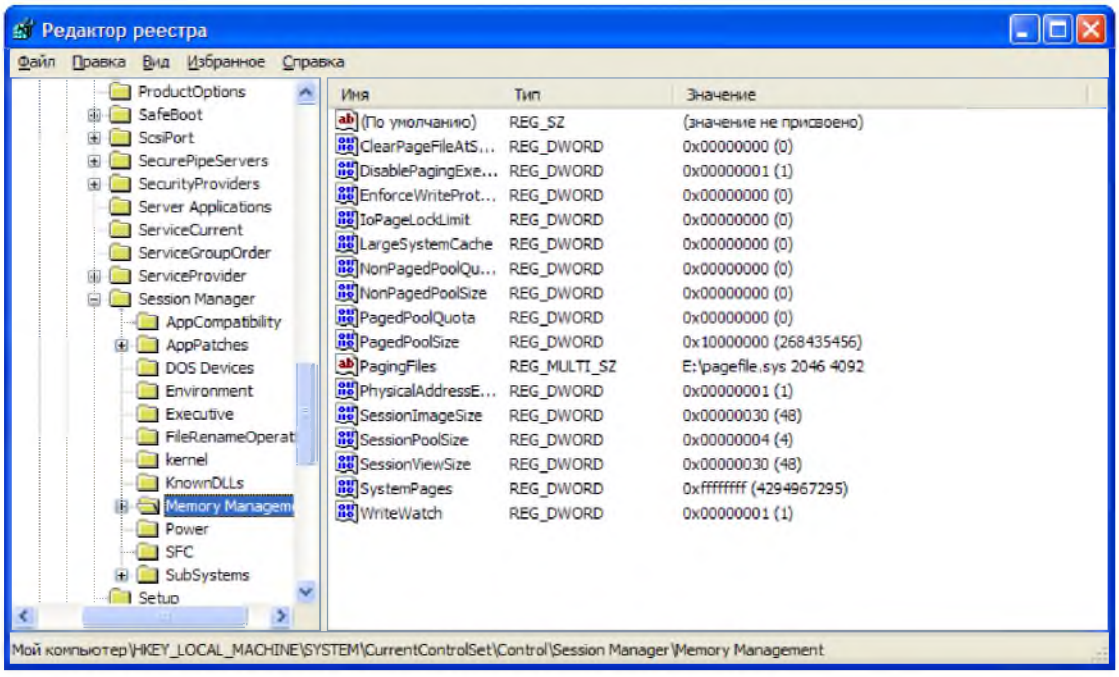

Рисунок 2.10 – Активація очищення файлу підкачування

Для того, щоб зміна набрала чинності, перезавантажити Windows, і тепер перед вимиканням комп'ютера своп-файл буде очищуватися. Єдиний побічний

ефект від використання цієї функції полягає в тому, що вимикання Windows тепер займатиме більше часу. Проте це дуже сильно залежить від апаратного забезпечення (тобто від чіпсета контроллера диска, швидкості обертання диска, швидкості процесора і т.п.), що визначає тривалість запису нулів у файл підкачування.[13]

2.5.10 Обмеження доступу користувача до застосунків

Забезпечення неможливості для користувачів запускати певні застосунки не дуже важливо при адмініструванні власної робочої станції. Але коли доводиться мати справу із звичайними користувачами, що працюють в мережевому середовищі підприємства, необхідно захистити їх від запуску «шкідливих» програм. До таких програм відносяться застосунки, які можуть порушити роботу операційної системи, створюють «діри» в системі безпеки і навіть організовують атаки на інші комп'ютери мережі. Існує декілька способів обмеження доступу користувачів до застосунків. По-перше, можна змінити ACL для конкретної програми таким чином, що користувачі не зможуть запустити її. Наприклад, на комп'ютері користувача для проведення діагностики встановлена програма, що є мережевим аналізатором (sniffer). Ця програма корисна для адміністратора, а звичайному користувачеві вона не потрібна. Захистити його від запуску цієї програми можна видаленням дозволу на її виконання для групи Users. Для цього потрібно знайти виконуваний файл програми і клацнути на ньому правою кнопкою. У контекстному меню вибрати пункт Властивості (Properties). Тепер перейти на вкладку Безпека (Security) і в списку, розташованому у верхній частині вкладки, вибрати групу Users. Встановити прапорець Відмовити (Deny), що відноситься до пункту Запис і виконання (Read & Execute). Після клацання на кнопці Застосувати (Apply) жоден член групи Users не зможе запустити цю програму. По-друге, можна також змінити ACL каталога, в якому знаходиться програма, позбавивши можливості читання. Такий підхід може виявитися корисним; якщо всі засоби адміністрування знаходяться в

одному каталозі, він дозволяє обмежити доступ до них за допомогою всього одній операції. Результат виконання зазначених дій приведено на рисунку 2.11.

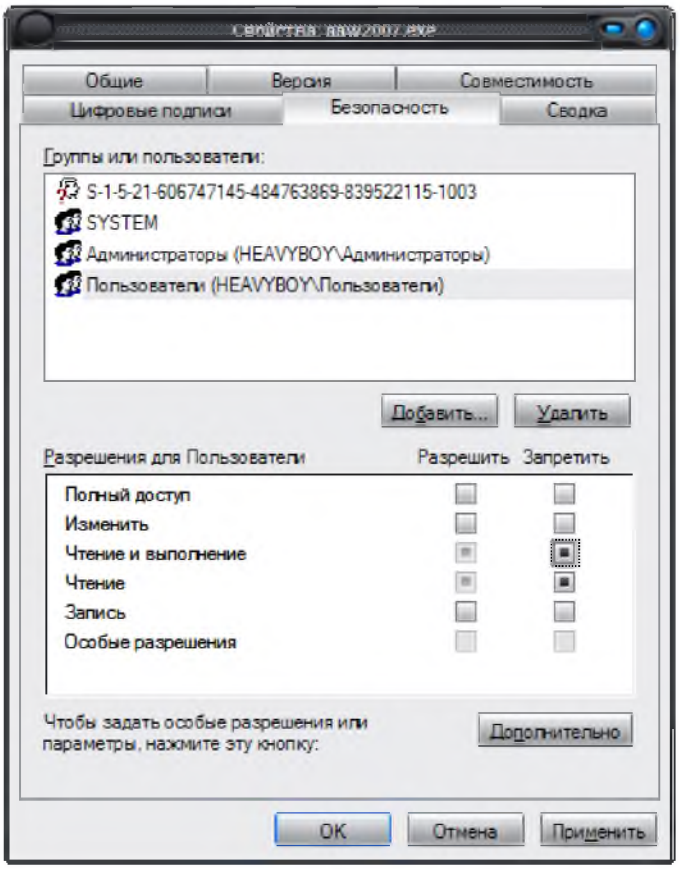

Рисунок 2.11 – Обмеження доступу користувачів до застосунків

2.6 Організація заходів щодо забезпечення мережевої безпеки ИКС підприємства

В міру того, як людство все більше покладається на зв'язок за допомогою масових мереж, усе більш важливими стають стабільність і надійність цих мереж. Світ бізнесу схвалив інформаційні технології, що допомагають упорядкувати процеси, підвищити їх продуктивність і знизити вартість. Таким чином, IT-підрозділ є ключовою ланкою багатьох компаній. З цієї причини в разі лиха (природного або цифрового), яке дуже часто перериває виконання мережевих операцій, багато підприємств просто припиняють свою діяльність.

2.6.1 Виявлення ARP-спуфінгу

Однією з найбільших загроз комп'ютерним мережам є шахрайські системи, що маскуються під вузол, якому довіряють. Хто-небудь, що успішно

представився іншим вузлом, може зробити багато неприємних речей. Він може перехопити і протоколювати трафік, що направляється іншому вузлу, або обдурити клієнтів, що підключилися, і перехопити конфіденційну інформацію. Спуфінг (діставання доступу обманним шляхом (ситуація, коли користувач намагається з'єднатися з сервером, proxy-сервером або брандмауером, використовуючи чужу ip-адресу)) вузла завдає особливо серйозного збитку ip-мережам, оскільки відкриває широкі можливості для атак. Однією з технологій спуфінга вузла в ip-мережі є спуфіпг протоколу дозволу адрес (Address Resolution Protocol, ARP). ARP-спуфінг обмежений локальним сегментом мережі і діє за рахунок використання способу, за допомогою якого ip-адреси транслюються в апаратні ethernet-адреси. При відправленні ip-пакету від одного вузла до іншого в одному і тому ж фізичному сегменті ip-адреса призначення повинна транслюватися в MAC-адресу. MAC-адреса – це апаратна адреса ethernet-карти, включеної в мережу. Для цієї трансляції використовується протокол дозволу адрес. Коли вузлу потрібно дизнатися ethernet-адресу іншого вузла, він посилає широкомовну структуру, яка виглядає приблизно так:

01:20:14.833350 arp who-has 192.168.0.66 tell 192.168.0.62

Ця структура називається ARP-запитом. При відправці його на широкомовну адресу цей запит повинні побачити всі ethernet-пристрої локального сегменту. Машина, яка задовольняє умовам запиту, відповідає:

01:20:14.833421 arp reply 192.168.0.66 is-at 0:0:dl:If:3f:fl

Оскільки ARP-запит вже містить MAC-адресу відправника, одержувач може відправити відгук без виконання ще одного ARP-запиту. На жаль, великим недоліком ARP є те, що цей протокол не перевіряє стан (stateless). Це означає, що він не відстежує відгуки на послані запити і може приймати відповіді навіть без відправки запиту. Якщо хто-небудь захоче приймати трафік, призначений іншому вузлу, він може посилати фальшиві ARP-відгуки, що визначають відповідність вибраної ip-адреси їх власній MAC-адресі. Машини, що отримали помилкові arp-відповіді, не можуть відрізнити їх від достовірних і починають відправляти пакети на MAC-адресу зловмисника.

Ще один побічний ефект такої поведінки ARP -протоколу полягає в тому, що ARP-табліці зазвичай використовують результати виконання лише останнього запиту. Для того, щоб продовжувати використовувати невірні ip-адреси, зловмисникові досить заповнити вузол ARP-відповідями, що перезаписують законні ARP-відповіді вихідного вузла. Такий тип атаки називається спотворенням ARP-кеша. Деякі засоби, такі як Ettercap, Dsniff і Hunt, застосовують подібні технології як для «нишпорення» в комутованих мережах, так і для атаки «посередника». Звичайно ж, ця технологія може бути застосована до будь-яких двох вузлів комутованого сегменту, включаючи локальний шлюз за умовчанням. Для двонаправленого перехоплення трафіку між вузлами А і В атакуючий вузол С може спотворити ARP-кеш вузла А, зробивши так, щоб він думав, що ip-адреса вузла В відповідає MAC-адресі вузла С. После цього він спотворює arp-кеш вузла В, щоб той вважав, що ip-адреса вузла А відповідає MAC-адресі вузла С. На щастя, існують методи виявлення саме такої поведінки як в спільно використовуваному, так і в комутованому ethernet-сегменті. Одна з програм, яка допоможе зробити це, – Arpwatch. Вона виконує моніторинг інтерфейсу в довільному режимі і періодично записує Mac/ip-пари. При виявленні аномальної поведінки, наприклад зміни однієї з відомих Mac/ip-пар, програма відправляє в системний протокол сигнал тривоги. Ця програма дуже ефективна в спільно використовуваній мережі, що застосовує концентратор, оскільки одна машина може переглядати весь ARP-трафік. Проте в комутованих мережах ця програма не працює унаслідок однонаправленості ARP-відповіді. Для досягнення високого рівня виявлення в комутованому середовищі Arpwatch повинна встановлюватися на максимальну кількість машин. І все-таки не можна зі стопроцентною достовірністю сказати, який з вузлів зловмисник вибрав як свою ціль. Багато сучасних комутаторів дозволяють визначити «спостерігаючий» порт, який може бачити трафік всіх останніх портів. Якщо встановлений такий комутатор, то можна встановити сервер на цей порт і просто запустити на ньому Arpwatch. Після завантаження Arpwatch з Інтернету його можна відкомпілювати і встановити звичайним способом, запустивши:

./configure && make && make install

При запуску Arpwatch на машині з безліччю інтерфейсів (мережевих адаптерів) необхідний інтерфейс можна вказати в командному рядку. Це робиться за допомогою параметра -i:

arpwatch -i iface

Як тільки Arpwatch почне вивчати Mac/ip-пари мережі, в протоколі почнуть з'являтися записи, подібні цьому:

Nov 1 00:39:08 zul arpwatch: new station'192.168.1.5 0:50:ba:85:85:ca

При зміні Mac/ip-пари там буде видно наступне:

Nov 1 01:03:23 zul arpwatch: changed ethernet address 192.168.1. 5 0:e0:81:3:d8:8e (0:50:ba:85:85:ca)

Nov 1 01:03:23 zul arpwatch: flip flop 192.168.1. 5 0:50:ba:85:85:ca (0:e0:81:3:d8:8e)

Nov 1 01:03:25 zul arpwatch: flip flop 192.168.1.5 0:e0:81:3:d8:8e (0:50:ba:85:85:ca)

В даному випадку перший запис – від першої прийнятої помилкової ARP-відповіді, а два подальших записи виникли із-за стану перегонів між помилковою і правильною відповідями. Для полегшення роботи з декількома Arpwatch в комутованому середовищі можна посилати повідомлення протоколювання в центральний системний журнал збираючи всі результати в одному місці. Проте через те, що комп'ютери можуть знаходитися під впливом атак, подібних тій, що виявляє Arpwatch, на сервері з системним журналом, а також на всіх вузлах, на яких встановлена Arpwatch, розумно використовувати статичні записи ARP-таблиць.

На рисунку 2.12 відображено результат перевірки правильності налаштування програми Arpwatch. Виведені дані вказують на правильність налаштування.

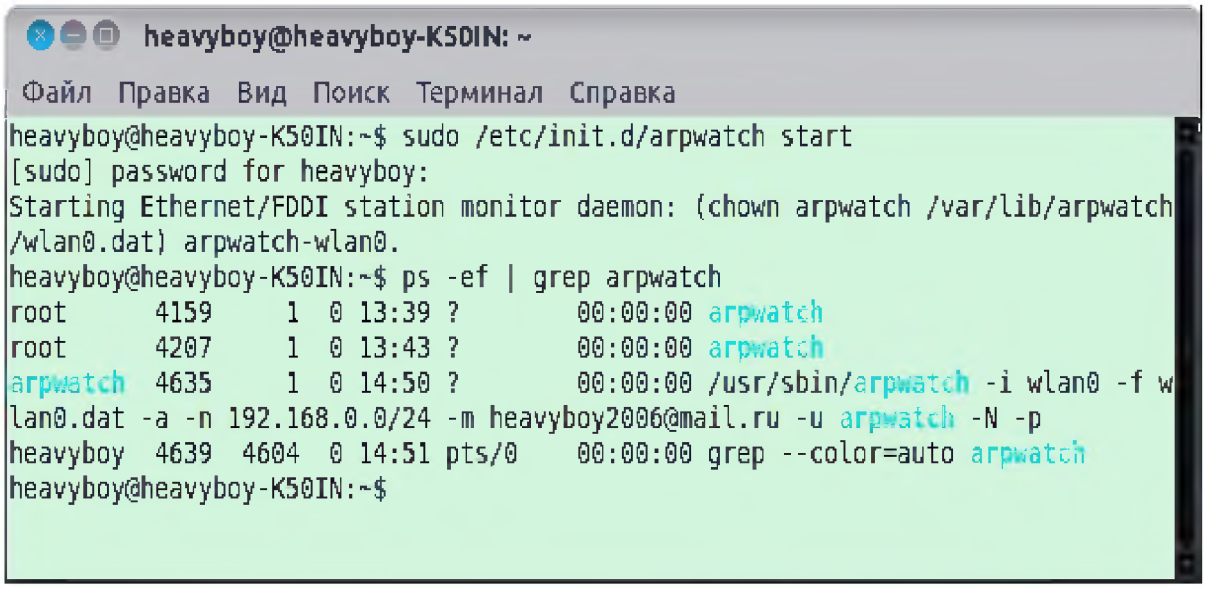

Рисунок 2.12 – Підтвердження правильного налаштування Arpwatch та його готовності до виявлення ARP-спуфінгу

2.6.2 Створення статичної ARP-таблиці

Одним з методів запобігти появі неприємних наслідків вторгнення в мережу є створення статичних записів ARP-табліць для всіх пристроїв локального сегменту мережі (у даному випадку – для сервера). Після цього ядро ігноруватиме ARP-відповіді з будь-якої ip-адреси і використовуватиме лише вказані MAC-адреси. Для цього можна скористатися командою arp, що дозволяє безпосередньо маніпулювати записами ARP-табліць ядра. Один статичний запис додається командою:

arp -s ip-адрес mac-адрес

Якщо відомо, що ip-адресі 192.168.1.7 відповідає MAC-адреса 00:50:BA:85:85:CA, то команда повинна виглядати таким чином:

arp -s 192.168.1.7 00:50:Ba:85:85:CA

Процес виконання безлічі записів може зайняти багато часу. Для більшої ефективності необхідно додати запис для кожного пристрою мережі на кожен вузол, що дозволяє створювати arp-табліці. На щастя, більшість версій команди arp як вхідні параметри для створення статичних записів можуть використовувати файл. У Linux це реалізується за допомогою ключа -f.

Необхідно лише згенерувати файл, що містить пари MAC- та ip-адрес, який потім можна скопіювати на всі вузли мережі. Для спрощення цієї операції можна використовувати сценарій на мові Perl:

```
#!/usr/bin/perl
# gen_ethers.pl <from ip> <to ip>
my ($start_1. $start_2. $start_3. $start_4) = split(/\./. $ARGV[0]. 4);
my ($end_1. $end _2. $end _3. $end _4) = split(/\./. $ARGV[1]. 4);
my $ARP_CMD=”/sbin/arp –n”;
for(my $oct_1 = $start_1; $oct_1 <= $end_1 && $oct_1 <= 255; $oct_1++){
  for(my $oct_2 = $start_1; $oct_2 <= $end_1 && $oct_2 <= 255; $oct_2++){
    for(my $oct_3 = $start_1; $oct_3 <= $end_1 && $oct_3 <= 255; $oct_3++){
      for(my $oct_4 = $start_1; $oct_4 <= $end_1 && $oct_4 <= 255; $oct_4++){
    system(“ping –c 1 –W 1 $oct_1.$oct_2.$oct_3.$oct_4 > /dev/null 2>&1”);
        my $ether_addr = ‘$ARP_CMD $oct_1.$oct_2.$oct_3.$oct_4 | egrep
         ‘HWaddress | (incomplete)’ | awk ‘{print \$3}’’;
    chomp($ether_addr);
    if(length($ether_addr) == 17){
      print(“$ether_addr\t$oct_1.$oct_2.$oct_3.$oct_4\n”);}}}}}
```

Цей сценарій по черзі «продзвонює» (ping) ip-адреси з певного діапазону. В результаті в ARP-таблиці машини буде представлена кожна активна ip-адреса. Після «продзвонювання» ip-адрес сценарій переглядає ARP-табліцю і роздруковує пари Mac/ip-адрес у форматі, зручному для переміщення їх у файл.

Цей сценарій написаний для Linux, але працюватиме і в інших unix-подібніх ОС. Якщо необхідно згенерувати файл для всіх ip-адрес від 192.168.1.1 до 192.168.1.32 і зберегти результати в /etc/ethers, то необхідно виконати наступний сценарій:

```
# ./gen_ethers 192.168.1.1 192.168.1.32 > /etc/ethers
```

При використанні arp з ключем -f для створення статичних записів автоматично використовується файл /etc/ethers, проте можна вказати і будь-який інший. Для збереження записів в /root/arp_entries команда виглядатиме так:

arp -f /root/arp_entries

Цей сценарій не є ідеальним, але він може заощадити багато часу. Після того, як створений файл з парами Mac/ip-адрес, його можна скопіювати на інші вузли і додати команду arp в сценарій завантаження системи для його автоматичного виконання при завантаженні. Основний недолік цього методу полягає в тому, що на момент виконання сценарію всі пристрої мережі вже мають бути включені, інакше вони не будуть внесені до списку. Крім того, при частій заміні машин в мережі доведеться переробляти і заново поширювати файл, що може зайняти більше часу, чим дозволяє виграти сам метод. Але цей спосіб надійно захищає від атак із спотворенням ARP сервери та пристрої, які не змінюють свої IP- і MAC-адреси.[14]

На рисунку 2.13 наведено схему реалізації подвійного захисту від ARP-спуфінгу.

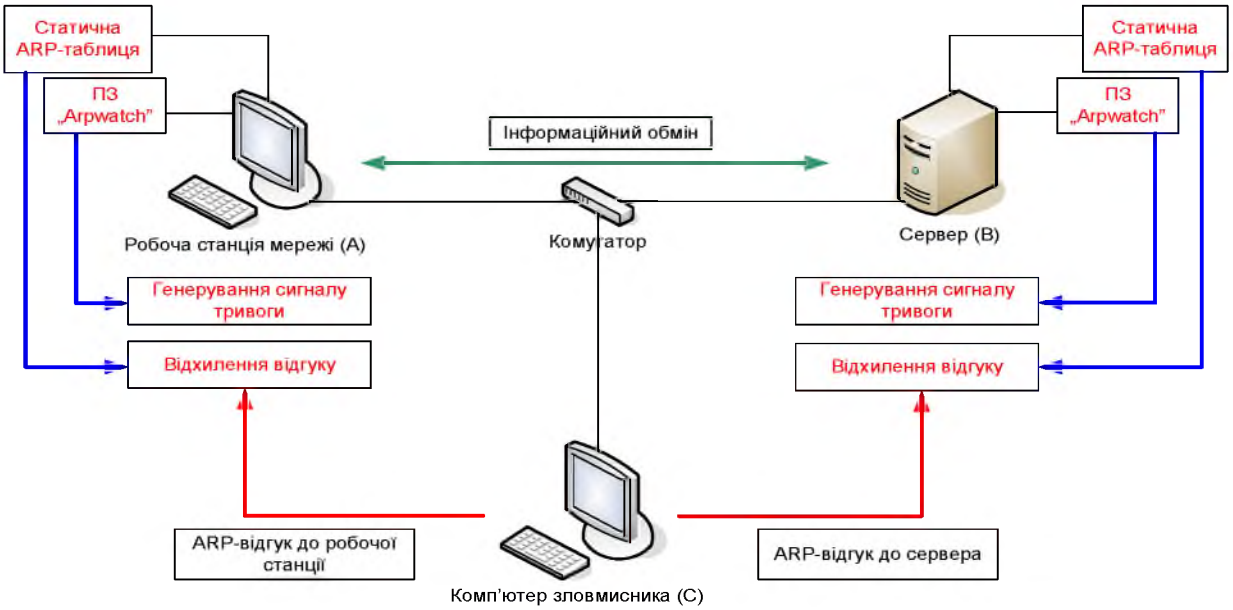

Рисунок 2.13 – Схема реалізації захисту від ARP-спуфінгу

2.7 Організація заходів щодо підвищення захищеності мережевих каналів зв’язку ІС підприємтсва

Небезпечні комп'ютерні мережі є ворожим середовищем, але їх можна «приборкати». Балансуючи між шифруванням і деякими прийомами інкапсуляції, можна побудувати надійніші мережі, незалежно від

використовуваної глобальної мережі, навіть якщо вона повна негідників, що намагаються поглянути або по-іншому використовувати особисті дані.

Шифрування і створення каналу для трафіку за допомогою SSL

Stunnel – це потужна і зручна програма, яка за допомогою SSL (криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером) шифрує трафік будь-якого TCP-порту. Вона утворює канал, практично так само, як SSH (мережевий протокол сеансового рівня, що дозволяє робити віддалене управління операційною системою і туннелювання TCP-з'єднань), надаючи для цього локальний порт. Програма шифрує трафік, що посилається на цей порт, направляє його віддаленій системі, дешифрує трафік і направляє його на локальний порт системи. Stunnel також може забезпечити явну SSL-підтрімку для inetd-сумісніх служб. Для установки stunnel просто потрібно виконати команду ./configure з каталога, який створений при розпаковуванні завантаженого з мережі архіву. Оскільки stunnel для роботи потрібний OPENSSL (криптографічний пакет з відкритим початковим кодом для роботи з SSL/TLS. Дозволяє створювати ключі RSA, DH, DSA і сертифікати X.509, підписувати їх. Також є можливість шифрування даних і тестування SSL/TLS з'єднань), то спочатку треба завантажити і встановити OPENSSL. Якщо є бажання відкомпілювати stunnel з підтримкою TCP-оболонок або встановити OPENSSL в нестандартне місце, то необхідно скористатися параметром --with-tcp-wrappers або --with-ssl. Для налаштування stunnel з підтримкою TCP-оболонок при установці OPENSSL в /opt/ потрібно виконати:

$ ./configure --with-tcp-wrappers --with-ssl=/opt/openssl

Після запуску цього сценарію для дійсної компіляції stunnel необхідно виконати команду make. Після цього буде запропоновано створити самопідписаний сертифікат. Цей сертифікат буде не лише самопідписаним, але і термін його дії буде обмежений одним роком. Якщо це не влаштовує, можна створити власний сертифікат. До версії 3.x програми stunnel була можливість налагтовувати всі параметри з командного рядка. У версіях 4.x і вище використовується файл налаштування stunnel.conf. Приклад цього файлу

зазвичай можна знайти в /etc/stunnel/stunnel.conf-sample або /usr/local/etc/stunnel/stunnel.conf-sample.

На рисунку 2.14 показана схема функціонування програми Stunnel в локальній обчислювальній мережі.

Далі буде розглянута базова форма файлу налаштування, вказуючого перенаправлення локального порту на віддалений порт.

Клієнтська сторона:

client = yes

[<server port>]

accept = <forwarded port>

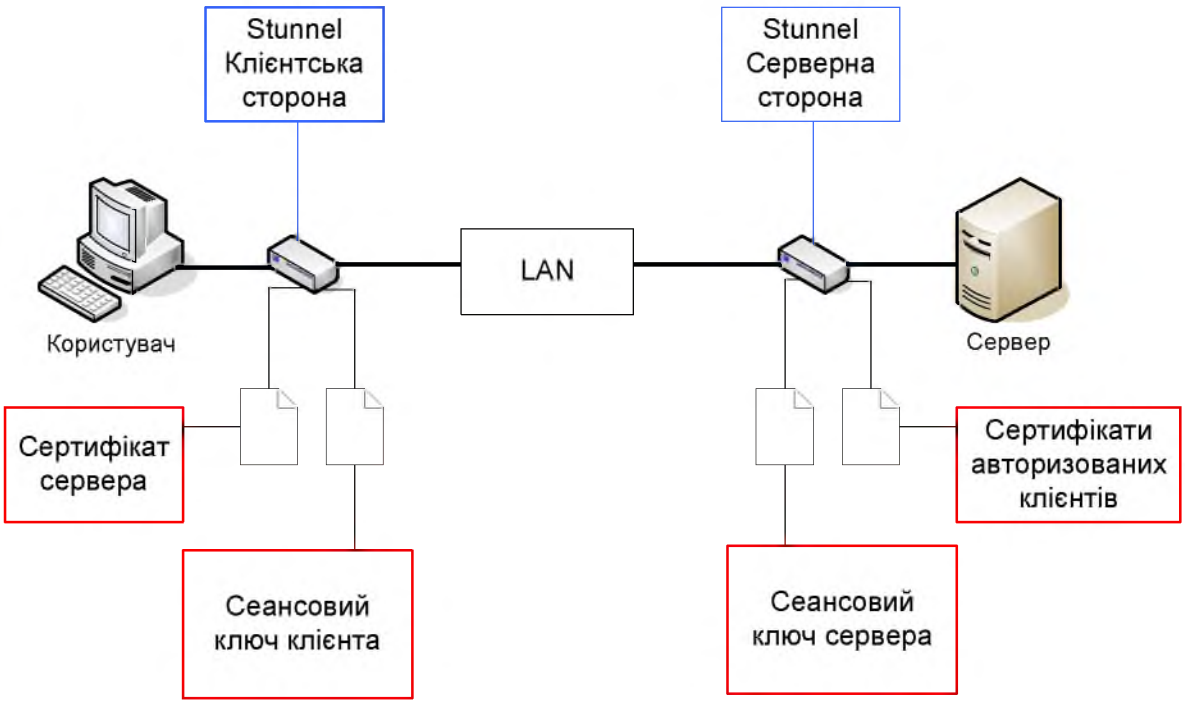

Рисунок 2.14 – Схема функціонування програми Stunnel

connect = <remote address>:<server port>

Серверна сторона:

cert = /etc/stunnel/stunnel.pem

pid -

client = no

[<forwarded port>]

accept = <server port>

connect = <forwarded port>

Можна використовувати стандартний файл налаштування або вибрати інший файл. У першому випадку stunnel запускається без всяких аргументів. Інакше слід вказати необхідний файл налаштування як перший аргумент. При налаштуванні, що є зараз, програма здатна підключитися до <Forwarded port> (порту, що перенаправляється) на клієнтській стороні. Після цього stunnel зашифрує трафік, що отримується з цього порту і відправить його на <Server port> (серверний порт) віддаленої сторони, вказаної в <remote address> (адреса віддаленої сторони). На віддаленій системі stunnel дешифрує отриманий на вказаний порт трафік і направить його програмі, яка «слухає» <forwarded port> на віддаленій системі. Еквівалентна команда перенаправлення портів в SSH могла б виглядати так:

ssh -f -N -L <forwarded port>:<remote address>:<forwarded port> \<remote address>

Якщо є бажання вказати PID файлу, можна налаштувати змінну pid для будь-якого файлу. Проте якщо повністю пропустити змінну pid, то stunnel спробує на основі налаштувань, що використалися при компіляції, створити або /var/run/stunnel.pid, або /usr/local/var/run/stunnel.pid (тобто $prefix/var/run/stunnel.pid). Окрім надання ssh-подібного перенаправлення портів, stunnel також може використовуватися для додавання ssl-можлівостей в inetd-подібні служби. Це ідеально личить для впровадження SSL в електронну пошту або інші сервіси, що не володіють «рідною» ssl-функціональністю. [15]

Також це ПЗ використовується для створення VPN-каналу, необхідного для забезпечення конфіденційності та цілісності інформації з обмеженим доступом, що передається до мережі керівництва, що розташована у Київі (принцип дії той самий).

Stunnel – дуже потужний засіб: він може не лише перенаправляти з'єднання по зашифрованому каналу, але і використовуватися для додавання SSL-можлівостей в поширені служби. Це особливо зручно, коли вже існують клієнти

з SSL-підтрімкою. Таким чином, на серверній стороні досить використовувати лише stunnel, який забезпечує шифрування служби без необхідності інсталяції додаткового програмного забезпечення.

2.8 Планування та впровадження системи виявлення вторгнень в ІС підприємства

Одними з типів засобів, що висуваються останніми роками на передній план, є системи виявлення мережевого вторгнення (network intrusion detection systems, NIDS). Ці системи розміщуються в мережі і спостерігають за трафіком до тих пір, поки не виявлять підозрілу поведінку, після чого виявляються і повідомляють про це.

NIDS є відмінним засобом для спільної роботи з журналами (протоколами), оскільки часто можуть виявити атаку до того, як вона досягне передбачуваної мети і відіб'ється в протоколах. На даний момент існує два основні типи NIDS. До першого типа відносяться системи, здатні виявляти вторгнення по наявності в трафіку певних байтових штампів, характерних для відомих атак. NIDS, що працюють за цим принципом, відомі як системи виявлення вторгнення на основі підпису. Другим типом є системи статистичного моніторингу. Системи цього типу спостерігають за трафіком, але замість пошуку певного штампу, або «підпису», вони нагромаджують статистичну хронологію пакетів, що передаються по мережі і сповіщають про появу пакетів, що «випадають» з узагальненого шаблону. NIDS, що використовують цей метод, називаються системами виявлення вторгнення на основі аномалій.

2.8.1 Виявлення вторгнення за допомогою Snort

Безперечним лідером серед систем виявлення з відкритим вихідним кодом є Snort. Настільки потужною Snort робить можливість розширення за допомогою надбудов і препроцесорів. Вони дозволяють розширювати Snort в будь-якому напрямі. Отже, користувач більше ні від кого не залежить в плані визначення правив, що запобігають новим зломам. Маючи базове уявлення про TCP/IP, він

може швидко і легко написати власні правила. Це, мабуть, найважливіша особливість Snort, оскільки нові атаки винаходяться постійно.

Крім того, Snort має дуже зручний механізм звітів, що дозволяє відправляти сигнали тривоги в syslogd, в звичайні файли і навіть в бази даних. Для компіляції і установки Snort потрібно завантажити з мережі останню версію і розпакувати її. Потім запустити сценарій налаштування, а потім команду make:

$ ./configure && make

Після цього отримати повноваження root і виконати:

make install

Слід звернути увагу на те, що всі заголовні файли і бібліотеки для libpcap мають бути встановлені до компоновки Snort, інакше компіляція завершиться невдачею. Крім того, для вказівки компілятору місцезнаходження бібліотек і заголовних файлів необхідно використовувати параметри --with-libpcap-includes і –with libpcap-libraries. Проте їх можна застосовувати лише в разі нестандартної установки бібліотек (не у підкаталоги каталога /usr або /usr/local). Якщо libpcap встановлена в /opt, то необхідно використовувати наступну команду:

$ ./configure --with-1ibpcap-includes=/opt/include\

--with-1ibpcap-1ibraries=/opt/1ib

Snort має можливість відповідати вузлу, який запустив (активізував) одне з правил. Ця можливість називається гнучким відгуком. Щоб включити її, необхідно використовувати параметр --enable-flexresp, для роботи якого потрібна бібліотека інжекції пакетів (packet injection library) libnet. Переконавшись в тому, що цей пакет встановлений, для вказівки його місця розташування потрібно скористатися ключами --with-1 ibnet-includes і --with-libnet-libraries. Якщо необхідно додати можливість відправки попереджень в БД, необхідно скористатися ключами --with-mysql, --with-postgresql або --with-oracle. Для перегляду повного списку параметрів налаштування ввести ./configure --help. Після установки перевірити роботу Snort в режимі аналізу
(sniffer mode):

./snort -evi ethO

На рисунку 2.15 приведено результат правильного налаштування системи виявлення вторгнень Snort.

У вихідному дистрибутиві Snort в каталозі etc/ є декілька конфігураційних (надбудовних) файлів, але при виконанні make install вони не встановлюються.

Рисунок 2.15 – Підтвердження правильної установки та налаштування Snort

Для їх розміщення необхідно створити підкаталог в /etc або /usr/local/etc і скопіювати відповідні файли подібною командою:

```
# mkdir /usr/local/etc/snort &&\
cp etc/[^Makefile]* /usr/local/etc/snort
```

Ймовірно, туди ж з'явиться бажання скопіювати і каталог rules. Тепер необхідно відредагувати файл snort.conf. Цей файл містить список змінних. Деякі

з них мають значення за умовчанням, і в кожної змінної є коментар, що пояснює її призначення. Ось, зокрема, дві змінні:

var HOME_NET any

var EXTERNAL_NET any

Інші змінні, що відносяться до ip-адрес або мережевих діапазонів (DNS_servers, SMTP_servers, HTTP_servers, SQL_servers і Telnet_servers), за умовчанням мають значення $HOME_NET. Ці змінні використовуються в наборі правив, який є в дистрибутиві Snort, і можуть застосовуватися для «тонкого» налаштування поведінки правил. Наприклад, правила, що відносяться до SMTP-атак (SMTP - це мережевий протокол, призначений для передачі електронної пошти в мережах TCP/IP), використовують змінну SMTP_servers для фільтрації трафіку, який фактично не відноситься до цього правила. Налаштування цих змінних не лише приводить до точнішої генерації сигналів тривоги і зменшення помилкових спрацьовувань, але і підвищує продуктивність. Ще однією важливою змінною є Rule_path, яка використовується в конфігураційному файлі для додавання наборів правил. Приклад конфігураційного файлу встановлює для неї значення ../, але для сумісності з попередніми прикладами вона повинна мати значення ./rules, оскільки snort.conf і каталог rules знаходяться в каталозі /usr/local/etc/snort.

Наступний розділ конфігураційного файлу дозволяє набудовувати вбудовані препроцесори Snort. Вони здатні робити все: від збірки (відновлення) фрагментованих пакетів до декодування http-трафіку з метою виявлення сканування портів. В більшості випадків досить налаштування за умовчанням. Проте якщо необхідно змінити будь-яке з налаштувань, то всі параметри препроцесорів можна знайти в описі конфігураційного файлу. Якщо при компіляції була задана підтримка баз даних, то необхідно дозволити роботу надбудови виводу в БД (database output), що дозволяє Snort зберігати будь-які сигнали, що генеруються, в БД. Включається вона в конфігураційному файлі подібним рядком:

output database: log. mysql, user=snort password=snortpass dbname=SNORT \

host=dbserver

output database: alert mysql, user=snort password=snortpass dbname=SNORT\

host=dbserver

Перший рядок набудовує в Snort відправку в БД відомостей, що генеруються правилами, що визначають дії протоколювання (log action). Другий рядок відправляє до БД відомості, що генеруються правилами, що визначають дії з тривоги (alert action). Якщо є бажання використовувати Snort спільно з БД, то необхідно створити нову БД і, можливо, новий обліковий запис користувача цієї БД. У каталозі contrib первинного коду Snort є сценарії створення баз даних підтримуваних типів: create_mssql, createjnysql, create_oracle.sql і create_postgresql. При використанні MYSQL БД з необхідними таблицями створюється наступною командою:

mysql SNORT -p < ./contrib/createjnysql

Вся інша частина конфігураційного файлу в основному присвячена правилам «підписів», використовуваних Snort при спостереженні за трафіком. Ці правила розділені на категорії і зберігаються в окремих файлах, а активізуються за допомогою директиви include. Для перевірки (або в мережі з ненапруженим трафіком) достатньо налаштувань за умовчанням, але бажано проглянути ці правила і вирішити, які категорії дійсно потрібні, а які – ні. Тепер, коли вся складна робота по налаштуванню виконана, треба перевірити файл snort.conf подібною командою:

snort -T -c /usr/local/etc/snort/snort.conf

В результаті її виконання буде виведений звіт про всі виявлені помилки. За відсутності помилок можна запустити Snort командою:

snort -Dd -z est -c /usr/local/etc/snort/snort.conf

Два прапори (-d і -c) вже використовувалися раніше (для декодування пакетів і використання вказаного конфігураційного файлу відповідно), а два інших – нові. Прапор -D заставляє Snort вивести повідомлення про запуск, а потім перейти у фоновий режим. Аргумент -z est примушує надбудову препроцесора streams ігнорувати TCP-пакети, що не відносяться до встановлених

сеансів, що зменшує чутливість системи до спуф-атак і, звичайно ж, до dos-атак. Корисними є також параметри -u і –g, що дозволяють Snort скидати свої привілеї і запускатися вказаним користувачем або групою. Це особливо корисно при спільному використанні з параметром -t, який за допомогою виклику chroot() помістить Snort у вказаний користувачем каталог. Тепер можна переходити до перегляду протоколів, що з'являються в /var/log/snort.[16]

2.8.2 Відстеження сигналів тривоги

Після налаштування Snort на протоколювання відомостей у БД можна дійти висновку, що користувачеві складно впоратися з обробкою усіх генерованих даних. Дуже завантажені і привертаючі увагу сайти можуть генерувати величезну кількість попереджень, які часто необхідно відкидати. Одним із способів полегшення цієї роботи є установка ACID. ACID (скорочення від Analysis Console for Intrusion Databases (консоль аналізу БД вторгнень)) - це web-інтейфейс для БД, що містить сигнали тривоги, які поступили від систем виявлення вторгнень. Він має можливість виконувати пошук сигналів за певним критерієм (по підпису, часу виявлення, за адресою і портом, а також по контексту "корисного навантаження" пакету або по значеннях прапорів). ACID може виводити пакети, які призвели до генерації сигналу тривоги, а також декодувати в них відомості 3-го і 4-го рівня. ACID також має функції управління сигналами тривоги, що дозволяють групувати сигнали по спрямованості, видаляти вивчені або неправдиві сигнали, відправляти сигнали по електронній пошті або архівувати їх в іншій БД. ACID також забезпечує різну статистику сигналів тривоги : хронологічну, по датчику, який їх згенерував, по підпису. Крім того, надається статистика пакетів (протокол, адреса або порт).

Для установки ACID передусім знадобиться web-сервер і працююча інсталяція PHP (Apache і mod_php), а також інсталяція Snort, налаштована на підключення до БД (MySQL). Потрібні будуть також декілька PHP-бібліотек: для абстракцій БД - ADODB, а для створення графіки - або PHPIot, або JPGraph. Після завантаження з мережі цих пакетів треба розпакувати їх в каталог, який

використовуватиметься для виконання PHP-контекста на web-сервері. Далі треба перейти в каталог, який був створений при розпаковуванні ACID (тобто в ./acid), і відредагувати файл acid_conf.php. У ньому необхідно вказати, де знайти ADODB і JPGraph, а також як підключитися до БД Snort :

$Dblib_path = \ ./adodb";

$Dbtype = "mysql":

$alert_dbname - "SNORT":

$alert_host = "local host":

$alert_port = "";

$alert_user="snort":

$alert_password = "snortpass";

При цьому ACID вестиме пошук коду ADODB в каталозі adodb, що знаходиться на одному рівні з каталогом acid. Окрім цього, ACID підключатиметься до БД MySQL (з ім'ям SNORT), розміщеної на локальній машині, що не вимагає вказівки номера порту. Якщо необхідно підключитися до БД, розташованої в іншій системі, вимагається вказати порт 3389, який для MySQL є портом за умовчанням. Додатково можна настроїти архів БД для ACID за допомогою змінних, аналогічних змінним, що використалися для налаштування БД сигналів тривоги:

$archive_dbname

$archive_host

$archive_port

$archive_user

$archive_password

Для вказівки місця розташування бібліотек графіки необхідно встановити змінну $ChartLib_path. При використанні JPGraph 1.13 і розпаковуванні її в той же каталог, в який розпакований дистрибутив ACID, можна ввести:

$ChartLib_path = "../jpgraph-1.13/src";

З налаштуванням скінчено. Тепер можна відкрити web-браузер і вказати URL, що відповідає каталогу, в який був розпакований ACID. Повинна з'явитися

сторінка налаштування БД. Перед використанням ACID необхідно створити декілька таблиць. Треба клацнути на кнопці Create ACID AG. Після цього на екрані з'явиться повідомлення про створення таблиць. Для таблиці подій краще створити індекси, якщо це не було зроблено під час налаштування. Індекси значно підвищують швидкість виконання запитів у великих таблицях, трохи збільшуючи використання дискового простору. Треба клацнути на посиланні Home для переходу до основної сторінки ACID. Інтерфейс ACID не вимагає ніяких пояснень. Основна таблиця має досить посилань, що дозволяють переглядати БД в різних представленнях: з виведенням списку IP-адрес джерела і пункту призначення, пов'язаних з сигналом тривоги, а також номерів портів.

2.8.3 Запобігання і стримування вторгнень за допомогою Snort_inline

Було б добре, якби NIDS могли не лише виявляти вторгнення, але і робити що-небудь для його нейтралізації. Добре, якби вони могли зупинити проникнення на вузол, що атакувався, але ще краще, якби вони блокували увесь мережевий трафік, що поширює атаку. Це можна зробити за допомогою Snort_inline.

Snort_inline - це оновлення (patch) для Snort, що дозволяє прочитувати дані з черги Netfilter в ядрі Linux, що забезпечує ефективну інтеграцію Snort в цей міжмережевий екран. При цьому не лише виявляється вторгнення, але і приймається рішення: скинути пакети або перенаправити їх на інший вузол (за допомогою Libnet). Звичайно ж, це вимагає перекомпіляції ядра з підтримкою IP-черги (IP queue) статично або в якості окремого модуля. Перевірити наявність цього модуля можна за допомогою команди:

$ locate ip_queue.o

/usr/src/linux-2.4.20-8/net/ipv4/netfilter/ip_queue.o

/usr/src/1 i nux-2.4.20-8/net/i pv4/netfi1ter/.ip__queue.o.f1ags

/lib/modules/2.4.20-8/kernel/net/ipv4/netfilter/ip_queue.o

В даному випадку останній рядок свідчить про наявність модуля. Якщо модуля немає, необхідно перевірити, чи існує файл /proc/net/ip_queue. Якщо

знайти модуль не вдається, а файл існує, то підтримка IP-черги скомпільована в ядро статично. Якщо і файлу немає, то необхідно включити його в ядро і перекомпілювати. Окрім підтримки IP-черги, для роботи Snort_inline вимагається також libipq. Це бібліотека, яка входить до складу Netfilter і використовується додатками для підключення до черги Netfilter. Перевірка наявності цієї бібліотеки здійснюється командою:

$ locate libipq

/usr/include/libipq.h

/lib/libipq.a

Якщо результат буде негативний - libipq не встановлена. Для установки необхідно завантажити початковий текст iptables з сайту Netfilter. Після компіляції запустити make install - dev. Окрім цієї бібліотеки, потрібна також бібліотека інжекції пакетів libnet. Для установки libnet просто завантажити з мережі початковий дистрибутив, розпакувати його і виконати з правами root команду run ./configure && make install.[17]

Тепер, коли усе підготовлено, можна скомпілювати Snort_inline. Спочатку треба завантажити і розпакуйте дистрибутив, а потім перейти в створений при цьому каталог і виконати команду:

$ ./configure --enable-inline && make

Також можна вказати параметри конфігурації, які зазвичай використовуються для Snort, оскільки в основі Snort_inline як і раніше залишається Snort.

Після завершення компіляції отримати root-привілеї і ввести make install. Тепер необхідно настроїти Snort_inline, так само начебто конфігурується Snort. Проте, якщо є бажання отримувати лише сповіщення, рекомендується запускати окремий екземпляр Snort, а для налаштування правил міжмережевого крану запускати тільки Snort_inline. Окрім того що Snort тепер захоплює пакети від Netfilter, а не від libpcap, оновлення додає в нього три нові типи правил : drop, sdrop і reject. Тип drop скидатиме пакети, що запустили правило без повідомлення відправляючого вузла (так само, як DROP в iptables), виконуючи

лише протоколювання. Тип sdrop аналогічний попередньому, але виконується без протоколювання. Використання типу reject призводить до блокування "ворожого" пакету, але з повідомленням відправляючої сторони за допомогою TCP RST або повідомленням про неможливість доставки ICMP - залежно від типу протоколу, використовуваного пакетом, що запустив правило: TCP або UDP відповідно. Новий параметр правил, доданий за рахунок оновлення Snort_inline, дозволяє замінювати будь-який зміст пакету будь-яким значенням. Єдина вимога до заміни: довжина заміщення має дорівнювати довжині потоку байтів, що заміщається. Це реалізується за допомогою параметра replace, використовуваного спільно з параметром content.

Snort_inline запускається так само, як і Snort, хоча оновлення Snort_inline додало новий ключ командного рядка - Q, що примушує використати IP-чергу, а не libpcap. Цей ключ використовується для включення inline-режима.

Усе, що залишилося зробити перед дійсним запуском в inline-режимі, - налаштувати в ядрі відправку пакетів в IP-черги. Це робиться за допомогою команди iptables:

iptables -A INPUT -j QUEUE

iptables -A OUTPUT -j QUEUE

iptables -A FORWARD -j QUEUE

Це призведе до "проштовхування" усього трафіку, що приходить, відправляється або проходить, в IP-чергу, з якої Snort_inline прочитуватиме пакети. Тепер можна запустити Snort_inline (тільки не слід забувати вказати ключ - Q) :

snort inline -Qvc /etc/snort/snort inline.conf

2.8.4 Виявлення аномільної поведінки

Більшість NIDS спостерігають за мережею, виконуючи пошук специфічних проявів атак. Ще один спосіб виявлення вторгнення полягає в накопиченні статистики звичайного трафіку мережі і подання сигналу тривоги при виявленні відхилень від усередненої норми. Однією з систем виявлення вторгнення такого

типу є Spade. Spade (скорочення від Statistical Anomaly Detection Engine (механізм виявлення статичної аномалії)) - це видозмінена версія Snort, функціональність якої розширена в область виявлення вторгнення на основі аномальної поведінки трафіку. Препроцесор Spade використовує Snort для спостереження за мережею і надалі будує імовірнісні таблиці, грунтуючись на відміченому трафіку. Після цього таблиця використовується для генерації для кожного пакету оцінного значення відхилення в діапазоні від 0 до 1 (де 0 – нормальний пакет, 1 – аномальний пакет). Установка Spade дуже проста. Просто треба завантажити дистрибутив з мережі, розпакувати його і перейти в створений при цьому каталог. Після цього виконати оновлення програмного коду Snort за допомогою команди:

$ make SNORTBASE=../snort-2.0.5

Зрозуміло, якщо дерево каталогів початкового коду Snort знаходиться не в ./snort-2.0.5, необхідно вказати точний шлях. Тепер перейти в каталог, що містить початковий код Snort, відкомпілювати і встановити його звичайним способом. Після цього необхідно конфігурувати Snort і Spade. Тут є два варіанти: налаштування використання лише функції Spade або спільна робота Snort і Spade. Для першого з цих варіантів в якості відправного пункту необхідно використати файл spade.conf, розміщений в дистрибутиві Spade. Більшість пропонованих значень нас влаштовують. Проте в змінній SPADEDIR необхідно визначити, відносно якого каталогу Snort має права запису і читання. У ньому Spade зберігатиме різні протоколи і значення контрольних точок, тому після перезапуску Snort таблиця вірогідності не загубиться: var SPADEDIR /var/log/snort/spade. Для Spade важливо знати, яка мережа є "домашньою". Визначити це можна за допомогою наступного запису в конфігураційному файлі:

peprocessor spade-homenet: 192.168.1.0/24

Можна вказати декілька мереж, розділивши їх двокрапками і взявши увесь список в квадратні дужки. Якщо вимагається забезпечити спільну роботу Snort і Spade і зберегти звичайну функціональність Snort, то файл spade.conf необхідно включити у файл snort.conf:

include spade.conf

Можна запустити Snort, як це робилося раніше. Тепер Spade при виявленні аномальної поведінки відправлятиме свій висновок у будь-яку конфігуровану надбудову виводу. Це спрацює в тих випадках, коли значення відхилення конкретного пакету складає 0,8-0,9 (залежно від типу пакету). Будь-які сигнали тривоги, згенеровані Spade, матимуть приставку Spade і будуть містити опис відхилення пакету від норми і значення цього відхилення.

2.8.5 Керування датчиками Snort

Керування IDS-датчиками і відстежування усіх генерованих ними сигналів тревоги може стати обтяжливим завданням, а за наявності декількох датчиків - взагалі важко здійснимою. Одним із способів об'єднати усі завдання управління IDS в одному застосуванні являється SnortCenter - система управління для Snort. SnortCenter складається з консолі, заснованої на web-интерфейсі, і агентів датчиків, що запускаються на кожній машині, що входить в інфраструктуру NIDS. Це дозволяє об'єднати усі завдання спостереження і управління в одній програмі, яка допомагає швидше виконувати роботу. SnortCenter має власну схему ідентифікації і підтримує зашифрований зв'язок між консоллю і окремими агентами датчиків. Це дозволяє оновлювати Snort-правила на безлічі датчиків або створювати власні правила і безпечно доставляти їх датчикам. SnortCenter також дозволяє віддалено запускати або зупиняти датчики. Для спостереження за сигналами тривоги SnortCenter може интегруватися з ACID.

Спочатку необхідно встановити консоль управління SnortCenter на web-сервер, що має як підтримку PHP, так і доступ до сервера баз даних MySQL, де SnortCenter зберігатиме БД конфігурації. Для установки консолі управління треба завантажити дистрибутив із мережі і распакувати його. При цьому буде створений каталог www (не треба розпаковувати у вже існуючий однойменний каталог), в який будуть поміщені PHP-сценарії SnortCenter, графіка і SQL - схемы. Після цього треба скопіювати вміст каталогу www у відповідне місце в кореневому каталозі документів web-сервера:

tar xfz snortcenter-v1.0-RC1.tar.gz

cp –R www /var/www/htdocs/snortcenter

Для забезпечення зв'язку SnortCenter з БД необхідно також встановити ADODB. Це PHP-пакет, що забезпечує абстрактнее функціонування БД. Після завантаження з мережі ADODB-кода треба розпакувати його у кореневий каталог документів (/var/www/htdocs). Окрім цього необхідно встановити curl. Треба завантажте дистрибутив, розпакувати його і запустити ./configure && make install. Після того, як усе виконано, необхідно уточнити config.php застосування SnortCenter (тобто файл /var/www/htdocs/snortcenter/config.php), вказавши в перелічених далі змінних значення, що відповідають необхідним умовам:

$DBlib_path = “../adodb/”;

$DBtype = “mysql”;

$DB_dbname = “SNORTCENTER”;

$DB_host = “localhost”;

$DB_port = “ ”;

$DB_user = “snortcenter”;

$DB_password = “snortcenterpass”;

$hidden_key_num = 182465687584776;

Таке налаштування вкаже SnortCenter, що ADODB-код потрібно шукати в каталозі adodb, розташованому в тому ж рівні, що і SnortCenter. Крім того, SnortCenter повинен підключатися до розташованої на локальній машині БД MySQL з найменуванням SNORTCENTER, використовуючи ім'я snortcenter і пароль snortcenterpass. Оскільки підключення до MySQL-серверу відбувається на локальній машині, немає необхідності вказувати номер порту. При підключенні до БД, встановленої на іншій системі, необхідно вказати 3389-й порт, що являється для MySQL портом за умовчанням. В якості значення змінної $hidden_key_num треба вказати випадковий номер.

Після редагування config.php необхідно створити БД і встановити для неї вказані раніше реєстраційні ім'я і пароль:

$ mysql –u root –p mysql

Enter password:

Reading table information for completion of table and column names

You can turn off this feature to get a quicker startup with –A

Welcome to MySQL monitor. Commands end with : or \g

Your MySQL connection id is 27 to server version 3.23.55

Type ‘help;’ or ‘\h’ for help. Type ‘\c’ to clear the buffer.

mysql> create database SNORTCENTER;

Query OK, 1 row affected (0.01 sec)

mysql> GRANT SELECT , INSERT , UPDATE , DELETE ON SNORTCENTER. * TO\

snortcenter@localhost IDENTIFIED BY ‘snortcenterpass’;

Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.02 sec)

mysql> Bye

Now create the database tables:

$mysql –u root –p SNORTCENTER < snortcenter_db.mysql

Тепер прийшла черга випробувати SnortCenter. Для цього необхідно ввести URL, що відповідає місцю установки в кореневому каталозі документів (http://snortcenter/).

Треба ввести ім'я і пароль за умовчанням і клацнути на кнопці Login. Відкриється консоль управління.

Зараз точно можна сказати про те, що консоль управління встановлена вірно, і тепер можна встановлювати агентів. Але перед цим необхідно змінити пароль облікового запису адміністратора. Для цього треба клацнути на кнопці Admin і вибрати пункт меню User Administration.

Тепер можна перейти до налаштування агента датчика. Агенти датчиків SnortCenter написані на мові Perl і для зв'язку з консоллю управління по зашифрованому каналу вимагають наявності модуля Net::SSLeay. Якщо

встановлений модуль CPAN (що відноситься до Perl), то установка Net::SSLeay виконується командою:

perl - MCPAN - e "install Net::SSLeay"

Для установки програмного коду агента його спочатку необхідно розпакувати. При розпаковуванні буде створений каталог sensor, в який буде поміщений увесь код агента. Треба скопіювати цей каталог в надійне постійне місце:

tar xfz /tmp/snortcenter-agent-v1.0-RC1.tar.gz

cp -R sensor /usr/local/snortcenter

Після цього необхідно створити SSL-сертифікат:

cd /usr/local/snortcenter

mkdir conf

openssl req –new –x509 –days 3650 –nodes \

-out conf/sensor.pem –keyout conf/sensor.pem

Далі потрібно запустити сценарій налаштування агента датчика:

sh setup.sh

Цей сценарій уточнить у вас деяку інформацію: каталог конфігураційного файлу агента і каталог протоколу, повний шлях до бібліотеки Perl (/usr/bin/perl), а також розташування двійкових файлів і правил Snort. Крім того, будуть уточнені операційна система, порт і IP-адрес, який повинен "прослуховувати" агент (за умовчанням 2525-й TCP-порт), а також IP-адреси, по яких можна підключатися до агента. Після отримання усієї информації сценарій запустить агента на вказаному в конфігураційному файлі порту. Тепер можна перевірити роботу агента, звернувшись до нього за допомогою web -браузера (використовувати треба протокол https, а не http).

Тепер можна повернутися до головної консолі керування й добавити створений датчик. Для цього знову треба увійти до сторінки консолі та вибрати пункт Add sensor з меню Sensor Console (керування сенсорами).[18]

Треба ввести ті ж відомості, які були вказані при виконанні сценарію налаштування, і клацнути на кнопці Save. Базову конфігурацію датчика можна

"проштовхнути" в датчик, вибравши Admin - Import/Update Rules - Update from Internet (Адміністрування - Імпорт/Оновлення правил - Відновити через інтернет). Після цього необхідно повернутися до списку датчиків, клацнувши Sensor Consoles – View Sensors – Push hyperlink for the sensor (Управління сенсорами – Показати сенсори – Передати сенсору гіперпосилання). Для запуску Snort відносно конкретного датчика треба клацнути на посиланні Start.

Тепер можна настроїти датчик за допомогою меню Sensor Config і Resources. Після того як створена конфігурація, що задовольняє адміністратора, вона може "проштовхуватися" необхідним датчикам за допомогою пункту Push.

2.8.6 Створення власних правил Snort

Однією з яскравих особливостей Snort є власний процесор правил.

Процесор правил має розширену мову, що дозволяє писати власні правила, що враховують особливості конкретної мережі. Правило Snort можна розділити на дві частини: заголовок і параметри. Заголовок містить виконувану дію, протокол, до якого застосовується правило, а також адреси і порти початкового пункту і пункту призначення. Параметри правила дозволяють створювати пов'язане з правилом пояснювальне повідомлення, а також перевіряти різні атрибути пакетів з використанням досить великої бібліотеки надбудов Snort. Ось загальна форма правила:

action proto src_ip src_port direction dst_ip dst_port (options)

При надходженні пакету його IP-адреса і порти посилача і одержувача порівнюються зі значеннями, вказаними в правилах. Якщо вони співпадають, то з пакетом порівнюються параметри. Якщо в результаті цих порівнянь виявляється збіг, то робиться вказана дія.

Snort пропонує декілька вбудованих дій, які можна використати у правилах. Для спрощення протоколювання співпадаючих пакетів використовується дія log. Дія alert, окрім протоколювання пакету, генерує сигнал тривоги вказаним в конфігураційному файлі або в командному рядку способом. Дуже приємна можливість Snort полягає в тому, що можна мати досить загальні

правила, а потім створити з них виключення, написавши ще одне правило, що використовує дію pass. Це особливо зручно при використанні наявних в Snort правив, але часто призводить до отримання невірного результату. У таких випадках ризику для системи безпеки немає, просто треба написати для них pass-правило.

Дві останні вбудовані дії правил, activate і dynamic, використовуються разом для динамічної зміни набору правил Snort в ході виконання. Правила, що використовують дію dynamic, подібні log-правилам, за виключенням того, що братимуться до уваги тільки після дозволу правилом activate. Snort примушує використати параметри правил activates і activated_by, для того, щоб знати, які dynamic-правила дозволити, після того, як спрацює activate-правило. Крім того, dynamic-правила вимагають використання параметра count для обмеження кількості записуваних правилом пакетів.

Якщо потрібно почати запис пакетів після того, як буде помічено "ненормальне" використання SSH-демона на машині 192.168.1.22, необхідно використати подібну пару правил :

```
activate tcp any any -> 192.168.1.21 22 (content:"Vbin/sh"; activates:!; \
msg:"Possible SSH buffer overflow"; )
dynamic tcp any any -> 192.168.1.21 22 (activated_by:l; count:100;)
```

Ці два правила не забезпечують повний "захист від дурня", але якщо хто-небудь запустить код оболонки відносно SSH-демона, то, швидше за все, він відправить рядок /bin/sh у відкритому виді, щоб породити підпроцес оболонки на тій системі, що атакується. Крім того, оскільки SSH зашифрований, такий рядок не міг бути посланий при звичайних обставинах. Після того, як спрацює перше правило, воно активізує друге, яке запише 100 пакетів, а потім зупиниться. Це корисно в тих випадках, коли необхідно присікти спробу зловмисника завантажити або встановити так званий root kit (набір суперкористувача) з перших же пакетів, а також якщо треба швидше проаналізувати скомпрометовану систему.

Крім того, окрім вбудованих дій, є можливість визначити користувацькі дії правил. Це робиться за допомогою ключового слова ruletype :

```
ruletype redalert
{
type alert
output alert_syslog: LOG_AUTH LOG_ALERT
output database: log, mysql. user=snort dbname=snort host=localhost
}
```

Ця дія вказує, що Snort повинен поводитися так само, як при виконанні дії alert, але що сигнал тривоги має бути посланий демону syslog, а пакети повинні протоколюватися у базу даних. При визначенні дії можна використати будь-яку з надбудов виведення Snort так само, як при налаштуванні їх в якості основного варіанту виводу.

Механізм виявлення Snort підтримує декілька протоколів. Поле proto використовується для вказівки протоколу, до якого відноситься правило. Допустовими значеннями є ip, icmp, tcp та udp.

Наступне поле в правилі використовується для вказівки IP-адрес і портів початкового пункту і пункту призначення, а також напряму переміщення пакетів. Snort сприймає один IP-адрес або список адрес. При завданні списку адреси розділяються комами, а увесь список ставиться в квадратні дужки:

[192.168.1.1,192.168.1.20,192.168.1.24]

Пропусків в списку бути не повинно. Крім того, дозволяється використання логічного оператора NOT відносно IP-адреси або діапазону, що вказує виключення з правила. Так само як і у випадку з IP -адресами, Snort сприймає як окремі порти, так і діапазон портів. При вказівці диапазона як роздільник використовується двокрапка. Далі показано, як вказуються усі порти з 1-го по 1024-ий:

1:1024

До номерів портів також можна застосовувати оператор NOT, дозволяється позначення діапазону без верхньої та нижньої меж. Для перевірки всіх портів вище 1024-го потрібно вказати:

1024:

По аналогії, усі порти до 1024-го можна вказати так:

:1024

Якщо IP-адреса і номер порту не критичні, то можна вказати any (будь-хто). Поле direction використовується для вказівки того, яка IP-адреса і порт є пунктом відправлення, а яка пара IP-адреси і порту - пунктом призначення. У ранніх версіях Snort в якості значень можна було використовувати оператори -> і <-. Проте оператор <- був замінений, оскільки, якщо поміняти місцями пари IP-адреса/порт, він стає рівнозначний іншому. Окрім знаку -> є ще один. Значення <> вказує, що правило застосовується в обох напрямках. Це особливо корисно, коли використовуються log- або dynamic-правила, оскільки протоколюються обидві сторони TCP-потоку.

Наступна частина правила складається з параметрів. Вона дозволяє вказувати багато атрибутів, відносно яких виконуватиметься перевірка. Кожен параметр реалізується надбудовою Snort. Коли спрацьовує правило, яке визначає параметр, для перевірки пакету Snort запустить ту надбудову, що відповідає цьому параметру. Snort має більше 40 надбудов - надто багато, щоб розглядати їх детально. Далі будуть розглянуті найбільш корисні з них.

Найпотужнішим параметром є msg. Він дозволяє вказувати повідомлення, які протоколюватимуться у вигляді сигналу тривоги при виявленні пакету, співпадаючого з правилом. Без нього багато сигналів тривоги не матимуть сенсу.

Параметр в якості аргументу має рядкове значення, поміщене в лапки. Наступний вираз визначає повідомлення, генероване при виявленні якого-небудь трафіку, витікаючого з адреси 192.168.1.31:

alert tcp 192.168.1.35 any -> any any (msg:"Traffic from 192.168.1.31":)

У рядковому значенні мають бути відсутніми будь-які лапки. Синтаксичний аналізатор в Snort дуже простий і не підтримує ці символи. Ще

одним корисним параметром є content. Він дозволяє знаходити пакети за послідовністю символів або шістнадцятиричних значень. При пошуку рядкового значення його необхідно просто помістити в лапки. При пошуку без урахування регістра необхідно додати уточнення nocase; у кінці усіх параметрів. При пошуку послідовності шістнадцятиричних чисел їх потрібно помістити між символами |. Це правило спрацює, якщо "побачить" число 0x90:

```
alert tcp any any -> any any (msg:"Possible exploit"; content:"|90|";)
```

Це цифрове значення є шістнадцятиричним еквівалентом інструкції NOP, що використовується в x86-архітектурі, і часто зустрічається в коді взлому, оскільки полегшує запис при виконанні атаки переповнювання буфера.

Параметри offset і depth можуть використовуватися спільно з параметром content для обмеження навантаження при виконанні контекстного пошуку в розділі даних, вони точніше вказують діапазон байтів, в яких виконуватиметься пошук. Якщо необхідно обмежити пошук інструкції NOP в розділі даних пакета між 40-м і 75-м байтом, те попереднє правило необхідно змінити таким чином:

```
alert tcp any any -> any any (msg:"Possible exploit": content:"|90|"; \
offset:40: depth:75;)
```

Можна виконувати пошук пакетів, в яких відсутня вказана послідовність, встановивши на початку символ !. Крім того, велика частина "корисного навантаження" коду оболонки може бути занадто великою в порівнянні із звичайною кількістю даних, що переносяться пакетом в певну службу. За допомогою параметра dsize можна перевірити розмір даних "корисного навантаження" в пакеті. Цей параметр отримує в якості аргументу число. Окрім цього, за допомогою операторів > і < можна вказати верхню або нижню межу відповідно:

```
alert tcp any any -> any any (msg:"Possible exploit"; content:"|90|"; \
offset:40; depth:75; dsize: >6000;)
```

На відміну від попереднього прикладу, цьому правилу відповідатимуть пакети, у яких (окрім інших умов) розмір даних "корисного навантаження" перевищує 6000 байт.

Для перевірки TCP-прапорів пакетів Snort надає параметр flags. Він корисний для виявлення сканування портів, під час якого використовуються недійсні поєднання прапорів. Це правило виявить сканування, коли одночасно будуть встановлені прапори SYN і FIN:

alert any any -> any any (flags: SF,12; msg: "Possible SYN FIN scan";)

Дійсними значеннями прапорів є S для SYN, F для FIN, R для RST, P для PSH, A для ACK і U для URG. Крім того, Snort дозволяє перевіряти стан двох зарезервованих розрядів-прапорів, вказаних значенням 1 чи 2. Вказавши значення 0, можна встановити відповідність пакету, що взагалі не містить прапорів. Окрім цього, параметр flags сприймає декілька операторів. Перед вказуваними значеннями можна додавати +, * або ! для відповідності одним прапорам плюс іншим, будь-яким прапорам і усім, окрім вказаних, відповідно.

В таблиці 2.5 представлено перелік модулів, що входять до системи виявлення вторгнень Snort, а також наведено призначення вказаних модулів. Кожен модуль виконує специфічні функції, притаманні тільки для нього.

Таблиця 2.5 – Перелік модулів, що входять до складу NIDS Snort

До впровадження	Модуль	Призначення
Відсутність засобів виявлення вторгнень	NIDS Snort	Система виявлення вторгнень
Незручність обробки даних, що генеруються NIDS	Analysis Console for Instruction Databases (ACID)	Web-інтерфейс для БД, що містять сигнали тривоги, які надійшли від NIDS
Неможливість запобігання вторгнень	Snort_inline	Оновлення для Snort, функція якого – запобігання та стримування вторгнень
Неможливість аналізувати трафік на предмет аномалій	Spade	Оновлення для Snort, що дозволяє виявляти аномальну поведінку трафіку
Неможливість створення централізованої системи сенсорів	SnortCenter	Система керування сенсорами

На рисунку 2.16 наведено детальну схему роботи комплексної системи виявлення вторгнень, яку було розгорнуто на підприємстві. Показано взаємодію основних елементів системи, принцип розташування сенсорів на робочих станціях для детального моніторингу кожного вузла мережі, а також підпорядкованість модулів системи. Слід зазначити, що головний елемент системи виявлення вторгнень – модуль Snort – встановлено на серверний комп'ютер.

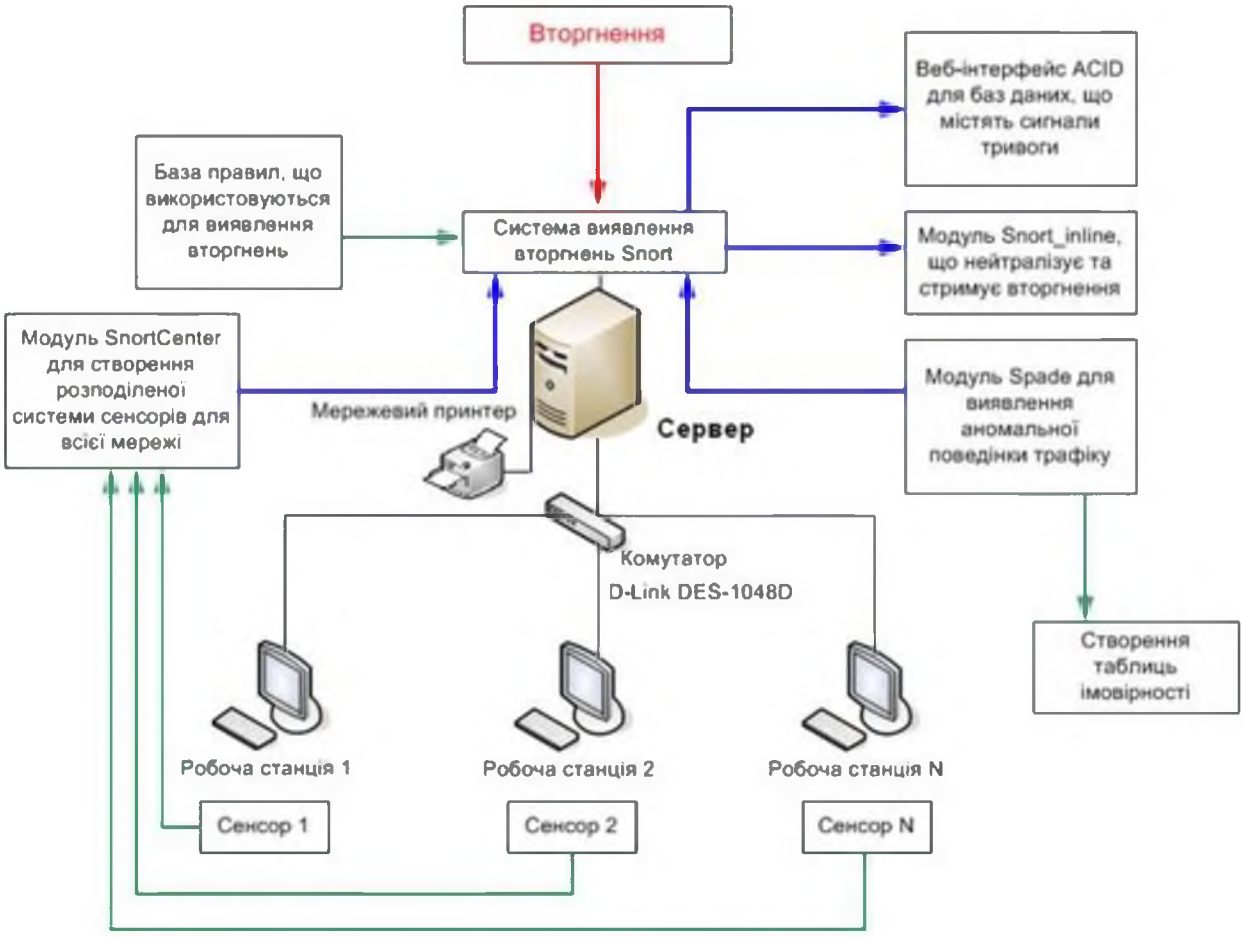

Рисунок 2.16 – Функціональна схема комплексної системи виявлення вторгнень

В узагальнюючій таблиці 2.6 представлено усі заходи, що були виконані з метою мінімізування ймовірності проведення мережевих атак на ІС ПП «АрдКом». В таблицю увійшли найбільш небезпечні для даної системи атаки, а також, відповідно, засоби захисту від них.

Таблиця 2.6 – Звідна таблиця захищеності підприємства

Тип атаки	Захищеність ІС від атаки			
	До застосування заходів захисту	Загрози, що можуть бути реалізовані	Після застосування заходів захисту	Чим забезпечується
Снiфінг	Не захищено	Крадіжка інформації	Захищено	Шифрування мережевого трафіку завдяки ПЗ «Stunnel»
ARP-спуфінг	Не захищено	Крадіжка інформації; модифікація та знищення інформації	Захищено	Застосування ПЗ «arpwatch», створення статичної ARP-таблиці
Dos-атаки	Слабкий захист	Порушення нормальної роботи	Захищено	Вбудований в серверну ОС брандмауер «Netfilter», відстеження зайвих пакетів за допомогою NIDS «Snort»
Nuke-атаки	Не захищено	Порушення нормальної роботи	Захищено	Більш якісне налаштування захищеності ОС робочих станцій
NetBios-атаки	Не захищено	Крадіжка інформації; модифікація інформації; знищення інформації; порушення встановлених правил доступу	Захищено	Відключення загальнодоступних ресурсів мережі, більш складні паролі користувачів, перекривання 135-го та 139-го портів
Виявлення мережевих вторгнень на початкових стадіях	Відсутнє	Усі можливі загрози, які приведені в моделі загроз	Присутнє	Система виявлення вторгнень NIDS «Snort» з деякими надбудовами, що покращують зручність та якість роботи системи

2.9 Висновок

Була проаналізована робота інформаційної системи ПП «АрдКом», наведена характеристика оброблюваної в ній інформації, проведено аналіз інформаційних загроз, вибрано профіль захищеності.

По-перше, було розроблено та запропоновано заходи щодо забезпечення захищеності робочих станцій з використанням ОС Windows та мережевої безпеки ІС підприємства.

По-друге, було сплановано та створено комплексну систему виявлення вторгнень, після чого система була впроваджена у робочий процес ІС підприємства.

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

Будь-яка інформаційно-комунікаційна система потребує індивідуального підходу до забезпечення якісного захисту та ефективної роботи. В будь-якій системі обробляється різна інформація з різною вартістю. За для прийняття рішення про забезпечення інформаційної безпеки підприємства в контексті зниження ймовірності проведення спектру атак на інформаційно-комунікаційну систему, які б відповідали вимогам конкретного підприємства, а також складались з оптимальної комбінації організаційних зауважень та програмних модулів, необхідно такі рішення обґрунтовувати як з технічної, так і з економічної точки зору. Тому метою даного розділу є обґрунтування економічної доцільності розробки системи виявлення мережевих атак на інформаційну систему приватного підприємства "АрдКом". Для досягнення цієї необхідно здійснити наступні розрахунки: капітальні витрати на придбання і налагодження складових системи інформаційної безпеки або витрат, що пов’язані з виготовленням апаратури, приладів, програмного забезпечення; річні експлуатаційні витрати на утримання і обслуговування об'єкта проектування; річний економічний ефект від впровадження запропонованих заходів; показники економічної ефективності впровадження системи захисту на підприємстві.

3.1 Розрахунок (фіксованих) капітальних витрат

Капітальними витрати називаються тому, що робляться, як правило, один раз, на початкових етапах створення ІС. До капітальних слід відносити наступні витрати: вартість розробки і впровадження проекту; залучення зовнішніх консультантів; первинні закупівлі основного ПЗ; первинні закупівлі додаткового ПЗ; первинні закупівлі апаратного забезпечення тощо.

Витрати на впровадження системи захисту інформації на підприємстві визначаються, виходячи з трудомісткості кожної операції, та належать до капітальних витрат.

Визначення трудомісткості розробки підходу щодо розробка системи виявлення мережевих атак на інформаційну систему приватного підприємства

Трудомісткість розробки системи захисту інформації на підприємстві визначається тривалістю кожної робочої операції, до яких належать наступні:

де $t_{тз}$ – тривалість складання технічного завдання на розробку системи виявлення мережевих атак на інформаційну систему приватного підприємства, $t_{тз}$=10;

$t_в$ – тривалість аналізу існуючих інформаційних потоків організації, вивчення ТЗ, літературних джерел за темою тощо, $t_в$=*24*;

$t_а$ – тривалість аналізу існуючих загроз безпеки інформації, $t_а$=*32*;

$t_р$ – тривалість розробки засобів захисту інформації в інформаційно-комунікаційній системі підприємства, $t_м$=*28*;

$t_д$ – тривалість підготовки технічної документації, $t_д$=*10*.

Отже,

$$t = t_{тз} + t_в + t_а + t_р + t_д = 10+24+32+28+10 = 104 \text{ години.}$$

Розрахунок витрат на розробка системи виявлення мережевих атак на інформаційну систему приватного підприємства

Витрати на розробку системи захисту інформації на підприємстві $К_{рп}$ складаються з витрат на заробітну плату спеціаліста з інформаційної безпеки $З_{зп}$ і вартості витрат машинного часу, що необхідний для розробки політики безпеки інформації $З_{мч}$.

$$К_{рп} = З_{зп} + З_{мч}\ .$$

$$К_{рп} = З_{зп} + З_{мч} = 27040 + 743{,}6 = 27783{,}6 \text{ грн.}$$

$$З_{зп} = t\ З_{пр} = 104*260 = 27040 \text{ грн.}$$

де t – загальна тривалість розробки політики безпеки, годин;

$З_{іб}$ – средньогодинна заробітна плата спеціаліста з інформаційної безпеки з нарахуваннями, грн/годину.

Вартість машинного часу для розробки політики безпеки інформації на ПК визначається за формулою:

$$З_{мч}= t*С_{мч}=104*7,15=743,6 \text{ грн.}$$

де $t_д$ – трудомісткість підготовки документації на ПК, годин;

$С_{мч}$ – вартість 1 години машинного часу ПК, грн./година.

Вартість 1 години машинного часу ПК визначається за формулою:

$$С_{мч} = 0,9 \cdot 4 \cdot 1,68 + \frac{12100 \cdot 0,1}{1920} + \frac{4500 \cdot 0,2}{1920} = 7,15 \text{ грн.}$$

В інформаційно-комунікаційній мережі приватного підприємства "АрдКом" використовується наступне програмне забезпечення: ОС серверних станцій Ubuntu Server; ОС робочих станцій Microsoft Windows 7; 1С Бухгалтерия 8; Microsoft Office 365; Avast.

В інформаційній мережі ПП «АрдКом» для забезпечення синтезу системи виявлення мережевих вторгнень були обрані наступні програми:

1) для протидії ARP-спуфінгу - програма Arpwatch.

2) для шифрування мережевого трафіку - програма Stunnel.

3) для протидії мережевим вторгненням - програмний комплекс Snort.

Обрані програми можна використовувати на безоплатній основі. У зв’язку з вищезазначеним, додаткові витрати на придбання та оновлення програмного забезпечення для забезпечення синтезу системи виявлення мережевих вторгнень не виникають.

При цьому виникатимуть додаткові витрати, пов’язані із залученням зовнішніх консультантів, які складатимуть 7000 грн.

Таким чином, капітальні (фіксовані) витрати на розробку засобів підвищення рівня інформаційної безпеки складуть:

$К = К_{рп} + К_{зпз} + К_{пз} + К_{аз} + К_{навч} + К_{н} = 27783,6 + 7000 = 34783,6$ грн.

де $К_{рп}$ – вартість розробки політики інформаційної безпеки та залучення для цього зовнішніх консультантів, тис. грн;

$К_{зпз}$ – вартість закупівель ліцензійного основного й додаткового програмного забезпечення (ПЗ), тис. грн;

$К_{пз}$ – вартість створення основного й додаткового програмного забезпечення, тис. грн;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів, тис. грн;

$К_{навч}$ – витрати на навчання технічних фахівців і обслуговуючого персоналу,

$К_{н}$ – витрати на встановлення обладнання та налагодження системи інформаційної безпеки.

3.1.1 Розрахунок поточних витрат

Річні поточні витрати на функціонування системи інформаційної безпеки складають:

$$С = С_{в} + С_{к} + С_{ак}, \quad грн.$$

де $С_{в}$ - вартість відновлення й модернізації системи ($С_{в} = 0$);

$С_{к}$ - витрати на керування системою в цілому;

$С_{ак}$ - витрати, викликані активністю користувачів системи інформаційної безпеки ($С_{ак} = 0$ грн.).

Витрати на керування системою інформаційної безпеки ($С_{к}$) складають:

$$С_{к} = С_{н} + С_{а} + С_{з} + С_{ел} + С_{о} + С_{тос}, \text{ грн.}$$

Витрати на навчання адміністративного персоналу й кінцевих користувачів визначаються сукупною величиною витрат на організаційні заходи, які складуть 12000 грн.

Річний фонд заробітної плати інженерно-технічного персоналу, що обслуговує систему інформаційної безпеки ($С_з$), складає:

$$С_з = З_{осн} + З_{дод}, \text{ грн.}$$

Основна заробітна плата визначається, виходячи з місячного посадового окладу, а додаткова заробітна плата – в розмірі 8-10% від основної заробітної плати.

Основна заробітна плата одного спеціаліста з інформаційної безпеки на місяць складає 24000 грн. Додаткова заробітна плата – 5% від основної заробітної плати. Виконання роботи щодо впровадження системи захисту інформації на підприємстві потребує залучення спеціаліста інформаційної безпеки на 0,2 ставки. Отже,

$$С_з = (24000*12 + 24000*12*0,05)*0,2= 60480 \text{ грн.}$$

Ставка ЄСВ для всіх категорій платників з 01.01.2021 р. складає 22%.

$$С_{єв} = 60480*0,22 = 13305,6 \text{ грн.}$$

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року ($С_{ел}$), визначається за формулою:

$$С_{ел} = Р \cdot F_р \cdot Ц_е, \text{ грн.,}$$

де Р – встановлена потужність апаратури інформаційної безпеки, (Р=0,9 кВт);

$F_р$ – річний фонд робочого часу системи інформаційної безпеки ($F_р$ = 1920 год.);

$Ц_е$ – тариф на електроенергію, ($Ц_е$ = 1,68 грн./кВт за годину).

Вартість електроенергії, що споживається апаратурою системою інформаційної безпеки протягом року:

$$С_{ел} = 0,9*1920*1,68 = 2903,04 \text{ грн.}$$

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки визначаються у відсотках від вартості капітальних витрат – 2% ($С_{тос}$ = 34783,6*0,02=695,67 грн).

Витрати на керування системою інформаційної безпеки ($С_к$) визначаються:

$$С_к = 12000+60480+13305,6+2903,04+695,67 = 89384,31 \text{ грн.}$$

Витрати, викликані активністю користувачів системи інформаційної безпеки ($С_{ак}$) можна орієнтовно визначити, виходячи з величини капітальних витрат та середньостатистичних даних щодо активності користувачів. За статистичними даними активність користувачів складає 25%. Тому:

$$С_{ак} = 34783,6*0,25=8695,9 \text{ грн.}$$

Таким чином, річні поточні витрати на функціонування системи інформаційної безпеки складають:

$$С = 0+89384,31+8695,9=98080,21 \text{ грн.}$$

3.2 Оцінка можливого збитку

Для розрахунку вартості такого збитку можна застосувати наступну спрощену модель оцінки.

Оцінка можливого збитку від атаки на вузол або сегмент мережі

Необхідні *вихідні дані* для розрахунку:

$t_п$ – час простою вузла або сегмента корпоративної мережі внаслідок атаки, 2 години;

$t_в$ – час відновлення після атаки персоналом, що обслуговує корпоративну мережу, 2 години;

$t_{ви}$ – час повторного введення загубленої інформації співробітниками атакованого вузла або сегмента корпоративної мережі, 2 години;

$З_о$ – заробітна плата обслуговуючого персоналу (адміністраторів та ін.), 24000 грн./міс.;

$З_с$ – заробітна плата співробітників атакованого вузла або сегмента корпоративної мережі, 18000 грн./міс.;

$Ч_о$ – чисельність обслуговуючого персоналу (адміністраторів та ін.), 1 особи;

$Ч_с$ – чисельність співробітників атакованого вузла або сегмента корпоративної мережі, 29 осіб.;

О – обсяг прибутку атакованого вузла або сегмента корпоративної мережі, 500 тис. грн. на рік;

$П_{зч}$ – вартість заміни встаткування або запасних частин, 0 грн;

I – число атакованих сегментів корпоративної мережі, 1;

N – середнє число атак на рік, 18.

Упущена вигода від простою атакованого сегмента корпоративної мережі становить:

$$U = П_п + П_в + V,$$

де $П_п$ – оплачувані втрати робочого часу та простої співробітників атакованого вузла або сегмента корпоративної мережі, грн;

$П_в$ – вартість відновлення працездатності вузла або сегмента корпоративної мережі (переустановлення системи, зміна конфігурації та ін.), грн;

V – втрати від зниження обсягу продажів за час простою атакованого вузла або сегмента корпоративної мережі, грн.

Втрати від зниження продуктивності співробітників атакованого вузла або сегмента корпоративної мережі являють собою втрати їхньої заробітної плати (оплата непродуктивної праці) за час простою внаслідок атаки:

$$П_п = \frac{\sum З_с}{F} t_п = \frac{18000*29}{176} * 2 = 5931{,}81 \text{ грн},$$

де F – місячний фонд робочого часу (при 40-а годинному робочому тижні становить 176 ч).

Витрати на відновлення працездатності вузла або сегмента корпоративної мережі включають кілька складових:

$$П_в = П_{ви} + П_{пв} + П_{зч},$$

де $П_{ви}$ – витрати на повторне уведення інформації, грн.;

$П_{пв}$ – витрати на відновлення вузла або сегмента корпоративної мережі, грн;

$П_{зч}$ – вартість заміни устаткування або запасних частин, грн.

Витрати на повторне введення інформації $П_{ви}$ розраховуються виходячи з розміру заробітної плати співробітників атакованого вузла або сегмента корпоративної мережі $З_с$, які зайняті повторним введенням втраченої інформації, з урахуванням необхідного для цього часу $t_{ви}$:

$$П_{ви} = \frac{\sum З_с}{F} t_{ви} = \frac{18000*29}{176} * 2 = 5931{,}81 \text{ грн}.$$

Витрати на відновлення сегмента корпоративної мережі $П_{ПВ}$ визначаються часом відновлення після атаки $t_в$ і розміром середньогодинної заробітної плати обслуговуючого персоналу (адміністраторів):

$$П_{ПВ} = \frac{\sum З_о}{F} t_в = \frac{24000*1}{176} * 2 = 272{,}2 \text{ грн.}$$

Тоді витрати на відновлення працездатності вузла або сегмента корпоративної мережі складуть:

$$П_в = 5931{,}81+272{,}2 = 6204{,}01 \text{ грн.}$$

Втрати від зниження очікуваного обсягу прибутків за час простою атакованого вузла або сегмента корпоративної мережі визначаються виходячи із средньогодинного обсягу прибутку і сумарного часу простою сегмента корпоративної мережі:

$$V = \frac{O}{Fг} \cdot (t_П + t_В + t_{ВИ})$$

$$V = \frac{500000}{2080} \cdot (2 + 2 + 2) = 1442{,}31 \text{ грн.}$$

де $F_г$ – річний фонд часу роботи філії (52 робочих тижні, 5-ти денний робочий тиждень, 8-ми годинний робочий день) становить близько 2080 год.

$$U = 5931{,}81+6204{,}01 +1442{,}31 = 13578{,}13 \text{ грн.}$$

Таким чином, загальний збиток від атаки на сегмент корпоративної мережі організації складе:

$$B = \sum_i \sum_n U = \sum_1 \sum_{18} 13578{,}13 = 244406{,}34 \text{ грн.}$$

3.2.1 Загальний ефект від впровадження системи інформаційної безпеки

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної:

$$E = B \cdot R - C \text{ грн.},$$

де В – загальний збиток від атаки у разі перехоплення інформації, тис. грн.;

R – вірогідність успішної реалізації загрози (52%);

С – щорічні витрати на експлуатацію системи інформаційної безпеки.

Загальний ефект від впровадження системи інформаційної безпеки визначається з урахуванням ризиків порушення інформаційної безпеки:

$$E = 244406{,}34 * 0{,}52 - 98080{,}21 = 29011{,}08 \text{ грн.}$$

3.3 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

Для встановлення економічної ефективності визначають такі показники як: коефіцієнт повернення інвестицій (ROSI) та термін окупності капітальних інвестицій (T_o).

Коефіцієнт повернення інвестицій ROSI показує, скільки гривень додаткового прибутку приносить одна гривня капітальних інвестицій на впровадження системи інформаційної безпеки:

$$ROSI = \frac{E}{K}, \quad \text{частки одиниці,}$$

де E – загальний ефект від впровадження системи інформаційної безпеки грн.;

К – капітальні інвестиції за варіантами, що забезпечили цей ефект, грн.

Коефіцієнт повернення інвестицій ROSI:

$$ROSI = \frac{29011,08}{34783,6} = 0,83 \text{ частки одиниці,}$$

Проект визнається економічно доцільним, якщо розрахункове значення коефіцієнта повернення інвестицій перевищує величину річної депозитної ставки з урахуванням інфляції:

$$ROSI > (N_{деп} - N_{інф})/100),$$

де $N_{деп}$ – річна депозитна ставка, (6%);

$N_{інф}$ – річний рівень інфляції, (5%).

Розрахункове значення коефіцієнта повернення інвестицій:

$$0,83 > (6 - 5)/100 = 0,83 > 0,01.$$

Термін окупності капітальних інвестицій T_o показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки:

$$T_o = \frac{K}{E} = \frac{1}{ROSI} = \frac{1}{0,83} = 1,2 \text{ роки.}$$

3.4 Висновок

Виходячи з наведених розрахунків, можна зробити висновок, що розробка системи виявлення мережевих атак на інформаційну систему приватного підприємства "АрдКом" може вважатися економічно доцільною. Про це свідчить отримане значення коефіцієнту повернення інвестицій (ROSI=0,83), що є вищим за альтернативний варіант вкладення коштів. Прогнозований економічний ефект складе 29011,08 грн. при капітальних витратах на рівні 34783,6 грн. і експлуатаційних витратах на рівні 98080,21 грн. Термін окупності складатиме 1,2 роки.

ВИСНОВКИ

У роботі було досліджено та запропоновано заходи, основна ціль яких – зробити синтез систем виявлення вторгнень, збільшити рівень захищеності інформації, що циркулює в ІС ПП «АрдКом», а також мінімізувати ймовірність проведення успішних атак на ІС.

Була проаналізована робота інформаційної системи ПП «АрдКом», характеристика оброблюваної в ній інформації, проведено аналіз інформаційних загроз, вибрано профіль захищеності. Також була доведена економічна окупність заходів та засобів, які запропоновані для захисту інформації в системі. На основі чого і було прийнято рішення про запровадження синтезу комплексної системи виявлення мережевих вторгнень на підприємстві.

По-перше, було розроблено та запропоновано заходи щодо забезпечення захищеності робочих станцій з використанням ОС Windows та мережевої безпеки ІС підприємства.

По-друге, було сплановано та створено комплексну систему виявлення вторгнень, після чого система була впроваджена у робочий процес ІС підприємства.

Запропоновані заходи щодо зниження ймовірності проведення атак не викликали ніяких нарікань з боку адміністраторів безпеки та співробітників ІС підприємства «АрдКом».

ПЕРЕЛІК ПОСИЛАНЬ

1 Глушаков С. В., Бабенко М. И., Тесленко Н. С. Секреты хакера. Атака и защита. Учебный курс. – АСТ Москва, 2008. – 544 с.: ил.

2 Бил Дж., Бейкер Э., Казуэлл Б. Snort Обнаружение вторжений. – Бином-Пресс, 2017. – 656 с.: ил.

3 Данжани Н., Кларк Д. Средства сетевой безопасности. – Кудиц-Пресс, 2017. – 368 с.: ил.

4 Локхарт Э. Антихакинг в сети. Трюки. – СПб., Питер, 2009. – 296 с.: ил.

5 Шаньгин В. Защита компьютерной информации. Эффективные методы и средства. – издательство «ДМК Пресс», 2014. – 544 с.: ил.

6 Мецатунян М., Ищейнов В. Защита конфиденциальной информации. – издательство «Форум», 2013. – 256 с.: ил.

7 Партыка П., Попов И. Информационная безопасность (3-е издание). – издательство «Форум», 2014. – 432 с.: ил.

8 Хорев П. Методы и средства защиты информации в компьютерных системах. – издательство «Academia», 2013. – 256 с.: ил.

9 Кинг Д.Р. Практические и доступные рекомендации по защите ПК. – издательство «НТ Пресс», 2017. – 240 с.: ил.

10 Смирнов С.Н. Безопасность систем баз данных. – издательство «Гелиос АРВ», 2017. – 352 с.: ил.

11 Шумский А.А., Шелупанов А.А. Системный анализ в защите информации. – издательство «Гелиос АРВ», 2015. – 244 с.: ил.

12 Девянин П. Модели безопасности компьютерных систем. – издательство «Academia», 2015. – 144 с.: ил.

13 Записки о Unix/Linux/BSD/Solaris (Електрон. ресурс) / Спосіб доступу: URL: http://unixa.com/. – Загол. з екрана.

14 Объединённый открытый проект. Сайт для компьютерщиков (Електрон. ресурс) / Спосіб доступу: URL: http://www.openproj.com/. – Загол. з екрана.

15 IT-Сектор (Електрон. ресурс) / Спосіб доступу: URL: http://it-sektor.ru/. – Загол. з еркана.

16 Системный администратор (Електрон. ресурс) / Спосіб доступу: URL: http://samag.ru/. Загол. з екрана.

17 LinuxCenter (Електрон. ресурс) / Спосіб доступу: URL: http://www.linuxcenter.com/. – Загол. з екрана.

18 НД ТЗІ 2.5-005-99 Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу / ДСТСЗІ СБ України – Київ, 1999.

19 НД ТЗІ 1.6-005-2013 Захист інформації на об'єктах інформаційної діяльності. Положення про категоріювання об'єктів, де циркулює інформація з обмеженим доступом, що не становить державної таємниці.

20 НД ТЗІ 1.4-001-2000 Типове положення про службу захисту інформації в автоматизованій системі.

21 НД ТЗІ 2.5-004-99 Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу / ДСТСЗІ СБ України – Київ, 1999.

22 Методы и средства защиты информации (Електрон.ресурс) / Спосіб доступу: URL: http://scanner.com/link/Safe/miszi.htm – Загол. з екрана.

23 Анализатор трафика (Електрон.ресурс) / Спосіб доступу: URL: http://ru.wikipedia.org/wiki/Анализатор_трафика – Загол. з екрана.

24 Социальная инженерия (Електрон.ресурс) / Спосіб доступу: URL: http://ru.wikipedia.org/wiki/Социальная_инженерия – Загол. з екрана.

25 Snort (Електрон.ресурс) / Спосіб доступу: URL: http://ru.wikipedia.org/wiki/Snort – Загол. з екрана.

26 DoS-атака (Електрон.ресурс) / Спосіб доступу: URL: http://ru.wikipedia.org/wiki/ DoS-атака – Загол. з екрана.

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітка
1	А4	Реферат	3	
2	А4	Список умовних скорочень	2	
3	А4	Зміст	2	
4	А4	Вступ	3	
5	А4	1 Розділ	34	
6	А4	2 Розділ	65	
7	А4	3 Розділ	11	
8	А4	Висновки	1	
9	А4	Перелік посилань	2	
10	А4	Додаток А	1	
11	А4	Додаток Б	1	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	

ДОДАТОК Б. Перелік документів на оптичному носії

1 Титульна сторінка.doc

2 Завдання.doc

3 Реферат.doc

4 Список умовних скорочень.doc

5 Зміст.doc

6 Вступ.doc

7 Розділ 1.doc

8 Розділ 2.doc

9 Розділ 3.doc

10 Висновки.doc

11 Перелік посилань.doc

12 Додаток А.doc

13 Додаток Б.doc

14 Додаток В.doc

15 Додаток Г.doc

Презентація.pptx

ДОДАТОК В. Відгуки керівників розділів

Відгук керівника економічного розділу:

Керівник розділу _____ _____

(підпис) (ініціали, прізвище)

ДОДАТОК Г. ВІДГУК

на кваліфікаційну роботу бакалавра на тему:
Розробка системи виявлення мережевих атак на інформаційну систему приватного підприємства "АрдКом"
Ярощука Владислава Володимировича

Пояснювальна записка складається з титульного аркуша, завдання, реферату, списку умовних скорочень, змісту, вступу, трьох розділів, висновків, переліку посилань та додатків, розташованих на __ сторінках та містить __ рисунків, __ таблиць, __ джерел та __ додатка.

Мета роботи: синтез комплексної системи виявлення мережевих вторгнень, що дозволяє мінімізувати ймовірності проведення вдалих атак на інформаційну систему приватного підприємства "АрдКом".

У спеціальній частині проаналізовано роботу мережі ПП «АрдКом», обрано та забезпечено профіль захищеності, проведено аналіз інформаційних загроз підприємства, запропоновано синтез комплексної системи виявлення мережевих вторгнень на інформаційну систему підприємства.

В економічному розділі виконано розрахунок вартості заходів щодо зниження ймовірності проведення атак, а також розрахунок збитку від атаки на обчислювальну мережу ПП «АрдКом». Надано оцінку економічної ефективності впровадження заходів щодо зниження ймовірності проведення атак, запропонованих для ПП «АрдКом».

Студент показав достатній рівень володіння теоретичними положеннями з обраної теми, показав здатність формувати власну точку зору (теоретичну позицію).

Робота оформлена та написана грамотною мовою. Містить необхідний ілюстрований матеріал. Автор добре знає проблему, уміє формулювати наукові та практичні завдання і знаходить адекватні засоби для їх вирішення.

В цілому робота задовольняє усім вимогам і може бути допущена до захисту, а його автор заслуговує на оцінку «_____».

Керівник