МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ КОМП`ЮТЕРНОЇ І ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА БЕЗПЕКИ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ О.Г. Корченко
«_____»_____ 20__ р.

На правах рукопису
УДК 004.056: 510.22 (043.2)

КВАЛІФІКАЦІЙНА РОБОТА
ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»

Тема: Модуль захисту приватної комп'ютерної системи від шкідливого програмного забезпечення

Виконавець: Т.П.Лук’яненко

Керівник: к.т.н., проф.: А.М.Давиденко

Нормоконтролер: О.О.Бурбела

Київ 2020

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки комп`ютерної і програмної інженерії
Кафедра безпеки інформаційних технологій
Освітній ступінь: «Магістр»
Спеціальність: 125 «Кібербезпека»
ОПП: «Адміністративний менеджмент у сфері захисту інформації»

ЗАТВЕРДЖУЮ
Завідувач випускної кафедри
д.т.н., проф. Корченко О.Г.
_____
«_____»_____2020р.

ЗАВДАННЯ
на виконання магістерської кваліфікаційної роботи
Лук'яненка Тараса Петровича

1. Тема: Модуль захисту приватної комп'ютерної системи від шкідливого програмного забезпечення

затверджена наказом ректора від №2067/ст від 19.10.2020 р.

2. Термін виконання 05.10.2020 по 31.12.2020

3. Вихідні дані роботи: *проаналізувати методи і засоби виявлення програмного шкідливого забезпечення інформаційних систем; розробити модуль захисту приватної комп'ютерної системи від впливу шкідливого програмного забезпечення; провести експериментальне дослідження.*

4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): аналіз методів і засобів виявлення програмного шкідливого забезпечення інформаційних систем. Розробити модуль захисту приватної комп'ютерної системи від впливу шкідливого програмного забезпечення. Провести експериментальне дослідження.

Календарний план
виконання магістерської кваліфікаційної роботи

№ п/п	Завдання	Термін виконання	Підпис керівника
1	Уточнення постановки задачі		
2	Аналіз літературних джерел		
3	Обґрунтування вибору рішення		
4	Збір інформації		
5	Аналіз методів і засобів виявлення програмного шкідливого забезпечення		
7	Розробка програмного модуля		
8	Експериментальне дослідження		
9	Оформлення і друк пояснювальної записки		
10	Оформлення презентації		
11	Отримання рецензії		
12	Підготовка до захисту в ДЕК		

Здобувач _____ Лук’яненка Т.П.
(підпис)

Керівник _____ Давиденко А.М.
(підпис)

РЕФЕРАТ

Дипломна робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел, додатків і містить --- сторінки основного тексту, --- рисунків, --- таблиці, --- сторінок додатків. Список використаних джерел містить --- найменування. Загальний обсяг роботи ---- сторінок.

Метою роботи є удосконалення приватної комп'ютерної системи від шкідливого програмного забезпечення, за рахунок розробки модуля, який відповідатиме за блокування шкідливих кодів.

Для реалізації зазначеної мети необхідно вирішити наступні задачі:

1. Провести аналіз методів і засобів виявлення шкідливого програмного забезпечення.
2. Розробити програмний модуль та алгоритм.
3. Провести експериментальне дослідження удосконаленої системи.

Об'єктом дослідження є процес захисту інформаційних систем.

Предметом дослідження є методи виявлення шкідливого програмного забезпечення.

В роботі запропоновано підвищити рівень захисту користувачів інформаційної системи за рахунок розробки модулю захисту від шкідливого програмного забезпечення.

ІНФОРМАЦІЙНА СИСТЕМА, РІВЕНЬ ЗАХИСТУ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, МОДУЛЬ ЗАХИСТУ, ШКІДЛИВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – база даних;

ЕОМ – електронна обчислювальна машина;

ІС – інформаційно-аналітична система;

ІР – інформаційні ресурси;

ІС – інформаційне сховище;

ІТ – інформаційні технології;

ОС – операційна система;

ПЗ – програмне забезпечення;

СОД – система обробки даних;

СУБД – система управління базами даних;

ШПЗ – шкідливе програмне забезпечення.

ВСТУП

На сьогоднішній день велика кількість компаній має робочі мобільні пристрої, які використовуються співробітниками в особистих цілях, що часто призводить до витоку конфіденційних даних підприємства. Тому мобільні девайси бізнес-організацій є найбільш привабливими для кіберзлочинців і потребують захисту в першу чергу. Однією з найважливіших задач інформаційної безпеки є боротьба зі шкідливим програмним забезпеченням (ШПЗ) і зокрема його виявлення. Ефективність управління в складних галузевих системах значною мірою визначається ефективністю реалізації процедур аналізу, обробки інформації та прийняття рішень [26]. Тому, **актуальною** задачею є підвищення ефективності виявлення шкідливого програмного забезпечення.

Метою роботи є розробка модуля захисту приватної комп'ютерної системи від шкідливого програмного забезпечення

Для реалізації зазначеної мети необхідно вирішити наступні **задачі**:

1. Проаналізувати методи і засоби виявлення шкідливого програмного забезпечення.
2. Розробити модуль захисту та удосконалену інформаційну систему.
3. Провести експериментальне дослідження.

Об'єктом дослідження є процес захисту інформаційних систем.

Предметом дослідження є модулі виявлення шкідливого програмного забезпечення.

Наукова новизна - удосконалено інформаційну систему за рахунок розробки модуля, що дозволить підвищити рівень захисту користувачів від шахрайських дій зловмисників.

РОЗДІЛ 1
АНАЛІЗ МЕТОДІВ І ЗАСОБІВ ВИЯВЛЕННЯ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Дослідження найпоширеніших видів шкідливих програмних впливів

Шкідлива програма - комп'ютерна програма або переносний код, призначений для реалізації загроз даним, що зберігаються в інформаційній системі, або для прихованого нецільового використання ресурсів системи, або іншої дії, що перешкоджає нормальному функціонуванню інформаційної системи [1, 2].

Комп'ютерні віруси — це спеціальні програми в машинних кодах або фрагменти програм, здатні без відома та згоди користувача розмножуватися й розповсюджуватися на інші програми шляхом копіювання свого коду у файли, що зберігаються в системі[1-3].

Найбільш поширеними видами шкідливих програм є трояни, черв'яки та віруси.

Для основного визначення, шкідливих програм, призначених для отримання несанкціонованого доступу до інформації, у вихідних існуючих правилах визначення доступу: санкціонований доступ до інформації (англ. Санкціонований доступ до інформації) - доступ до інформації, що не повідомляє правилу; несанкціонований доступ до інформації (англ. Несанкціонований доступ до інформації) - доступ до інформації, що повідомляє про правила розмежування доступу при використанні штатних засобів, що надають засобами обчислювальної техніки та автоматизованих систем. Під штатними засобами розуміється сукупність програмного, мікропрограмного та технічного забезпечення засобів обчислювальної техніки або автоматизованої системи [1-3].

Наприклад, функція шпигунського програмного забезпечення - це збір даних про активність машин, про криптоключі, паролі та критичні ініціативи та критичні критичні дані Він може потрапити до вас разом із вітальною листівкою. Якщо хочете, щоб один користувач відкрив таку листівку, вся система виявиться компрометованою. 80% даних шпигунського програмного забезпечення відсилає його господареві поштою (порт = 25), деякі різновиди цих програм містять у сервер. Витрати на протидію шпигунських програм збільшуються щорічно приблизно в п'ять разів. Сюди входять витрати на створення програми та оновлення бази даних сигналу шпигунських програм.

Симптомами наявності шпигунського програмного забезпечення у комп'ютері можуть бути такі ознаки:

• спонтанно відкриваються вікна;

• переадресують на сайті, відмінний від того, ім'я якого великого;

• у вікні браузера з'являється від іконки нового програмного засобу;

• з'явилася нова іконка у списку даних на нижній частині екрану;

• змінена базова сторінка браузера;

• змінено сторінку пошукової програми браузера;

• перестали працювати деякі клавіші у браузері (напр., Клавіша табулятора);

• з'являється незрозуміле повідомлення про помилки;

• комп'ютер несподівано уповільнює свою роботу (наприклад, під час упорядкування файлів тощо). [55]

СПАМ вимагає до 90% повного обсягу поштових повідомлень. Пов’язавши це з тим, що розсилка СПАМ стала прибутковою частиною нового кримінального бізнесу. Це стосується втрат мережевих ресурсів, про час надання послуг я вже не кажу. Частина таких повідомлень часто заражається вірусами, хробаками або троянськими кіньми. Я, наприклад, отримую до 300 таких повідомлень за день. У мене є спеціальний поштовий

ящик, який містить такі повідомлення. Ефективність сучасних фільмів СПАМ досягає 90%. До цього слід прослухати, що такі фільтри сильно завантажують сервери DNS. Для мінімізації СПАМ рекомендується рекомендувати декілька поштових адрес, наприклад, один для приватної переписки, один для ділової та один для інформаційних обмінів, передплати та ін Це полегшує розпізнавання СПАМ. Корисно самостійно створити унікальні адреси для кожного виду обміну та часу від часу їх зміни. Легше робити в разі підписних листів. Рекомендується видалити свою поштову адресу від свого веб-сервера. За даними Sophos 60% СПАМ розсилюється через зламані ЕОМ. СПАМ не лише дратує, що з'єднує мережеві ресурси, він робить бізнес менш ефективним. Уявіть скільки часу співробітники витрачають на перегляд таких повідомлень, скільки використовують і ресурси витрачають на придбання та програму обслуговування, фільтруючий пошту, і ви знаєте, що СПАМ НЕ шкідливий і вже сьогодні робить збитків більше значних, ніж мережеві віруси.

(У травні 2008 року ботнет Srizbi збільшив до 60 мільйонів спаму-повідомлень про добу.

"Пасивні" атаки за допомогою, наприклад sniffer особливості небезпечних, тому що, пер-перше, практично неструктуровані, по-друге, робляться з локальної мережі (зовнішній безпечний брендмауер).

Шахрайство - шахрайський трюк, який хотів би тому, щоб посилатися на авторитетних осіб, вступаючи. Першовідкривачами цього виду шахрайства були адреси з Нігерії. Починаючи з 2008 року, став активно використовувати шахрайський трюк, коли пропонується антивірусне (або будь-яка програма, що захищає засіб) при спробі завантаження машин, виявленої із зароблено шкідливим кодом. До того ж, що доступна для цієї програми, вона часто з'являється на платформі, що додає додаткову інформацію.

Ще одна компанія, яка користується новим загрозою, має міттєві повідомлення - за суто цією Інтернет-таблицею реле (IRC). Більша кількість системних IM (MSN, Yahoo IM, AIM та ін.) Має стандартні номери портфоліо, блокується доступ до цього сервісу. закрити ці порти, не можна, так як система може скористатися іншими номерами портів, надіслати. д .. До цього класу уразливостей слід віднести і сервіс ICQ. Можливо, деякі читачі стикалися з появою на екрані їх відображення вікон, що запрошували в "венхотство" Деякі користувачі можуть вважати, що вони знаходяться в безпеці, тому що не потребують електронної пошти. Для захисту від цього виду атак потрібно спеціалізоване програмне забезпечення. Слід також пам'ятати, що одного разу піклуються IM, не можна бути впевненим, що є цілим зухом. Наприклад, автоматичне оновлення Windows XP SP2 включає в себе завантаження Windows Messenger, що створює нанівець та зусилля.

До числа небезпечних програм PUP (потенційно небезпечна програма) можна віднести:

1. Утиліти, які є частинами руткіту (програма, яка є шкідливими і в тому часі практично неструктурована) або троянськими кіньми віддаленого доступу, модифікованими так, щоб приховати їх наявність.

2. Програми, які в момент створення були викликані полегшити адміністрування або обдурити систему безпеки для того, щоб дозволити віддалене адміністрування. До цього різновиду відносяться різні зломщики паролів, утиліти віддаленого управління тощо.

Ясно, що проводитимуть читання між шкідливими кодами, написаними молодими людьми антисоціального спрямування, і подібними програмами, створеними легальними фірмами, стає все важливим (а іноді це зробити просто неможливо). Прикладом програми останнього типу може служити утиліта "віртуального маркетингу". Зараз ви можете використовувати ресурси користувачів для отримання звітів без останнього повідомлення. Чи використовуємо автори шкідливих програм технології

віртуального маркетингу або розробників цієї технології, використовуємо досвід керівництва для досягнення своїх цілей, не цілком ясно. Очевидно, що і ті й інші ведуть війну за оволодіння ресурсами комп'ютера для витягання власного витягання власника.

Класифікація цієї програми використовує той факт, що багато їх шкідливих утиліт поєднують особливості. Наприклад, деякі різновиди троянських коней можуть самі розширюватись за допомогою електронної пошти, інші, як, наприклад, троянський кінь SpamThru, здійснює розсилку спаму та блокує роботу антивірусної системи. Якщо традиційний вірус є одним із виконуваних файлів (. Exe), така сучасна версія надає вбудовування різних програм та інформаційних модулів у кілька інших програм та конфігураційних масивів. Ці дії, крім іншого, сприяють вивченню факту зараження та передують видаленню шкідливих.

Шкідливі програми можуть використовувати наступні функції:

встановлення - вхід у систему та встановлення або модифікацію ділових програм.

обстеження - виявлення нових мішеней для атаки (характерно для вірусів).

копіювання - копіювання програм у задачі об'єктів-мішенів (характерно для вірусів).

впровадження - введення в код або дані нешкідливих вставок, які надають привілеї або передбачають привчання певних дій, або отримують деякі дані, усі для хакера.

транспортування даних - передача інформації про третій бік або отримання команди з третьої сторони [56].

Програми, які реалізують, хочуть, щоб одна з названих вище функцій могла бути сміливо віднесена до довірливих. Але і деякі інші особливості можуть відрізнятися від коду до категорії потенційно небезпечних.

Наприклад, прослуховування процесів, файлів, сервісів, регістрів або ключів, а також блокування роботи антивірусної програми Firewall, переадресація чи пересилання персональних даних. (Широка популярність файлообмінних мереж, що базуються на техніці P2P, а також IP-телебачення сервісу.

Як зазначалося в актуальності, мобільні оператори зараз фактично є системами для організації будь-яких платежів. Тому все частіше шахраї використовують телефон для того, щоб залізти в кишеню користувачів мобільного зв'язку. Способи для цього можуть бути найрізноманітнішими: від розсилки SMS-спаму до зараження троянцем смартфона або вимоги викупу за розблокування домашнього комп'ютера.

На сьогодні існуть низка ПЗ щодо виявлення шкідливого програмного забезпечення.

Kaspersky Security

Версія антивіруса від Лабораторії Касперського для пристроїв на базі операційної системи Symbian. Kaspersky Security 9.0 - комплексне рішення для захисту смартфонів від усіх типів інформаційних загроз, а також через вікно конфіденційних даних при втраті смартфона, яке не має своїх робітників за рівневу технологію та за допомогою безпрецедентного рівня захисту від різних загроз, що створюється перед користувачами смартфонів. Наприклад, за допомогою технологій SMS-Find Ви можете виявити точне місце знаходження заблудженого смартфона. Відіславши SMS із паролем на номер загубленого пристрою, користувач отримав відповідь на посилення.

Програма MobiShield

За допомогою даної програми для смартфонів Ви не можете лише забезпечити надійний захист від вірусів та шкідливих файлів у реальному часі, а також керувати процесами та завданнями, автозапусковими додатками, а також контролювати Інтернет-трафік з можливістю встановлення місячного ліміту. Крім того, MobiShield має ряд додаткових

інструментів, здатних суттєво поліпшити працездатність смартфона та запобігти несанкціонованому підключенню до мережі Інтернет. Утиліта очищення від застарілих та непотрібних файлів допомагає швидко вирішити проблему вільного місця в пам'яті телефону та на картах пам'яті, а вбудований фаверл забезпечує надійний захист від підключення до мережі сторонніх додатків.

Основні можливості:

• захист від вірусів та шкідливих програм у режимах реального часу;

• два режими сканування файлової системи із занесенням результатів у журнал;

• управління автозапуском додатків;

• управління запущеними процесами з можливістю централізації будь-якого процесу;

• вбудований деінсталятор додатків;

• вбудований міжмережевий екран (фаєрвол);

• огляд статистичних даних Інтернет-трафіку з можливостями встановлення ліміту

• менеджер мережевих підключень;

• утиліта для очищення файлової системи від застарілих та непотрібних файлів;

• регулярно оновлювані бази даних.

Програма F-Secure Security

● Дана програма для смартфонів, призначена для розповсюдження нецільового використання конфіденційної інформації, а також зарахування телефону шпигунським ПО, програми незапланованого перекладу платежів та інших шкідливих програм. Завдяки вбудованій функції віддаленого блокування та видалення у випадку крадіжок, а також механізму контролю, який дозволяє змінювати карти SIM-картки, надсилаючи відповідні відомості за допомогою SMS на вказаний номер SMS.

- Особливості F-Secure Security:
- • вбудований брендмауер, захист від вірусного та шпигунського ПЗ у реальному часі;
- • розширені протиугінні функції;
- • блокування та видалення даних у смартфонах за допомогою команди SMS;
- • повне блокування смартфона в разі зміни SIM-карти;
- • дворівнева система оновлень;
- • робота програми в фоновому режимі.

Dr.Web for Symbian OS

За допомогою програм для смартфонів Dr.Web для Symbian OS телефон надійно забезпечений антивірусним захистом від шкідливих об'єктів, Інтернет-загрози, Які спеціально створені для інфікування мобільних пристроїв, спаму. Програма є антивірусом для смартфонів Nokia на базі SymbianOS.

У можливостях програми також присутня фільтрація вхідних дзвінків та SMS повідомлень з використанням. Інтуїтивно зрозумілий інтерфейс дозволяє Налаштувати робочі програми для зручності її використання та встановити оптимальний рівень захисту мобільного пристрою.

Основні можливості Dr.Web для ОС Symbian:

• сканування всієї файлової системи або окремих файлів / папок за запитом користувача;

• сканування файлів та папок на карті пам'яті;

• видалення виявлених небезпечних об'єктів або зміна їх у карантині;

• фільтрація вхідних дзвінків та SMS-повідомлень на основі налаштованих чорного та білого списків;

• ведення деталізованого звіту про системи сканування;

• оновлення вірусних баз через Інтернет.

Програма NetQin Anti-Virus Pro

Відмінна програма-антивірус для смартфона Nokia. Антивірус NetQin Pro чудово справляється з вірусною загрозою - час сканування трохи більше, ніж у Касперського. NetQin може видалити інфіковані файли всередині sis-дистрибутивів, не чіпаючи інше. При спробі відкрити файл заробітку, а також при отриманні інфікованого повідомлення з'являється діалогове віконце з попереднім і записом про видалення.

Встановлення дозволяють встановити рівень захисту та автозапуск антивірусного монітора.

Особливості антивіруса NetQin Anti-Virus Pro:

• оновлення антивірусного базу вручну і за розкладом, прямо з програм;

• сканування SIS та SISX пакетів на сайті шкідливих об'єктів;

• перевірка ZIP, RAR та JAR пакетів на наявність шкідливих об'єктів

• дозволити вибору області сканування;

• сканування за розкладом;

• висока швидкість роботи;

Програма NetQin Anti-Virus Pro

Відмінна програма-антивірус для смартфона. Антивірус NetQin Pro чудово справляється з вірусною загрозою - час сканування трохи більше, ніж у Касперського. NetQin може видалити інфіковані файли всередині sis-дистрибутивів, не чіпаючи інше. При спробі відкрити файл заробітку, а також при отриманні інфікованого повідомлення з'являється діалогове віконце з попереднім і записом про видалення.

Встановлення дозволяють встановити рівень захисту та автозапуск антивірусного монітора.

Особливості антивіруса NetQin Anti-Virus Pro:

• оновлення антивірусного базу вручну і за розкладом, прямо з програм;

• сканування SIS та SISX пакетів на сайті шкідливих об'єктів;

• перевірка ZIP, RAR та JAR пакетів на наявність шкідливих об'єктів

• дозволити вибору області сканування;

• сканування за розкладом;

• висока швидкість роботи;

Програма ESET Security Beta

Антивірусна програма для смартфонів з операційною системою Symbian, випущена компанією ESET. Ця програма забезпечує розміщення в стадіях бета-тестування та доступності для користувачів.

Включає в себе такі нові можливості та функціональність:

• функція Анти-Злодій функцій;

• список довірених SIM-карт;

• дозволити віддалене зниження конфіденційної інформації, використовуючи видалену певну парку;

• повністю інтегрований міжмережевий екран, що збільшить кількість вхідних та вихідних повідомлень відповідно до заданих членів правилами та режимами;

• MMS / SMS антиспам: включає в себе чорний та білий списки, блокує небажані повідомлення;

• поліпшена євристика, оптимізована для використання на мобільних платформах;

• заражені файли можуть бути поміщені в карантин, з якого користувач може їх або відновити, або ви;

• перевірка на вимогу: сканує всі або конкретні папки;

• журнали та статистика;

• зручний інтерфейс, налаштування, інтуїтивно зрозуміле меню.

Програма Panda Security

Програма для смартфонів представляє одну з кращих програм у галузі безпеки для мобільних пристроїв на ринку, яка включає в себе об'єднання нових рішень.

Включає в себе такі можливості:

• на вимогу сканування файлів з різних дисків на мобільному пристрої: перевіряє, при виявленні.

• захист файлів та повідомлень.

• захист від бездротового зв'язку: фільтри і виявляє шкідливі програми, які можуть бути відірвані від відра

Программа BitDefender Security

Потужна антивірусна програма для смартфонів, що захищає телефон від шкідливих програм. Заснована на цінному досвіді BitDefender у боротьбі з комп'ютерними погрозами. Програма BitDefender Security легка та проста у використанні, що дозволяє їй смартфон бути завжди в робочому стані.

Включає в себе такі особливості:

• постійний антивірусний захист;

• сканування доступу до вірусів та видалення останніх;

• низьке розміщення ресурсів;

• швидке оновлення через GPRS або ПК;

• проста установка та використання.

Програма AntiVir

Хороший антивірус для смартфонів від відомого розробника системи безпеки німецької фірми Avira Gmb. Включає в себе такі переваги:

• великий список сигналів (базується опис 310 вірусів)

• євристичний аналіз

• можна включити / виключити постійну перевірку карти пам'яті

• малі витрати системних ресурсів

• малі витрати трафіку при оновленні базу

• звичайно, німецька надійність та якість

Програма Антивіруса Касперського для мобільних систем.

Антивірус Касперського - зручне та надійне рішення для захисту користувачів смартфонів від ШПЗ.

Віруси, черв'яки та троянські програми переставляють бути атрибутом лише миру персональних комп'ютерів - з ростом популярності смартфонів та комунікаторів (принципово від комп'ютера нічим, розміром, що не передбачається) з'являються через усі різноманітні шкільні програми для мобільних пристроїв .

Антивірус Касперського Мобільний об'єкт пропонує новітні розробки «Лабораторії Касперського» в області захисту мобільних платформ та багаторічного досвіду боротьби з шкідливим ПО [4]. Цей продукт надійно захищає смартфон та дані, які зберігаються в майбутньому, від шкідливих програм.

На основі проведеного аналізу можна зробити узагальнення за основними ознаками.

Таблиця 1.1

	Використання ресурсів	Запобігання несанкціон. підключення	Швидкість виявлення ШПЗ	Контроль вхідних SMS	Виявлення STUD
MobiShield	+	+	-	-	-
F-Secure Security	-	-	-	-	-
Dr.Web for Symbian OS	+	-	-	+	-
NetQin Anti-Virus	+	-	+	-	-
NetQin Anti-Virus Pro	+	-	-	-	-
ESET Security Beta	-	+	-	+	-
Panda Security	-	+	-	-	-
BitDefender Security	+	-	+	-	-
AntiVir	+	-	+	-	-
Kaspersky Anti-Virus	-	-	+	-	-

РОЗДІЛ 2
РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ ТА АЛГОРИТМ

2.1. Механізми виявлення шкідливого ПЗ

Розглянемо різні способи ідентифікації шкідливого коду, функціональні і, початки, хронологічні зв’язки між ними, їх технологічні та складні особливості [21]. З одного боку, багато з описаними тут технологіями та принципами актуальні не тільки в антивірусах, але і в більш широкому контексті системної комп'ютерної безпеки. Крім того, за межі статей залишилося багато важливих, але більше приватних технологій антивірусної індустрії, таких як розпакування упакованих програм або потоків сигнатурного детектування.

Найкраща технологія пошуку шкідливих програм була заснована на використанні сигналу - ділянку коду, однозначно ідентифікує ту чи іншу шкідливу програму. У міру того, як еволюціонували віруси, ускладнювали і розвивали технології їх виявлення. Всі ці просунуті технології - різні види «євристика» та «поведінкових аналізаторів» - можна узагальнальнити. Мова піде переважно про несигнатурні технології [32].

Узагальнюючи, можна виділити наступні способи збору даних для виявлення шкідливих програм:

1. Робота з файлом як з масивом байтів.
2. Емуляція коду програми.
3. Запуск програми в «пісочниці».
4. Моніторинг системних подій.
5. Пошук системних аномалій.

Способи перераховані відповідно до підвищення рівня абстракції при роботі з кодом. Під рівнем абстракції у даному випадку, коли відбувається повага, під яким кутом зору розглядається програма, яка виконується: як первинний цифровий об'єкт (дані в базі даних) як поведінка (більше

абстрактних значень з боку байту) або як сукупність ефектів у системі операцій (більше наслідок з поведінки) [40]. Приблизно з цього ж вектору йшло і розвиток антивірусних технологій: робота з файлами, робота з поділами через файл, робота з файлом через поділи, робота з самим середовищем - тому представлений список природних зображень описував збудованими та за хронологією.

Підкреслимо, що наведені способи - не просто відомі технології, скільки умовні етапи безперервного процесу розвитку технологій збору даних для виявлення шкідливих програм. Технології розвиваються і переходять один в один і той самий більший раз поступово, наприклад, емуляція може виявити ближче до простої роботи з файлом, якщо вона реалізується лише приватним, перетворює файл як загальний байт, або ж до "пісочників", якщо мова йде про повну віртуалізацію системних функцій [11]. Розглянемо ці способи докладно.

1. Зчитування файлів

. Незважаючи на те, що відбувається "архаїчний", він не має застави і так чи інакше використовується у всіх сучасних антивірусах - але вже не як єдиний і навіть не як основний, а просто як один із кількох.

2. Емуляція

Емулятор розбирає байтовий код програми в командах та кожній команді, що запускається у віртуальній копії корпоративної копії. Ми можемо допомогти захистити спостерігати за поведінкою програм, не встановлюючи під загрозою операційну систему та дані користувачів, що неминуче стають під час виконання програм у реальному середовищі.

3. Віртуалізація: «пісочниця»

Віртуалізація в тому випадку, коли вона використовується в «пісочницях», являє собою логідіцве. А саме: «пісочниця» вже працює з виконуванням у реальному середовищі програми, але все ще її контр.

Таким чином, засіб захисту, засвоєння віртуалізації описаного типу, працює вже не з файлу, але з поведінкою програм - проте все ще не із системою [8].

Механізм типу "пісочниця", так само як і емулятор, особливо не застосовується в антивірусах - головним тим часом, що в програмі реалізації він вимагає значного обсягу ресурсів. Антивіруси, що мають у складі "пісочниці", легко ідентифікуються для сучасної тимчасової затримки між запущеними програмами та початком її виконання (або - у разі успішної ідентифікації шкідливих програм - між її запуском та повідомленнями, отриманими від антивіруса, про її успішне виявлення) [6]. Вслуховуючись, що зараз проводять активні дослідження в області апаратної віртуалізаці..

Поки що рухається тип «пісочниці», є лише в кількох антивірусах.

4. Моніторинг системних подій

Моніторинг системних подій є більш «абстрактним» способом збору інформації для виявлення шкідливих програм. Якщо емулятор або «пісочниця» спостерігають за кожною програмою окремо, то монітор спостерігає за всіма програмами відразу допомогою реєстрації всіх подій, що відбуваються в операційній системі і породжених працюють програмами [14].

5. Пошук системних аномалій

Даний метод заснування на наступних позиціях:

- операційне середовище разом із усіма виконуватися в ній програмах - це інтегральна система;

- їй влади якесь «системне стан»;

- якщо в середовищі вищої шкідливий код, то система станцій є "нездоровою" і не відображається від стану "здорової" системи, в якій шкідливого коду немає.

Виявляючи з цих положень ми можемо судити про стан системи (і, відже, про можливу наявність у ній шкідливих програм), що здійснює його з

еталоном (для еталону, що приймається "здоровими" станційними системами) або аналізуючи сукупність окремих її параметрів.

Для ефективного виявлення шкідливого коду методом аналізу аномалій необхідно скласти аналітичну систему - на зразок експертної системи або нейронної мережі [20]. Виникає багато питань: як визначити "здоровий стан", оскільки він відображається від "нездорового", які дискові параметри можна відстежувати і як їх аналізувати? Унаслідок такої складності в даний час цей спосіб розроблений мало. Зачатки його можна виявити в деяких антірутках-утилітах, реалізованих на рівнях порівняння з певними зрізовими системами, взятих за еталон (застарілі утиліти PatchFinder Kaspersky Inspector) або окремих її параметрів (GMER, руткіт Unhooker) [5].

Аналітичний компонент

Складність алгоритму прийняття рішень може бути абсолютно будь-яким. .

1. Просте порівняння

Вердикт висвітлення результатів порівняння єдиного об'єкта з наявним зразком Результат порівняння бінарних («так» чи «ні»). Приклад: ідентифікація шкідливого коду за строго певної послідовності байту. Інший підручник, більше високорівнева: ідентифікація підозрілої поведінки програм за єдиним удосконаленим чином.

2. Складне порівняння

Вердикт висвітлення результатів порівняння одного або кількох об'єктів із відповідними зразками. Шаблони для порівняння можуть бути гнучкими, а результат порівняння - імовірнісним. Підготовка: ідентифікація шкідливого коду через одну з таких байтових сигнатур, кожна з яких задається нежиттєво (наприклад, так, що окремі байти не визначені). Інший приклад, більше високорівнева: ідентифікація шкідливого коду за кількома використаннями їх та викликуванням невідомих API-функцій з певними параметрами.

3. Експертна система

Вердикт виноситься в результаті тонкого аналізу даних. Це може бути система, яка містить у собі зачатки штучного інтелекту. Підготовка: ідентифікація шкідливого коду не для жорсткого завдання параметрів параметрів, але для результатів багатосторонньої оцінки всієї сукупності параметрів у цілому, з присвоєнням кожному з поділів ваги «потенційна шкільність» та розрахунку загального результату.

2.2. Удосконалена система

Ефективне зберігання інформації досягається наявністю в складі інформаційно-аналітичної системи цілого ряду джерел даних. Обробка і об'єднання інформації досягається застосуванням інструментів вилучення, перетворення і завантаження даних. Аналіз даних здійснюється за допомогою сучасних інструментів ділового аналізу даних.

Архітектура інформаційної (комп’ютерної) системи організації в узагальненому вигляді представлена на рис. 2.1.

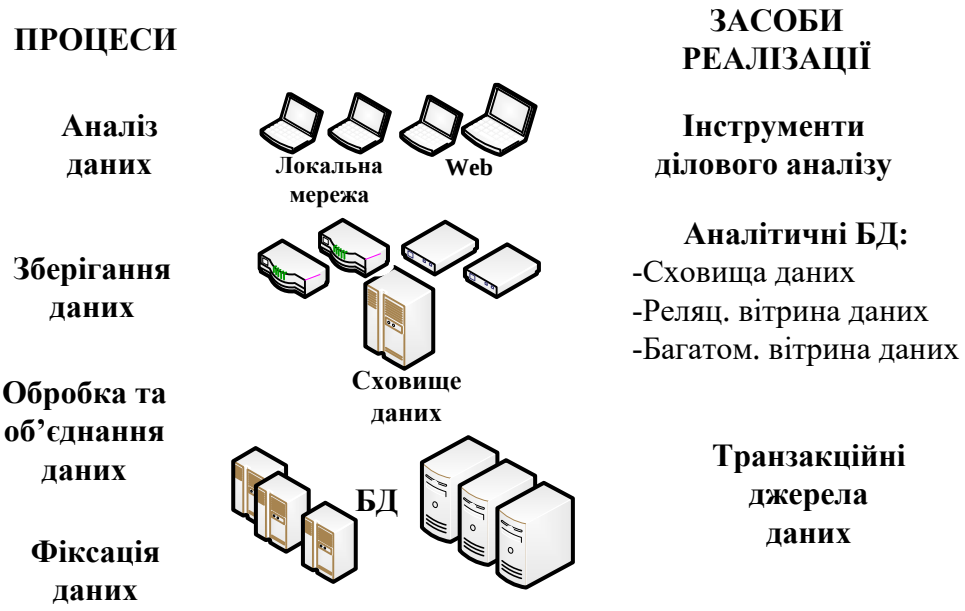


Рис. 2.1. Структура інформаційної системи.

Різноманітність джерел даних та необхідність їх використання у кожному конкретному випадку, пояснюється потребою по-різному зберігати інформацію в залежності від створених перед організацією завдань

Наведена архітектура демонструє довгий шлях, який надає дані, перед тим, як шукати на стіні.

Різноманітність джерел даних та необхідність їх використання у кожному конкретному випадку, коли пояснюється потреба по-різному зберігати інформацію в залежності від створених перед організацією завдань. Якщо ми класифікуємо дані джерел для типів і призначень, кожного з них можна умовно віднести до одного з трьох груп: дані про транзакційні джерела, дані про дані, вітрині дані.

Дані в системі можуть заноситися як вручну, так і автоматично. Транзакційні бази даних в організації можуть бути кількома.

Тому на наступному етапі вирішення завдання консолідації даних, їх перетворення та очищення, внаслідок чого дані надходять у так звані

аналітичні бази даних. Аналітичні бази даних, будь то схожі дані або вітринні дані, є основними джерелами, з яких аналітик черпає інформацію, використовуючи відповідні інструменти ділового аналізу.

До цієї інформаційно-аналітичної системи середнього та великого підприємства, організація якої створює користувачів, доступна для отримання аналітичної інформації, захищеної від несанкціонованого використання та відкриття як через внутрішню мережу організації, так і користувачів мереж Інтернет та Інтернет. Таким чином, структура системи визначає такі рівні:

1) збір і первинна обробка даних;

2) витяг, перетворення та завантаження даних;

3) складування даних;

4) подання даних у вітринах даних;

5) аналіз даних;

6) Веб-портал.

Розглянемо перераховані рівні архітектури та централізації на підставі типових інструментальних інструментальних інструментальних інструментів.

Збір та первинна обробка даних

До першого рівня структура ІС відносяться згадувані вже джерела даних, як правило іменовані транзакційними або операційними джерелами (базами) даних, які є частиною так званих OLTP-систем (online transactional processing). Транзакційні бази даних включають в себе джерела даних, орієнтовані на фіксацію результатів повсякденної діяльності організації. Вимоги, що пред'являються до транзакційних баз даних, зумовили їх наступні відмітні особливості: здатність швидко обробляти дані і підтримувати високу частоту їх зміни, орієнтованість, як правило, на обслуговування одного процесу, а не всієї діяльності організації в цілому.

Прикладами тут можуть служити бази даних, які використовуються в білінгових системах операторами стільникового зв'язку, в автоматизованих банківських системах комерційних і державних банків, в Інтернет-магазинах.

Інформація в таких базах даних орієнтована на конкретний додаток і управляється транзакціями, вона сильно деталізована і часто коригується.

Транзакційні бази даних відмінно справляються з валом повсякденної інформації, яка повинна рутинно оброблятися кожен день, але не дозволяють отримати загальну картину стану справ в організації в цілому і рідко можуть служити джерелами для проведення комплексного аналізу.

Отже, сукупність транзакційних джерел даних утворює нижня ланка архітектури інформаційно-аналітичної системи будь-якої організації. Надалі будемо виходити з того, що ІС підприємства будується на основі вже наявних на озброєнні систем збору і первинної обробки даних, що включають транзакційні джерела даних.

Витяг, перетворення і завантаження даних

Процес вилучення, перетворення і завантаження даних підтримується так званими ETL-інструментами (extraction, transformation, loading), призначеними для отримання даних з різних транзакційних джерел нижнього рівня, їх перетворення і консолідації, а також завантаження в цільові аналітичні бази даних – сховища даних і вітрини даних. На етапі перетворення усувається надмірність даних, проводяться необхідні обчислення та агрегування. Триступеневий процес вилучення, перетворення і завантаження повинен здійснюватися на основі встановленого регламенту.

Сховища даних

До третього рівня структури ІС відносяться джерела даних, які називають сховищами даних (від англ. Data Warehouse). Сховища даних включають в себе джерела даних, орієнтовані на збереження та аналіз інформації. Такі джерела можуть об'єднувати інформацію з декількох

транзакційних систем і дозволяють аналізувати її в комплексі з застосуванням сучасних програмних інструментів ділового аналізу даних.

Згідно з визначенням родоначальника ідеї складування даних Б. Інмона, сховище даних є предметно-орієнтованою, інтегрованою, некоректованою, залежною від часу колекцією даних, призначеної для підтримки прийняття управлінських рішень.

Характерними особливостями сховищ даних є: відносно рідкісне коригування більшості даних, оновлюваність даних на періодичній основі, єдиний підхід до пойменування та зберігання даних незалежно від їх організації у вихідних джерелах.

Сховище даних, будучи одним з головних ланок структури ІС будь середньої або великої організації, виступає в якості основного джерела даних для всебічного аналізу всієї наявної в організації інформації.

Представлення даних у вітринах даних

До четвертого рівня структури ІС відносяться джерела даних, звані вітринами даних (data marts), призначені для проведення цільового ділового аналізу. Вітрини даних будуються, як правило, на основі інформації зі сховища даних, але можуть також формуватися з даних, взятих безпосередньо з транзакційних систем, коли сховище даних в організації з якихось причин не реалізовано.

За типом зберігання інформації вітрини підрозділяються на реляційні і багатомірні. Вітрини першого типу організовуються у вигляді реляційної бази даних зі схемою "зірка", де центральна таблиця, таблиця фактів, призначена в основному для зберігання кількісної інформації, пов'язана з таблицями-довідниками.

Багатовимірні вітрини організовуються у вигляді багатовимірних баз даних OLAP (Online Analytical Processing), де довідкова інформація представляється у вигляді вимірювань, а кількісна – у вигляді показників.

Аналіз даних

До наступного рівня рівня ІС організація відновить сучасні програмні засоби, іменовані інструменти інтелектуального або ділового аналізу даних (інструменти бізнес-аналітики) або BI-інструменти.

BI-інструменти дозволяють керівництву організації організації проводити всебічний аналіз інформації, сприяти успішному приєднанню до великих обсягів даних, аналізувати інформацію, створювати основу аналізу об'єктів вісності та приймати рішення щодо обгрунтування, прогнозувати прогнози, зменшувати ризики прийняття неправильних рішень до мінімального мінімуму.

Інструменти інтелектуального аналізу даних використовують кінцеві користувачі для доступу до інформації, її візуалізації, багатозначного аналізу та формування як зумовлених для форми та складу, так і довільних звітів, що створюються керівництвом та аналітикою (без програми). Як вже було сказано, за якістю вхідної інформації для дилового аналізу виступають не просто "сирі" дані з транзакційних систем, скільки видаються збірних даних про схожі зображення або представлені у вітринах даних.

Веб-портал

В даний час компанії всі активніше починають впроваджувати у себе різні Інтернет-технології. Сьогодні сьогодні всі більше фахівців, які працюють не лише у сфері інформаційних технологій, починають розуміти вигоду від використання цих рішень у цілях підвищення ефективності свого бізнесу. Проведення інтелектуального аналізу даних із застосуванням програмних рішень не лише у місцевому середовищі, але і в середовищі Інтернет та Інтернету, відкриває аналітики нових можливостей роботи з даними. Сучасні тенденції розвитку архітектури інформаційно-аналітичної системи базуються на застосованому матеріалі застосованому Традиційний вид структури ІС у недавньому минулому допоміжний веб-портал, поступово набирає ведролм набирає дедролму Можливість доступу до інформації через звичний веб-браузер дозволяє економити на витратах, пов'язаних із закупівлею та підтримкою національних аналітичних програм для великого числа клієнтських місць. Реалізація веб-порталу дозволяє завантажувати аналітичну інформацію через користувачів у всьому офісі, а також мобільні користувачі-аналітики в будь-якій точці світу, підключені до порталу через Інтернет.

Вирішення задачі забезпечення користувачів інформацією в ІС визначається в основному правильним підбором інструментів ділового аналізу. Але важливим є і вибір інструментів підтримки процесів вилучення, перетворення, завантаження та зберігання даних.

При реалізації ІС підприємства можуть бути використані програмні рішення як різних фірм-виробників – змішані рішення, так і одного виробника – переносних базовані рішення. І в першому і в другому випадку є свої переваги і недоліки. Тому вибір інструментів для структури ІС, незважаючи на їх різноманіття, завдання не з простих.

Опис вдосконаленої модулем захисту системи

Рішення щодо забезпечення збереження конфіденційних даних та захисту від усіх видів вірусів, зокрема, через Alert, троянську програму, рекламні SMS та небезпечні дзвінків.

Основні можливості системи після вдосконалення її модуля захисту:

Захист від шкідливого коду STUD

.

Антивірусний захист на високій швидкості

Перевірка смартфона проводиться у фоновому режимі та не впливає на продуктивність операційної системи

Захист від крадіжки

За допомогою крадіжок телефону Ви можете визначити місцезнаходження злочинців за координатами GPS, віддалено заблокувати структуру та видалити всі контакти, фотографії та дані особи в майбутньому

Блокування SMS / MMS / викліків

'

Захист дітей

За допомогою налаштувань правил викликів та SMS повідомляє батьків, що можуть контролювати спілкування дитититиритітити Крім того, за даними GPS можна визначити місце розташування дитини в будь-який час

Ключові особливості:

Поведінковий аналіз загроз

Попередньо про будь-яку підозрілу активність додатків у телефонах. Виправлення знову з'явилися загроз, а також перевірка всіх програм, файлів, папок і карти пам’яті.

БлокВор

Багаторівнева система захисту даних, завдяки якій можна контролювати і запобігати дії шахраїв удосконалення

Надійний номер

Контакт, у разі втрати смартфона, буде надіслано повідомлення про те, що в режимі було всрановано смартфоном. За необхідності використання даної функції можна відновити захист пароля для виявлення рішень ESEN.

Управління викликами

Блокування небажаних вхідних та вихідних дзвінків. Ця функція дозволяє дозволити захищати користувачів від непотрібних дзвінків, а також вводити нерозподілені рахунки для додаткових послуг операторів стільникового зв'язку.

SMS / MMS Антиспам

Блокування небажаних SMS та MMS-повідомлень. Користувач може самостійно набудувати «чорні» та «білі» списки контактів, захищаючи себе.

Блокування прихованих номерів (CLIR)

Користувач може ігнорувати виклики абонентів, номери яких визначаються як «номер закріпленого».

Захист від видалення

Нова особливість ESET NOD32 Security - самозахист програм. Зловмисник не може легко видалити дані додаток без знання певних комбінацій, заданих членом.

Вбудований аудит безпеки

За результатами перевірок мобільного телефону користувач надає повну інформацію про стан системи, що включає дані про запущені додатки, зарядні батареї, Bluetooth, вільну пам'ять та підключені пристрої.

Віддалене блокування

Користувач може заблокувати смартфон за допомогою команд, переданих по SMS з будь-якого телефону, таким чином, обмеживши доступ до особистих даних на мобільному пристрої в разі втрати.

Безпека відстані

При втраті мобільного пристрою можна бути впевненим у безпеці особистих даних. Відправляючи SMS-команду з будь-якого телефону, можна видалити не лише контакти на телефоні, а також інформацію

Новий зручний інтерфейс

Швидкий доступ до всіх функцій та налаштування програми програми відповідно до власних уподобань кожного.

GPS-виявлення

За допомогою спеціальної SMS-команди можна визначити місце розташування телефону.

Системні вимоги:

- Вимога до дозволу екрана: від 320x480px і вище
- Процесор: 600+ Mhz
- Оперативна пам'ять: 128+ МБ
- Внутрішня пам'ять: 5+ МБ
- SD-карта: обов'язково
- Наявність Інтернет-підключення для оновлення

Операційні системи

- Android 2.0 і вище

- Symbian S60 3rd Edition Feature Pack 1 або 2 (тільки Nokia)
- Symbian S60 5th Edition (Nokia)
- Symbian 3 (Nokia)
- Windows 5.0, 6.0, 6.1 та 6.5

Вільна пам'ять

1 МБ

2.3. Розробка програми модуля захисту для виявлення ШПЗ

З урахуванням викладеного, актуальною науковою задачею, що має практичне значення, є розробка модуля захисту у складі ІС, що дозволяє дозволити забезпечити цілісність, доступність та захищеність інформації, яка зберігається та обробляється в ІС. Метою вдосконалення ІС є захист інформаційних ресурсів (ІР) від шкідливих програм, за рахунок розробки додаткових модулів, які відповідають на виявлення ШПЗ.

У конкретних реалізаціях окремих компонентів цієї схеми може бути відсутність. До такої організації ІС СД функціонує наступний сценарій: для заданого регламенту у збірці даних із різних джерел - система БД оперативної коробки. У СД підтримується хронологія: дані із поточних даних зберігаються в історичних даних із зазначенням часу роботи. У результаті необхідний доступ до даних про об'єкт управління збирається в одному місці, приєднуючись до єдиного формату, узгоджуючись і в ряді випадків, агрегуючись до мінімально необхідного рівня узагальнення.

Незважаючи на те, що СД створює інформацію про інформацію, яка є у базі даних або файлах оперативних систем, з'являється концепція СД, тимчасово аналізується інформація про оперативні системи, що неможливо або дуже важливо. Це

• розрізненість даних (OLTP-системи, текстові звіти, xls-файли);

• зберігання їх у форматах різних СУБД та у різних вузлах корпоративної мережі. Але навіть якщо у підприємстві дані зберігаються на

центральному сервері BD (що буває вкрай рідко), аналітик майже напевно не розбереться в їх складних, в часі додаткових структурах;

• складні аналітичні записки до оперативної інформації, що забезпечують потокову роботу компанії, надовго компанії, надовго компанії, надовго компанії, надовго.

Можна встановити, що практично в будь-якій організації складе парадоксальну ситуацію - інформація про початок, де і є, її дуже багато, але вона не має структурованості, неузгоджена, розрізна, не завжди достовірна, її практично неможливо знайти і отримати. У результаті можна говорити про відсутність інформації про її доступність і навіть велику кількість.

Для того, щоб вдосконалити ІС та забезпечити безпеку ІР, що в жодному обробляються, реалізованому захисному модулі, який виявляє та ліквідує ШПЗ, яке забирає в середину систем. Модуль представляє свого власного сканера, який працює, як сигнатурна антивірусна програма. Під час завантаження файлу в систему, прогресивний рівень збору, очищення та зведення даних із зовнішніх джерел, він надходить до розробленого модуля, який проводить над ним послідовність дій. Запускається база сигналів, після чого файл відкривається та перевіряється при встановленні зловмисних сигналів порівняння з тими, що є у базі сканера. Якщо сигналу виявлено, тоді з'явиться запит на відображення, який буде передано в карантинний файл, якщо сигнал не буде виявлений, цей файл буде передано далі, до подібних даних та наступного файлу, який можна сканувати.

Таким чином, навіть якщо ШПЗ потрапить в ІС, воно не може бути спроможне забезпечити зручне використання інформації про всі системи, а також, як буде виявлено та ліквідовано за допомогою маркування та інтегрованого модуля захисту. За відсутності конкретного вказівки коду вірусу, або якщо сайт не розміщений на сигналізації, діагностика проводиться для такої схеми, від простого до складного:

1.1. У вихідному коді HTML-сторінки шукаємо входження до слів "iframe" та "javascript". . Особливим підозрілими є iframe малої або нульової ширини і висоти, а javascript - із використанням eval, unescscape, String.fromCharті. У javascript особливу увагу слід звернути на document.write із записом іншого javascript або iframe, або введення мета-редиктатротротектро У деяких випадках вірусний код маскується під лічильниками вірогідності. Іноді проводити повну заміна обфускової бібліотеки JavaScript з використанням jquery на так ж, але містить вірус. У таких випадках потрібно звільнити розмір активної бібліотеки з розмірами того ж файлу в доступній формі.

Якщо в iframe, javascript або в редирект фігурує будь-який чужий домен (не і не розміщено там Вамі) - це сигнал тривоги, навіть якщо на домені пусто або там нормальний сайт. Віруси дуже часто йдуть "матрьошкою", коли реальне шкідливе вміст вискакує лише на третьому чемекоті.

1.2. Проводимо таке дослідження під час завантаження зовнішніх файлів javascript. У зовнішніх css проводимо пошук поведінки, що відображає чужий код.

1.3. Якщо на сайті є картинки, завантажені з інших сайтів - перевіряємо, що видається за запитом. За допомогою цього реферatора та агента він повинен бути як звичайно відкритою сторінкою веб-сайту за цією картинкою. Якщо замість картинок буде видалено редирект, запишіть пароль або інший чужий вміст - це як проріло.

1.4. Перераховані в пп. 1.1-1.3 дії потрібно виконати із записом сторінок та скриптів декількох розділів, в ідеалі з різними ір з різними файлами cookie та різними користувацькими агентами (броузерами), оскільки вірусний код може приєднатися до випадкових людей або лише тимчасовим браузером, який вирішує, або лише пошуковим запитом. або по іншому критерію.

1.5. Додаємо веб-сайт на Яндекс-веб-майстрах та веб-майстрах Google, у деяких випадках ці сервіси дають покази конкретного шкідливого коду або доменів, які під час завантаження вірусу.

1.6. Якщо ви перейшли на заробітну плату з включеним javascript у браузері (чого взагалі краще не робити), то антивірусна програма може дати список загроз, які було виявлено при оцінці веб-сайту. З цих даних також можна виділити список вірусних доменів.
1.7. Дівимося коди http-відповіді сервера на тему редиректів різних користувацьких агентів і з різних ip-адрес, що часто редиректується, видаляється випадковим випадком або заблокуванням. Іноді вірус веде щоденник і видає редирект або потрапляє лише один раз за кожному рейтингу.

2. Видалення вірусу

Знання, який саме код вірусу видає веб-сайт, допомагає знайти серверні джерело проблем. Якщо в ході діагностики видається оцінка шкільного коду, не було конкретизовано - не біда, очищення може бути успішно проведено і без цього, просто буде надано складніше.

2.1. Викачуємо себе на локальному комп'ютері всіх файлів веб-сайту, робимо резервну копію перед перевіреними повідомленнями перед перевіряючими.

2.2. Проводимо повнотекстовий пошук (за самими файлами, а не лише за їх заголовками), шукаємо вхідне первинне походження. 1.1-1.3 і знайдених в пп. 1,5 і 1,6 вірусних доменів. Альтернативний варіант - вести пошук прямо на сервері спеціальним серверним скриптом.

2.3. За допомогою команди ssh або серверного скрипта ми можемо знайти на серверах всі файли веб-сайту, які були змінені в день зарахування веб-сайту та вивчили їх за темою зовнішніх небезпечних доповнень. Це може бути:

• включати файли з вірусних доменів (не залежно від того, якщо дозволений віддалений включати за даними phpinfo),

• eval отриманих з інших сайтів даних,

• eval декодований функцією base64_decode даних,

• обфустірованний php-код,

• перевизначені функції,

• включати або видаляти зовнішні дані, що передаються скриптом через глобальні масиви GET, POST, COOKIE, SERVER ('HTTP_REFERRER_ є' HTTP_USтй, азвоог_USтй,

• сторони коди посилальних бірж (частота обслуговування веб-сайту продається за посиланнями),

• http-заголовки з повторного виправлення вірусних доменів, що відображає функцію заголовка,

• exec, system, popen, passthru та інші функції, які використовують виконувану програму, якщо їх використання не передбачено cms. Якщо cms не надає даних функції, а також функцію eval, щоб краще взагалі їх відключити в php.ini,

• бекдорі в тригерах mysql,

• auto_prepend_file або auto_append_file у php, з бекдорів або вірусним кодом,

• у дуже рідкісному випадку команда запуску лежачого у tmp вірусному файлі запускає користувачів crontab.

При аналізі небезпечних доповнень може допомогти знання коду, поява в ході діагностики (с.1).

Крім видалення оцінювачів шкідливого вмісту, перераховані вище чуттєві вхідні дані можуть бути створені через веб-бекдор-оболонку за допомогою якого ліходій контролює веб-сайт.

2.4. Робімо дамп бази даних, і вивчаємо аналогічно п.1.1, але з урахуванням того, що в брандмауері може бути банетично

2.5. Виділяємо всі чужі вихідні, виявлені в ході роботи за перерахованими вище пунктами.

2.6. Перевіряємо працездатність веб-сайту, його функціонал. Іноді вірус затирає себе важливими файлами або змінює їх синтаксис, і після очищення необхідного тривання необхідного тренування обов'язкового тренування обов'язкового тренування обов'язкового тренування. У дуже рідкісних випадках вірус затирає все так, що файли веб-сайту вже не пов'язані. Добре, якщо є копія в хостерах або підключена послуга резервного копіювання.

2.7. Створюємо резервну копію очищеного сайту. У разі повторного зарахування можна відновити сайт із цієї резервної копії.

Якщо орієнтуватись на цей наступний день або в п’ятницю введення на цій сторінці. Тому потрібно рухатися далі.

1. Для того, щоб дізнатись, хто з антивірусів може бути краще, ніж їхні обов'язки, можна дійсно. Найвідомішою лабораторною тестуванням є Вірусний бюлетень, який регулярно проводить випробування. При цьому відбираються шкідливі програми з WildList, колекція «диких» вірусів, які потім «атакують» тести. Проте результати цих випробувань найчастіше можуть бути недоступними, наприклад, грудне випробування 2007 року провалило всі російські антивіруси, наприклад, продукт Microsoft Forefront від корпорацій успішно пройшов його. Стадія до результатів неоднозначного і у самих виробників антивірусів - за словами Бориса Шарова, генерального директора компанії Dr.Web - "Особливість того, що інші антивіруси

виявляють на 100% від цієї колекції, що залежать від значущого рівня доступності відповідних членів сім'ї. Відповідальної антивірусної компанії ». Звичайні користувачі також не дуже схильні довіряти подібного роду тестування, враховуючи, що високі назви та нагороди можуть бути пов'язані з підтасовками та невід'ємним судочинством.

2. Довідкове тестування чи ні, як спровокувати випробування - для більшості з нас це залиється таємниця. Повідомлення при абсолютно різних умовах може призвести до трактування за різним і, мабуть, людський фактор може впливати на дослідження самого високого роду.

3. Тим часом кожен просунутий член, сисадмін або програма може самостійно провести тестування. Таке дослідження може виявити сумніви, по краєвих мережах, на великих підприємствах. Розглядом методики Звернення до антивірусних лабораторій. Провідні виробники, такі як Антивірус Касперського, Доктор Веб мають російські корені, так що, по крайней мере, мовного бар'єру при зверненні не виникає. Інші компанії, такі як ESET NOD 32, мають свої представництва в Інтернеті, так що їм можна просто взяти і написати. Однак з корпоративних міркувань антивірусні лабораторії вельми неохоче діляться зразками вірусів. Вони точно не захочуть мати справу з фізичними особами, і вкрай малоймовірно, що вони нададуть зразки за зверненням юридичної особи. Так, на своїх сайтах техпідтримки лабораторії люблять вивішувати назви актуальних загроз, але як справа доходить до відправки файлів цих самих загроз, всі стають надто впертими. Проблема також ускладнюється тим, що відправка зразків вірусів по електронній пошті потрапляє під статтю Кримінального Кодексу при його формальному тлумаченні.

4. Самостійне збирання колекції або залучення спільноти однодумців. Це один з найкращих варіантів – справді, якщо в Інтернеті гуляють віруси, найкраще їх «нацеплять» а потім використовувати в наукових цілях. Поставивши перед собою таке завдання, можна збирати

віруси і з флешок. Включаємо відображення прихованих і системних файлів, при вставці USB-пристрої утримуємо клавішу «Shift» і, відкриваючи через праву кнопку (не подвійним клацанням!) Диск, архівуємо файл вірусу, причому з міркувань безпеки краще встановити при цьому також пароль.

Таким же чином упаковуються віруси, які збираються в Інтернеті. Однак цей спосіб має істотний недолік – збір зразків здійснюється вкрай повільно. Бази вірусів тестових лабораторій Virus Bulletin оперують мільйонами зразків Зрозуміло, що ручний пошук – навіть за допомогою спільноти навряд чи дозволить зібрати достатню колекцію в розумні терміни. Втім, таким чином можна знайти зразки найбільш актуальних загроз.

6. Пошук готових архівів. Готові бази вірусів, які містять сотні та тисячі розрізів, можна знайти на хакерських сайтах, в омновонах. Як правило, це безкоштовні архіви, тоді, зустрічаються та збираються продажі. Після того, як архів отриманий (або куплений), можна повідомити його вміст із відомими класифікаціями. Встановивши таким чином «паспортні дані» зразків, можна наблизитися до самого тестування.

В основі роботи будь-якого антивірусу лежить євристичний алгоритм пошуку зловмисних дій і дестротрон. Саме цей алгоритм визначає «інтелектуальність» цього антивірусного пакету. Однак творчі віруси постійно вдосконалюють маскування та способи проникнення, тому євристичного. Крім того, евристичний аналіз часто призводить до помилкових спрацьовувань, коли корисні проюрами варісньсіачами.

Для підвищення ефективності майже з самого початку створення антивірусних пакетів слід звернутися до співробітників. Й Фактично програма звіряє поточні файли із зразками (сигнатурами), наявними в базі. При виявленні подібності та появі визначення вірусу.

Для сигнатурного методу велике значення мають актуальні антивірусні бази. Старіння базу (оновлюваних через Інтернет) упродовж

кількох днів, часто і протягом години, може приносити до того, що комп'ютер виявить хильний до зараження нової вірусної епідемії.

Євристичний модуль закладається в антивірусний пакет на етапі створення програм. Оновлення антивірусної бази кожного користувача робить самостійно, після встановлення та придбання програм.

Піратські версії антивірусів часто можуть іноді оновлюватися. Але виробники (особливо російські) дуже оперативно реагують на підключення до своїх серверів неліцензійних програм і дуже швидко блокують їх доступ. Крім того, можна змінити, що вносять хакерами в антивірусні пакети, можна змінити їх функціональністрансанкиональністра Тому піратські копії ми розглядаємо не будемо.

Найкращим варіантом тестування є ліцензійні антивірусні пакети з актуальними оновленими на mtementv. Проте, після того, як буде використана необхідність використання ліцензій до достатньо великої програми, яка відповідає частині

Бюджетним і, мабуть, самим розумним варіантом є використання пробних (пробних) версій програми. Зазвичай вони можуть працювати протягом одного місяця (30 днів). Антивірусні бази, що входять до складу власних версій, як правило, північної давності. Потім для окремих випробувань можна долучитись, навпаки, найсвіжіші віруси.

2.4. Експериментальне дослідження

Проводячи дослідження розробленого нами модулю, ми проведемо просте тестування його на справність роботи, та здатність до виявлення ШПЗ. Таким чином, створюємо на диску С папку, в яку розміщуємо файл з вірусом. Це текстовий файл і вписуємо в нього сигнатуру STUD: STUD-Worm. HCF2R

Зберігаємо файл та відправляємо його на мобільний телефон посередництвом USB кабелю. Антивірус, що встановлений на телефоні не вдосконалений

модулем захисту. Запускаємо сканування пристрою, і спостерігаємо, що нічого не відбувається.

Потім, видаляємо тестовий файл та антивірус. Встановлюємо вдосконалений антивірус і повторюємо попередні дії, запускаємо сканування і бачимо, що в цей раз STUD виявлено і ліквідовано.

Зараз, на основі проведеного в першому розділі аналізу інших існуючих продуктів, можна провести порівняння для визначення переваг вдосконаленої системи.

Головні критерії:

1. Інтерфейс.
2. Споживання ресурсів.
3. Швидкість сканування.
4. Кількість виявлення.
5. Виявлення STUD.

Схема тестування:

1. Установка продукту.
2. Оновлення баз.
3. Перевірка каталогу з підбіркою шкідливих програм.
4. Перерахування переваг та недоліків продукту.

Вдосконалена система STUD.

Дане рішення забезпечує збереження конфіденційних даних і захист від всіх видів вірусів, в тому числі від STUD, троянських програм, рекламних SMS і небажаних дзвінків.

Основні можливості системи після вдосконалення її модулем захисту:

- Захист від шкідливого коду STUD
- Антивірусний захист на високій швидкості
- Низьке споживання ресурсів

- Запобігання несанкціонованого підключення

- SMS / MMS Антиспам
- Блокування прихованих номерів (CLIR)
- Зручний інтерфейс

Таблиця 3.1

	Використання ресурсів	Запобігання несанкціон. підключення	Швидкість виявлення ШПЗ	Контроль вхідних SMS	Виявлення STUD
MobiShield	+	+	-	-	-
F-Secure Security	-	-	-	-	-
Dr.Web for Symbian OS	+	-	-	+	-
NetQin Anti-Virus	+	-	+	-	-
NetQin Anti-Virus Pro	+	-	-	-	-
ESET Security Beta	-	+	-	+	-
Panda Security	-	+	-	-	-
BitDefender Security	+	-	+	-	-
AntiVir	+	-	+	-	-
Kaspersky Anti-Virus	-	-	+	-	-
Вдосконалена система STUD	+	+	+	+	+

В результаті проведеного порівняння можна зробити висновок, що вдосконалена ІС, за рахунок розробленого модуля система має значні переваги перед існуючими продуктами, основною з яких є виявлення шкідливого коду STUD/

ВИСНОВКИ

1. Проаналізовано методи і засоби виявлення шкідливого програмного забезпечення, що дало можливість визначити переваги, недоліки та особливості використання.

2. Розроблено модуль захисту та удосконалену інформаційну, що дало можливість підвищити рівень захищеності приватної інформаційної системи за рахунок блокування ШПЗ.

3. Проведене експериментальне дослідження дає зробити висновок, що розроблений модуль може виявляти тільки шкідливі програми, що поширюються цільним файлом, тобто не заражають інші файли. Так ШПЗ, якщо проникне в систему, то знешкоджується за допомогою захисного модулю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Касперский Е. В. Компьютерные вирусы в MSDOS. М.: Эдель, 1992. — 175 с.

2. Касперский Е. В. Компьютерные вирусы, что это такое и как с ними бороться. М.: Издательство «СК Пресс», 1998. — 288 с.

3. Касперски Крис. Азбука хакера 3. Компьютерная вирусология. М.: Майор, 2006. — 512 с.

4. Касперски Крис. Записки исследователя компьютерных вирусов. СПб: Питер, 2005. — 316 с.

5. Бобряков А.В., Титов В.Л. Региональные информационно-аналитические системы: опыт разработки и внедрения //Энергонадзор и энергосбережение. 2000. – № 3

6. Парфенов В. И. Защита информации: Словарь. Воронеж: НП РЦИБ «Факел», 2003. — 292 с.

7. Бобряков А.В., Гурфинкель Л.М., Перейма В.И., Тихонов В.А. Автоматизация работ Управлений Госэнергонадзора: цели, задачи, перспективы // Энергонадзор и энергосбережение сегодня. Спецвыпуск. 2001

8. Бобряков А.В., Титов В.Л. Региональные информационно-аналитические системы: сбор и обработка первичной статистической информации // Энергонадзор и энергосбережение сегодня. 2000. – № 4

9. Введение в OLAP и многомерные базы данных. – Михаил Альперович;

10. Бобряков А.В., Титов В.Л., Тихонова Е.А. Информационно-аналитическая система «Энергоэффективность»: разработка и внедрение подсистемы «Учет и анализ несчастных случаев на энергоустановках» //Энергонадзор и энергоэффективность. 2003. – № 2

11. Боркус В. Web-сервисы в корпоративной среде // PcWeek/RE. - 2004, -№27-35

12. Никишин, А. Эволюция вирусов и антивирусов. Эпохи DoS и Интернет / А. Никишин // CNews-аналитика. - 2006. ---№ 8. - С. 36-38.

13. Боэм Б., Каспар Х. Характеристики качества программного обеспечения Пер. с англ. /Под ред. Е.К. Масловского. -М.:Мир, 1981

14. Боэм Б.У. Инженерное проектирование программного обеспечения: Пер. с англ. /Под ред. А.А. Красилова. -М.: Радио и связь, 1985

15. Симонович, С.В. Информатика. Базовый курс / С.В. Симонович. -- СПб.: Питер, 2004. -- 640 с.

16. Виноградов В.И. Информационно-вычислительные системы: распределенные модульные системы автоматизации. -М.: Энергоиздат, 1985

17. Волкова В.Н., Кузина Б.И. Информационные системы. -СПб.: Изд-во СПбГПУ, 2004

18. Бобряков А.В., Стефанцов А.Г. Подход к проектированию информационных систем с учетом изменчивости внешней среды // Международная конференция «Информационные средства и технологии», Доклады международной конференции, М:Янус-К, 2004

19. Хомоненко, А.Д. Основы современных компьютерных технологий / А.Д. Хомоненко. - М.: Корона принт, 2009. ? 448 с.

20. Гламаздин Е.С., Новиков Д.А., Цветков А.В. Управление корпоративными программами: информационные системы и математические модели.-М.:ИПУ РАН, 2003

21. Острейковский, В.А., Полякова, И.В. Информатика. Теория и практика / В.А. Острейковский, И.В. Полякова. - М.: Оникс, 2008. -- 608 с.

22. Дмитриев А.К., Мальцев П.А. Основы теории построения и контроля сложных систем. -Л.: Энергоатомиздат, Ленинградское отд-е, 1988

23. Дмитриев А.К., Мальцев П.А. Основы теории построения и контроля сложных систем. -Л.: Энергоатомиздат, Ленинградское отд-е, 1988

24. Соболь, Б.В. Информатика / Б.В. Соболь. - Ростов-на-Дону, Феникс, 2007. -- 446 с.

25. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. -М.: Финансы и статистика, 2002

26. Кузнецов Н.А., Кульба В.В., Ковалевский С.С., Косяченко С.А. Методы анализа и синтеза модульных информационно-управляющих систем. -М.: ФИЗМАТЛИТ, 2002

27. Когнитивный анализ и управление развитием ситуаций (СА8С'2001)//Материалы 1-й Междунар. конф: в 3т./Сост. В.И. Максимов. - М.:ИПУ РАН, 2001

28. Фигурнов, В.Э. IBM PC для пользователя / В.Э. Фигурнов. - Уфа: НПО «Информатика и компьютеры», 2007. - 480 с.

29. Зегжда Д.П. Общая архитектура систем обнаружения вторжений // Проблемы информационной безопасности. Компьютерные системы. – 2001. – № 4. – С. 100-110.

30. Зегжда Д.П. Общая архитектура систем обнаружения вторжений // Проблемы информационной безопасности. Компьютерные системы. – 2001.– № 4. – С. 100-110.

31. Компьютерная поддержка сложных организационно-технических систем/ В. В. Борисов, И. А. Бычков, А. В. Дементьев, А. П. Соловьев, А. С. Федулов. М.: Горячая линия -Телеком, 2002

32. Ларичев О.И., Мошкович Е.М. Качественные методы принятия решений. -М.: Наука. Физматлит, 1996

33. Кузнецов Н.А., Кульба В.В., Ковалевский С.С., Косяченко С.А. Методы анализа и синтеза модульных информационно-управляющих систем. -М.: ФИЗМАТЛИТ, 2002

34. Ларичев О.И. Наука и искусство принятия решений. -М.: Наука, 1979

35. Когнитивный анализ и управление развитием ситуаций (СА8С'2001)//Материалы 2-й Междунар. конф: в 2т./Сост. В.И. Максимов. - М.:ИПУ РАН, 2002

36. Лачинов В.М., Поляков А.В. Информодинамика или путь к миру открытых систем.-СПбГТУ, 1999

37. Леман М.М. Программы, жизненные циклы и законы эволюции программного обеспечения ТИИЭР, Техника программного обеспечения: Пер с.англ. -М.:Мир, 1980, т.68, №9

38. Липаев В.В. Выбор и оценивание характеристик качества программных средств. Методы и стандарты Серия «Информационные технологии». -М.: Синтег, 2001.

39. Липаев В.В. Выбор и оценивание характеристик качества программных средств. Методы и стандарты Серия «Информационные технологии». -М.: Синтег, 2001.

40. Оперативная аналитическая обработка данных: концепции и технологии. – Л.В. Щавелёв, Ивановский государственный энергетический университет;

41. Липаев В.В. Системное проектирование сложных программных средств для информационных систем Издание второе, переработанное и дополненное. Серия «Управление качеством». -М: Синтег, 2002

42. Макаров В. Ф. Основные методы исследования программных средств скрытого информационного воздействия // Безопасность информационных технологий. – 2009. – № 4. – С. 11-17.

43. Нестерук Г. Ф. О применении нейронечетких сетей в адаптивных системах информационной защиты // Нейроинформатика-2005: Материалы VII всероссийской научно- технической конференции. – М МИФИ (ТУ). – 2005. – С. 163-171.

44. Макаров В. Ф. Основные методы исследования программных средств скрытого информационного воздействия // Безопасность информационных технологий. – 2009. – № 4. – С. 11-17.

45. Нестерук Г. Ф. О применении нейронечетких сетей в адаптивных системах информационной защиты // Нейроинформатика-2005: Материалы VII всероссийской научно- технической конференции. – М МИФИ (ТУ). – 2005. – С. 163-171.

46. Растригин J1.A. Адаптация сложных систем. Рига: Зинатне, 1981

47. Саати Т., Керне К. Аналитическое планирование. Организация систем. -М.: Радио и связь, 1991.

48. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем. М.: Финансы и статистика, 2003

49. Саати Т., Керне К. Аналитическое планирование. Организация систем. -М.: Радио и связь, 1991.

50. Создание систем поддержки принятия решений на основе хранилищ данных. – В.Львов.

51. Стефанцов А.Г. К вопросу о создании адаптивных информационных систем поддержки принятия управленческих решений // Труды ГНИИ МО РФ. Проблемы технического обеспечения и ремонта ВВТ, выпуск №86. -Б.М.: Б.И., 2003

52. Способы аналитической обработки данных для поддержки принятия решений. – Л. В. Щавелёв;

53. Туманов Ю. М.. Обнаружение вредоносных сценариев javascript на основе поведенческих сигнатур // Безопасность информационных технологий. – 2009. – № 4. – С. 63-65.

54. Туманов Ю. М.. Обнаружение вредоносных сценариев javascript на основе поведенческих сигнатур // Безопасность информационных технологий. – 2009. – № 4. – С. 63-65.

55. Коваленко М. Комп'ютерні віруси і захист інформації: Навчальний посібник. Наукова думка 1999 р. 268 ст.

56. Володимир Шаньгін Информационная безопасность и защита информации. – 2016.

ДОДАТОК А
ФРАГМЕНТ ЛІСТИНГУ ПРОГРАМИ

```
import java.util.*;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

class InterfaceFormatter extends Canvas {
  int   x, y;      // Location of cross hairs
  String event = "";   // Last key event type
  int   keyCode;    // Last keyCode pressed

  SampleCanvas() {
    w = getWidth();
    h = getHeight();
    font = Font.getFont(Font.FACE_SYSTEM,
     * Don't let the charts get too small
     */
    pad = 2;
    titleHeight = fh + pad * 2;
    eventHeight = fh * 3;

  protected void keyPressed(int key) {
    keyCode = key;
    event = "Pressed";
    handleActions(key);
    repaint();
  }

  protected void keyRepeated(int key) {
    event = "Released";
    repaint();
  }

  protected void pointerPressed(int x, int y) {
```

```
  this.x = x;
  repaint();
}
protected void pointerReleased(int x, int y) {
```

Продовження додатку А

```
 this.x = x;
  this.x = x;
    x += 1;
    break;
    case UP:
    y -= 1;
    break;
    case DOWN:
    y += 1;
    break;
  }
}
        */

  int yorig = barSize;
  if (bw < 2)
     bw = 2;
  g.setColor(255, 0, 0);
  g.fillRect(bw*1, yorig-h1, bw+1, h1);
  g.setColor(0, 255, 0);
  g.fillRect(bw*3, yorig-h2, bw+1, h2);
  g.setColor(0, 0, 255);
  g.fillRect(bw*5, yorig-h3, bw+1, h3);
  g.setColor(0);
  g.drawRect(bw*1, yorig-h1, bw, h1);
  g.drawRect(bw*3, yorig-h2, bw, h2);
  g.drawRect(bw*5, yorig-h3, bw, h3);

  g.translate(-(w + pad) / 2, pieSize + pad);

  g.setColor(128, 128, 128);
  int col1 = font.stringWidth("Action:");
  g.drawString("Key: ",   col1,   0,
```

```
            Graphics.TOP|Graphics.RIGHT);
    g.drawString(keyString(keyCode), col1, 0,
            Graphics.TOP|Graphics.LEFT);
    g.drawString("Action:",  col1,   fh,
            Graphics.TOP|Graphics.RIGHT);
    g.drawString(actionString(keyCode), col1, fh,
            Graphics.TOP|Graphics.LEFT);
    g.drawString("Event:",   col1,   fh*2,
```

Продовження додатку А

```
Graphics.TOP|Graphics.RIGHT);
    g.drawString(event,    col1,   fh*2,
            Graphics.TOP|Graphics.LEFT);
    int col2 = 80;
    g.drawString("x:",     col2,   0,
            Graphics.TOP|Graphics.RIGHT);
    g.drawString(Integer.toString(x), col2, 0,
            Graphics.TOP|Graphics.LEFT);
    g.drawString("y:",     col2,   fh,
            Graphics.TOP|Graphics.RIGHT);
    g.drawString(Integer.toString(y), col2, fh,
            Graphics.TOP|Graphics.LEFT);


    g.setColor(0, 0, 0);
    g.drawLine(x, y - 5, x, y + 5);
    g.drawLine(x - 5, y, x + 5, y);
   }

    if (keyCode == 0) {
      return "";
    }
   }

    if (keyCode == 0) {
      return "";
    }
        switch (action) {
        case FIRE:
```

```
            return "Fire";
          case LEFT:
            return "Left";
          case RIGHT:
            return "Right";
          case DOWN:
            return "Down";
          case UP:
            return "Up";
          case 0:
            return "";
          default:
```

Продовження додатку А

```
 return Integer.toString(action);
          }
  }
}
StockChecker checker = new StockChecker();
 TickerForm  form = new TickerForm();
 Alert    alert = new Alert( "Stock Alert!" );

 public StockWatcher() {
  display = Display.getDisplay( this );
  alert.setTimeout( Alert.FOREVER );
 }

  display.setCurrent( form );
  timer.schedule( checker, 0, 30000 );
 }

 protected void pauseApp() { }

 public void exit(){
  timer.cancel();
  destroyApp( true );
  notifyDestroyed();
 }
  public TickerForm(){
```

```
 super( "Stock Watch" );
 setTicker( ticker );
 addCommand( exitCommand );
 setCommandListener( this );
}

Random    generator = new Random();
int     sybsValue = 20000;
int     sunwValue = 30000;
int     ibmValue = 40000;
StringBuffer buf = new StringBuffer();

public void run(){
 String values = getStockValues();

 ticker.setString( values );
```

Продовження додатку А

```
if( sybsValue < 18000 || sybsValue > 22000 ||
      sunwValue < 28000 || sunwValue > 32000 ||
      ibmValue < 38000 || ibmValue > 42000 ){
  alert.setString( values );
 }

 if( !alert.isShown() ){
  display.setCurrent( alert, form );
 }
}

 sybsValue = randomStockValue( sybsValue );
 sunwValue = randomStockValue( sunwValue );
 ibmValue = randomStockValue( ibmValue );

 buf.setLength( 0 );
 appendValue( "SYBS", sybsValue );
 appendValue( "SUNW", sunwValue );
 appendValue( "IBM", ibmValue );
```

```
  return buf.toString();
 private int randomStockValue( int oldVal ){
  int incr1 = ( generator.nextInt() % 2 );
  int incr2 = ( generator.nextInt() % 16 );

  if( incr1 < 1 ){
   oldVal -= incr1 * 1000;
  } else {
   oldVal += ( incr1 - 2 ) * 1000;
  }

  if( incr2 < 8 ){
   oldVal -= incr2 * 250;
  } else {
   oldVal += incr2 * 250;
  }

  return oldVal;
 }
```

Продовження додатку А

```
private void appendValue( String stock, int val ){
  buf.append( stock );
  buf.append( ' ' );
  buf.append( Integer.toString( val / 1000 ) );
  buf.append( '.' );
  buf.append( Integer.toString( val % 1000 ) );
  buf.append( ' ' );
 }


 Display   display;
 Ticker    ticker = new Ticker( "" );
 Command   exitCommand = new Command(
                   "Exit", Command.EXIT, 1 );
 Timer     timer = new Timer();
public StockWatcher() {
  display = Display.getDisplay( this );
```

```
 alert.setTimeout( Alert.FOREVER );
 }
protected void startApp() {
 display.setCurrent( form );
 timer.schedule( checker, 0, 30000 );
 }

 protected void pauseApp() { }

 public void exit(){
 timer.cancel();
 destroyApp( true );
 notifyDestroyed();
 }

 public TickerForm(){
  super( "Stock Watch" );
  setTicker( ticker );
  addCommand( exitCommand );
  setCommandListener( this );
 }
  exit();
 }
}
```

Продовження додатку А

```
if( sybsValue < 18000 || sybsValue > 22000 ||
       sunwValue < 28000 || sunwValue > 32000 ||
       ibmValue < 38000 || ibmValue > 42000 ){
   alert.setString( values );
  }

  if( !alert.isShown() ){
   display.setCurrent( alert, form );
  }
Ticker    ticker = new Ticker( "" );
protected void destroyApp( boolean unconditional ) { }
```

```
 protected void startApp() {
  display.setCurrent( form );
  timer.schedule( checker, 0, 30000 );
 }

 protected void pauseApp() { }

 public void exit(){
  timer.cancel();
  destroyApp( true );
  notifyDestroyed();
 }
  public TickerForm(){
   super( "Stock Watch" );
   setTicker( ticker );
   addCommand( exitCommand );
   setCommandListener( this );
   exit();
  }
 }

lass StockChecker extends TimerTask {
  Random    generator = new Random();
  int     sybsValue = 20000;
  int     sunwValue = 30000;
  int     ibmValue = 40000;
  StringBuffer buf = new StringBuffer();
```

Продовження додатку А

```
public void run(){
   String values = getStockValues();

   ticker.setString( values );

   if( sybsValue < 18000 || sybsValue > 22000 ||
       sunwValue < 28000 || sunwValue > 32000 ||
       ibmValue < 38000 || ibmValue > 42000 ){
    alert.setString( values );
```

```
   }

   if( !alert.isShown() ){
    display.setCurrent( alert, form );
   }
  }

  public void prepare() {
    input = new TextBox("Enter some text: ", "", 5, TextField.ANY);
    input.addCommand(backCommand);
    input.setCommandListener(this);
    input.setString("");
    display.setCurrent(input);
  }

   prepare();
   currentMenu = "item1";
  }

  /**
   * Test item2.
   */
  public void testItem2() {
    prepare();
    currentMenu = "item2";
Test item3.
   */
  public void testItem3() {
    prepare();
    currentMenu = "item3";
  }
```

Продовження додатку А

```
  /**
   * Test item4.
   */
  public void testItem4() {
    prepare();
```

```
   currentMenu = "item4";
  }
public class StockWatcher extends MIDlet {

 Display   display;
 Ticker    ticker = new Ticker( "" );
 Command   exitCommand = new Command(
                   "Exit", Command.EXIT, 1 );
 Timer    timer = new Timer();
 StockChecker checker = new StockChecker();
 TickerForm  form = new TickerForm();
 Alert    alert = new Alert( "Stock Alert!" );

 public StockWatcher() {
  display = Display.getDisplay( this );
  alert.setTimeout( Alert.FOREVER );
 }
 protected void destroyApp( boolean unconditional ) { }

 protected void startApp() {
  display.setCurrent( form );
  timer.schedule( checker, 0, 30000 );
 }
 protected void pauseApp() { }

 public void exit(){
  timer.cancel();
  destroyApp( true );
  notifyDestroyed();
 }
  public TickerForm(){
   super( "Stock Watch" );
   setTicker( ticker );
  addCommand( exitCommand );
   setCommandListener( this );
  }
```

Продовження додатку А

```
 }
```

```
}
 class StockChecker extends TimerTask {
Random    generator = new Random();
int     sybsValue = 20000;
int     sunwValue = 30000;
int     ibmValue = 40000;
StringBuffer buf = new StringBuffer();

public void run(){
 String values = getStockValues();

 ticker.setString( values );

 if( sybsValue < 18000 || sybsValue > 22000 ||
     sunwValue < 28000 || sunwValue > 32000 ||
     ibmValue < 38000 || ibmValue > 42000 ){
  alert.setString( values );
 }
 if( !alert.isShown() ){
  display.setCurrent( alert, form );
 }
}
*/
 String label = c.getLabel();
   destroyApp(true);
 } else if (label.equals("Back")) {
   if(currentMenu.equals("item1") || currentMenu.equals("item2") ||
     currentMenu.equals("item3") || currentMenu.equals("item4")) {
    // go back to menu
    mainMenu();
   }
 } else {
  switch(down.getSelectedIndex()) {

case 0: testItem1();break;
   }

 }
```

```
    }
}
```

Додаток Б

Магістерська атестаційна робота

На тему:

Модуль захисту приватної комп'ютерної системи від шкідливого програмного забезпечення

Виконав: Лук’яненко Т.П.

Керівник: Давиденко А.М.

Кафедра безпеки інформаційних технологій

Актуальність

На сьогоднішній день велика кількість компаній має безліч пристрої, які використовуються співробітниками в особистих цілях, що часто призводить до витоку конфіденційних даних підприємства. Тому інформаційні ресурси бізнес-організацій є найбільш привабливими для кіберзлочинців і потребують захисту в першу чергу. Однією з найважливіших задач інформаційної безпеки є боротьба зі шкідливим програмним забезпеченням (ШПЗ) і зокрема його виявлення. Ефективність управління в складних галузевих системах значною мірою визначається ефективністю реалізації процедур аналізу, обробки інформації та прийняття рішень. Тому, **актуальною** задачею є підвищення ефективності виявлення шкідливого програмного забезпечення.

Продовження додатку Б

Мета і задачі

Метою роботи є розробка модуля захисту приватної комп'ютерної системи від шкідливого програмного забезпечення

Для реалізації зазначеної мети необхідно вирішити наступні **задачі**:

- Проаналізувати методи і засоби виявлення шкідливого програмного забезпечення.
- Розробити модуль захисту та удосконалену комп'ютерну інформаційну систему.
- Провести експериментальне дослідження.

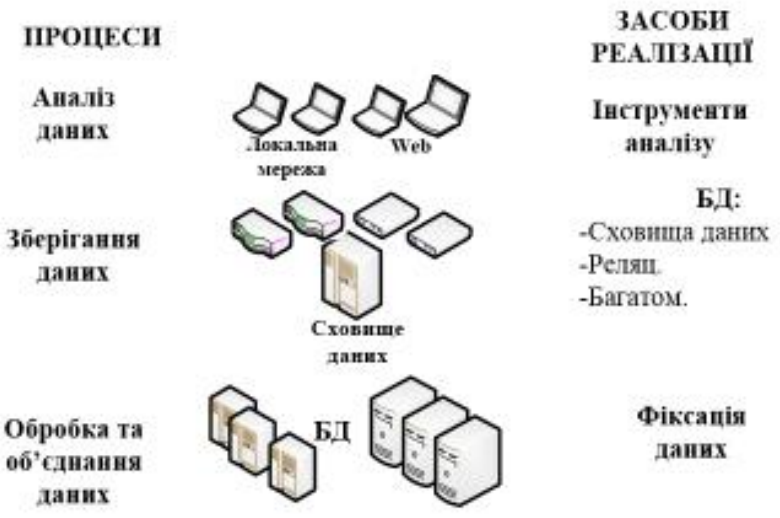
Новизна

Наукова новизна - удосконалено інформаційну систему за рахунок розробки модуля, що дозволить підвищити рівень захисту користувачів від шахрайських дій зловмисників

Аналіз

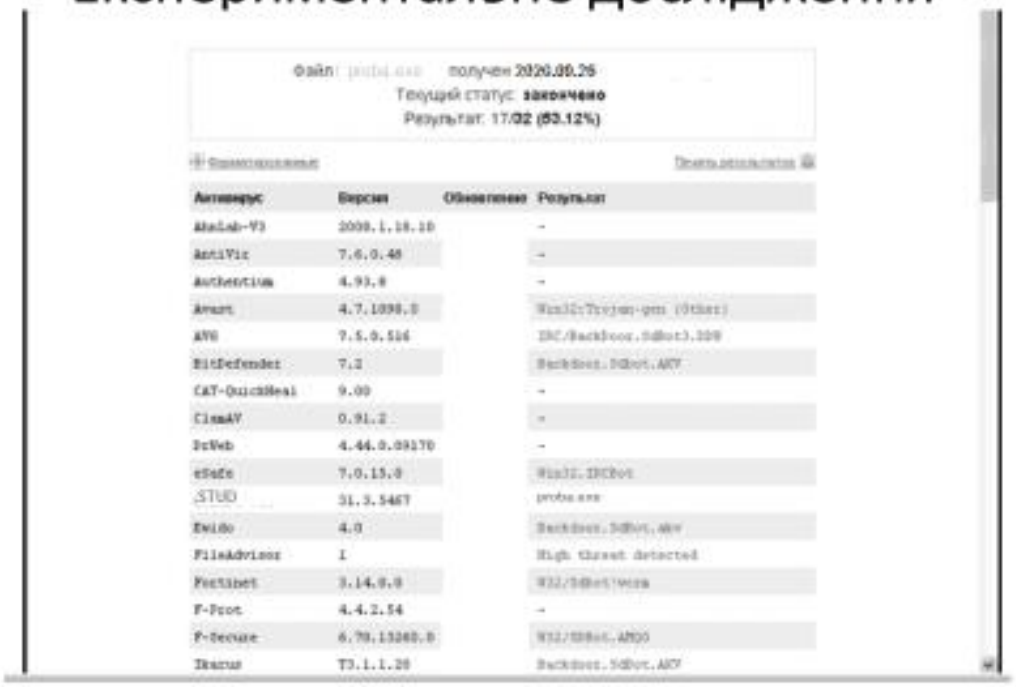
Програмний засіб	Використання ресурсів	Запобігання несанкціон. підключення	Швидкість виявлення ШПЗ	Контроль вхідних SMS	Виявлення
MobiShield	+	+	-	-	-
F-Secure Security	-	-	-	-	-
Dr.Web for Symbian OS	+	-	-	+	-
NetQin Anti-Virus	+	-	+	-	-
NetQin Anti-Virus Pro	+	-	-	-	-
ESET Security Beta	+	+	-	+	-
Panda Security	-	+	-	-	-
BitDefender Security	+	-	+	-	-
AntiVir Mobile	+	-	+	-	-
Kaspersky Anti-Virus	-	-	+	-	-

Структурна схема удосконаленої системи



Продовження додатку Б

Експериментальне дослідження



Переваги вдосконаленої розробленим додатком системи

Провівши дане експериментальне дослідження можна виявити переваги вдосконаленої системи для виявлення шкідливого програмного забезпечення порівняно з існуючими засобами:

- зручний, простий інтерфейс;
- низьке споживання ресурсів;
- запобігання несанкціонованого підключення;
- висока швидкість сканування;
- контроль вхідних SMS/MMS;
- виявлення STUD.

Продовження додатку Б

Експериментальне дослідження

Програмний засіб	Використання ресурсів	Запобігання несанкціон. підключення	Швидкість виявлення ШПЗ	Контроль вхідних SMS	Виявлення
MobiShield	+	+	-	-	-
F-Secure Security	-	-	-	-	-
Dr.Web for Symbian OS	+	-	-	+	-
NetQin Anti-Virus	+	-	+	-	-
NetQin Anti-Virus Pro	+	-	-	-	-
ESET Security Beta	+	+	-	+	-
Panda Security	-	+	-	-	-
BitDefender Security	+	-	+	-	-
AntiVir Mobile	+	-	+	-	-
Kaspersky Anti-Virus	-	-	+	-	-
Вдосконалена система STUD	+	+	+	+	+

Переваги вдосконаленої розробленим додатком системи

Провівши дане експериментальне дослідження можна виявити переваги вдосконаленої системи для виявлення шкідливого програмного забезпечення порівняно з існуючими засобами:

- зручний, простий інтерфейс;
- низьке споживання ресурсів;
- запобігання несанкціонованого підключення;
- висока швидкість сканування;
- контроль вхідних SMS/MMS;
- 100 % виявлення ШПЗ.

Продовження додатку Б

Висновки

1. Проаналізовано найпоширеніше шкідливе програмне забезпечення та сучасні методи захисту від нього, що дало можливість формалізувати вимоги до удосконалення системи захисту.

2. Розроблено захисний модуль, який відповідає захист від дій шкідливих кодів у телекомунікаційних системах.

3. Розроблено програмне забезпечення, що дало можливість провести експериментальне дослідження та визначити основні переваги відносно існуючих засобів.