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 *Бєлоха Германа Костянтиновича*

академічної групи *125м-19-1*

спеціальності *125 Кібербезпека*

спеціалізації[1]

за освітньо-професійною програмою *Кібербезпека*

на тему *Методи захисту web-додатків від інформаційних атак на основі java-аплетів*

Керівники	Прізвище, ініціали	Оцінка за шкалою		Підпис
		рейтинговою	інституційною	
кваліфікаційної роботи	д.т.н., проф. Корнієнко В.І.			
розділів:				
спеціальний	ас. Ковальова Ю.В.			
економічний	к.е.н., доц. Пілова Д.П.			

Рецензент				

Нормоконтролер	ст. викл. Мєшков В.І.			

Дніпро
2020

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

«_____»_____20___року

ЗАВДАННЯ
на кваліфікаційну роботу
ступеня магістр

студенту *Бєлоху Герману Костянтиновичу* **академічної групи** *125м-19-1*
(прізвище ім'я по-батькові) (шифр)

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

на тему ***Методи захисту web-додатків від інформаційних атак на основі java-аплетів***

затверджену наказом ректора НТУ «Дніпровська політехніка» від _____ № ____

Розділ	Зміст	Термін виконання
Розділ 1	Привести класифікацію атак на WEB-сервера, проаналізувати принцип роботи java-аплетів та існуючи технології безпеки в Java. Проведести поглиблений аналіз існуючих засобів безпеки Інтернет-браузерів.	10.10.2020
Розділ 2	Виконати аналіз загроз WEB-додатків та розробити модель загроз, обґрунтувати вибір профілю захищеності. Узагальнити переваги та недоліки кожного з видів Інтернет-браузерів. Запропонувати засоби захисту WEB-додатків від атак, які реалізуються за допомогою java-аплетів.	20.11.2020
Розділ 3	Виконати розрахунок витрат на розробку засобів захисту WEB-додатків. Розрахувати можливі збитки від реалізації атак на WEB-додатки.	05.12.2020

Завдання видано _____ _____
(підпис керівника) (прізвище, ініціали)

Дата видачі: 01.09.2020р.

Дата подання до екзаменаційної комісії: 11.12.2020р.

Прийнято до виконання _____ Бєлох Г.К.
(підпис студента) (прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: ___ с., __ рис., __ табл., __ додатки, __ джерел.

Об’єкт досліджень – процес захисту WEB-додатків від атак реалізованих за допомогою java-аплетів.

Предмет досліджень – методи захисту WEB-додатків від атак реалізованих за допомогою java-аплетів.

Мета магістерської роботи: підвищення захисту WEB-додатків від атак реалізованих за допомогою java-аплетів, що загрожують конфіденційності, доступності та цілісності інформації.

У розділі «Стан питання. Постановка задачі» були приведена класифікація атак на WEB-сервера, проаналізований принцип роботи java-аплетів та існуючи технології безпеки в Java. Проведений поглиблений аналіз існуючих засобів безпеки Інтернет-браузерів.

У спеціальній частині був проведений аналіз загроз WEB-додатків та розроблена модель загроз, обґрунтовано вибір профілю захищеності. Узагальнено переваги та недоліки кожного з видів Інтернет-браузерів. Запропоновано засоби захисту WEB-додатків від атак, які реалізуються за допомогою java-аплетів.

В економічній частині виконано розрахунок витрат на розробку засобів захисту WEB-додатків. Розраховані можливі збитки від реалізації атак на WEB-додатки.

JAVA-АПЛЕТ, БРАУЗЕР, WEB-ДОДАТОК, ЗАГРОЗА, АТАКА, JAVA.

РЕФЕРАТ

Пояснительная записка: __ с., __ рис., __ табл., __ приложений, __ источников.

Объект исследований – процесс защиты WEB-приложений от атак реализованных с помощью java-апплетов.

Предмет исследований – методы защиты WEB-приложений от атак реализованных с помощью java-апплетов.

Цель магистерской работы: повышение защиты WEB-приложений от атак реализованных с помощью java-апплетов, которые угрожают конфиденциальности, доступности и целостности информации.

В разделе «Состояние вопроса. Постановка задачи» были приведена классификация атак на WEB-сервера, проанализирован принцип работы java-апплетов и существующие технологии безопасности в Java. Проведенный углубленный анализ существующих средств безопасности Интернет-браузеров.

В специальной части был проведен анализ угроз WEB-приложений и разработана модель угроз, обоснован выбор профиля защищенности. Обзор преимущества и недостатки каждого из видов Интернет-браузеров. Предложены средства защиты WEB-приложений от атак, которые реализуются с помощью java-апплетов.

В экономической части произведен расчет затрат на разработку средств защиты WEB-приложений. Рассчитаны возможные убытки от реализации атак на WEB-приложения.

JAVA-апплет, браузер WEB-ПРИЛОЖЕНИЕ, УГРОЗА, АТАКА, JAVA.

ABSTRACT

Explanatory note: ___ p., __ fig., __ tab., __ application, __ sources.

The object of research is the process of protecting WEB-applications from attacks implemented using java-applets.

The subject of research - methods of protecting WEB-applications from attacks implemented using java-applets.

The purpose of the master's thesis: to increase the protection of WEB-applications from attacks implemented using java-applets that threaten the confidentiality, accessibility and integrity of information.

In the section "Status of the issue. Problem statement "the classification of attacks on the WEB-server, the principle of operation of java-applets and existing security technologies in Java were analyzed. An in-depth analysis of existing security features of Internet browsers.

In the special part the analysis of threats of WEB-applications was carried out and the model of threats was developed, the choice of a security profile was substantiated. The advantages and disadvantages of each type of Internet browser are summarized. Means of protection of WEB-applications from attacks which are realized by means of java-applets are offered.

In the economic part, the calculation of costs for the development of means of protection of WEB-applications is performed. Possible losses from the implementation of attacks on WEB-applications are calculated.

JAVA-APLET, BROWSER, WEB-APPLICATION, THREAT, ATTACK, JAVA.

СПИСОК УМОВНИХ СКОРОЧЕНЬ

ІБ	–	інформаційна безпека;
ІС	–	інформаційна система;
КЗЗ	–	комплекс засобів захисту;
КС	–	комп’ютерна система;
КСІБ	–	комп'ютерна система інформаційної безпеки;
НСД	–	несанкціонований доступ;
ПЗ	–	програмне забезпечення;
ПК	–	персональний комп’ютер;
СІБ	–	система інформаційної безпеки;
ТЗ	–	технічне завдання.

ЗМІСТ

с.

ВСТУП

Технологія Java міцно завойовує сучасний комп'ютерний світ. Її широкі можливості створення розподілених обчислень не можуть залишити байдужими жодного розробника програмного забезпечення. Однак часто буває так, що серйозні можливості породжують не менш серйозні проблеми, пов'язані із забезпеченням безпеки.

Програмне середовище в результаті появи Інтернету перетворилася в середу інтерактивних програм, прикладом якої є Java. У цьому середовищі користувач взаємодіє з сервером з допомогою мережі. Сервер завантажує додаток (аплет) на власний комп'ютер, який потім його виконує. При використанні такої стратегії має місце безліч ризиків.

Раніше стверджувалося, що при використанні таких мов як Java не вдається занести вірус з-за обмежень, вбудованих в мова для управління доступом до файлової системи і для управління виконанням програми. Зараз ці заяви до-сить ефективно спростовані.

Мова Java дає програмістам можливість не просто розробляти нові програми, але і використовувати елементи вже написаних і перевірених програм. Такий модульний принцип дозволяє швидко писати нові програмні продукти та ефективно модернізувати старі. Крім того, у стандарт мови входить безліч корисних бібліотек, на основі яких можна будувати обчислювальні системи будь-якої складності.

Ще однією особливістю Java є аплети. Аплет – це невелика програма, в якій повинно бути визначено декілька обов'язкових функцій. Аплет завантажується по мережі і може виконуватися на Web-браузері, який підтримує Java. Саме ця частина Java-технології призначена для використання у всесвітній мережі Internet, і тому захист повинна поширюватися як на сам аплет, так і на клієнта мережі, який використовує цей аплет.

РОЗДІЛ 1. СТАН ПИТАННЯ. ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність обраної теми

Актуальністю обраної теми є те, що більшість атак через Інтернет здійснюється за допомогою експлойтів, які використовують помилки в ПЗ, щоб пробити захист комп'ютера і отримати можливість виконання шкідливого коду без відома користувача.

Ця проблема є суттєвою, оскільки проаналізувавши динаміку розвитку вірусів можна зробити висновок, що для розповсюдження шкідливих програм кіберзлочинці все частіше роблять ставку на незакриті вразливості в платформі Java, що є самим слабким елементом захисту операційних систем, на яких вона встановлена[1].

Протягом останнього року антивірусні технології Microsoft виявили або заблокували в середньому 6,9 мільйона спроб застосування аплетів до компонентів Java, або в цілому, 27,5 мільйона спроб протягом року[2].

За даними глобальної системи моніторингу загроз Kaspersky Security Network, кількість Java-аплетів склала близько 14% від загального числа виявлених загроз[3].

Діаграма розвитку java-аплетів за останній рік наведена на рисунку 1.1.

Рисунок 1.1 – Діаграма розвитку java-аплетів

1.2 Мета дослідження

Підвищення захисту Web-додатків від атак реалізованих за допомогою java-аплетів, що загрожують конфіденційності, доступності та цілісності інформації.

1.3 Об’єкт дослідження

Об’єктом дослідження є процес захисту Web-додатків від атак в мережі Інтернет.

1.4 Предмет дослідження

Предметом дослідження є організаційні засоби захисту Web-додатків від атак реалізованих за допомогою java-аплетів.

1.5 Задачі дослідження

1 Аналіз загроз Web-серверів;

2 Аналіз Web-браузерів;

3 Аналіз засобів захисту від java-аплетів;

4 Вибір профілю захищеності;

5 Розробка організаційних заходів захисту Web-додатків в мережі Інтернет.

1.6 Класифікація атак на Web-сервера

Класифікація атак на Web-сервера має ієрархічну структуру та розділяється на шість основних класів.

1 Атаки на засоби аутентифікації;

2 Атаки на засоби авторизації;

3 Логічні атаки;

4 Атаки направлені на розголошення інформації;

5 Атаки на клієнтів;

6 Атаки направлені на виконання коду.

1.6.1 Атаки на засоби аутентифікації

Атаки цього класу спрямовані на використовувані Web-додатком методи перевірки ідентифікатора користувача, служби або програми. Вони направлені на обхід чи експлуатацію вразливостей в механізмах реалізації аутентифікації Web-серверів.

а) Підбір (Brute Force)

Підбір – автоматизований процес проб і помилок, що використовується для того, щоб вгадати ім'я користувача, пароль, номер кредитної картки, ключі.

Багато систем дозволяють використовувати слабкі паролі або ключі шифрування, і користувачі часто вибирають легко вгадуванні парольні фрази або такі, які містяться у словниках.

Використовуючи цю ситуацію, зловмисник може скористатися словником і спробувати використати тисячі або навіть мільйони комбінацій символів в якості пароля.

Якщо випробуваний пароль дозволяє отримати доступ до системи, атака вважається успішною і атакуючий може використовувати обліковий запис.

Подібна техніка проб і помилок може бути використана для підбору ключів шифрування. У разі використання ключів недостатньою довжини, зловмисник може отримати необхідний ключ, перебравши всі можливі комбінації.

Існує два види підбору: прямий і зворотній. При прямому підборі використовуються різні варіанти пароля для одного імені користувача. При зворотньому - перебираються різні імена користувачів, а пароль залишається незмінним. У системах з мільйонами облікових записів ймовірність використання різними користувачами одного пароля досить висока. Не дивлячись на популярність і високу ефективність, підбір може займати кілька годин, днів або років.

Цей вид атак широко використовується переважно там, де відсутнє блокування у разі невірного поєднання[4].

Приклад:

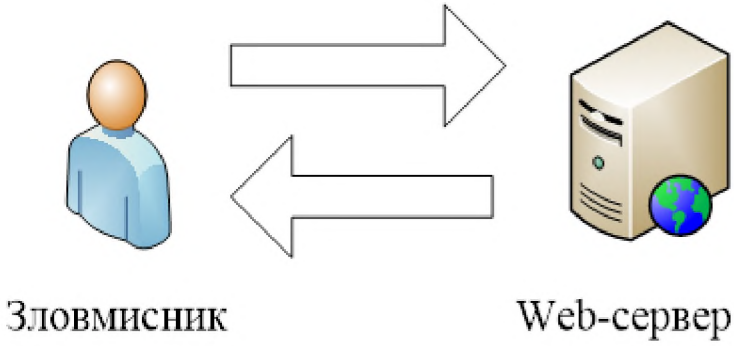
Им’я користувача = Jon

Паролі = smith, michael-jordan, [pet names], [birthdays], [car names], ...

Имена користувачів = Jon, Dan, Ed, Sara, Barbara, ...

Пароль = 12345678

На рисунку 1.2 зображений механізм підбору пароля.



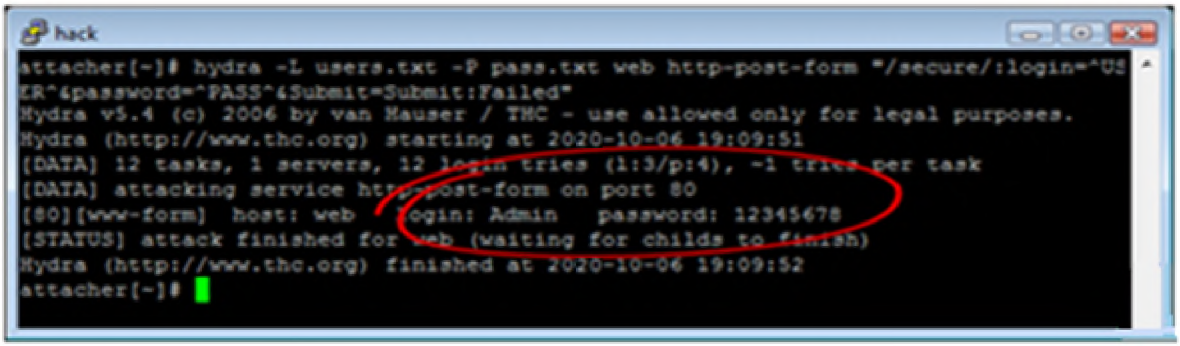


Рисунок 1.2 – Атака типу «підбір»

б) Недостатня аутентифікація (Insufficient Authentication)

Ця уразливість виникає тоді, коли Web-сервер дозволяє зловмиснику діставати доступ до важливої інформації або функцій сервера без належної аутентифікації. Атаки подібного роду дуже часто реалізуються за допомогою інтерфейсу адміністрування через Web. Щоб не використовувати аутентифікацію, деякі ресурси по дефолту використовують певну адресу, яка не вказана на основних сторінках сервера або інших загальнодоступних ресурсах.

Необхідний URL може бути знайдений шляхом перебору типових файлів і директорій (таких, як /admin/) з використанням повідомлень про помилки журналів перехресних посилань або шляхом простого читання документації.

Подібні ресурси мають бути захищені адекватно важливості їх вмісту і функціональних можливостей.

Приклад:

Багато Web-додатки стандартно використовують для адміністративного доступу посилання в кореневій директорії сервера (/ admin /). Зазвичай посилання на цю сторінку не фігурує у вмісті сервера, проте сторінка доступна за допомогою стандартного браузера. Оскільки користувач або розробник припускає, що ніхто не скористається цією сторінкою, так як посилання на неї відсутнє, найчастіше реалізацією аутентифікації нехтують. І для отримання контролю над сервером зловмисникові достатньо зайти на цю сторінку.

в) Небезпечне відновлення паролів(Weak Password Recovery Validation)

Ця вразливість реалізується завдяки тому, що Web-сервер дозволяє зловмиснику несанкціоновано отримувати, модифікувати або відновлювати паролі інших користувачів.

Часто аутентифікація на Web-сервері вимагає від користувача запам'ятовування пароля або парольної фрази. Строга політика безпеки передбачає що тільки користувач повинен знати пароль, причому пам'ятати його чітко.

Прикладом реалізації функції відновлення паролю є використання «секретного питання», відповідь на який вказується в процесі реєстрації. Питання або вибирається із списку, або вводиться самим користувачем. Ще один механізм дозволяє користувачеві вказати «підказку», яка допоможе йому згадати пароль. Інші способи вимагають від користувача вказати частину персональних даних - таких, як номер паспорта, домашня адреса, поштовий індекс і так далі, - які потім використовуватимуться для встановлення особи. Після того як користувач доведе свою ідентичність, система відобразить новий пароль або перешле його поштою.

Уразливості, пов'язані з недостатньою перевіркою при відновленні пароля, виникають, коли зловмисник отримує дані, які використовуються механізмом відновлення. Це трапляється, коли інформацію для перевірки користувача легко

вгадати або сам процес підтвердження можна обійти. Система відновлення пароля може бути скомпрометована шляхом використання підбору вразливостей системи або із-за легковгадуваної відповіді на секретне питання.

Приклад:

Слабкі методи відновлення паролів

Перевірка інформації

Багато серверів вимагають від користувача вказати його email в комбінації з домашньою адресою і номером телефону. Ця інформація може бути легко отримана з мережевих довідників. В результаті, дані, які використовуються для перевірки, не є великим секретом. Крім того, ця інформація може бути отримана зловмисником з використанням інших методів, таких як міжсайтового виконання сценаріїв або phishing.

Прольні підказки

Сервер, що використовує підказки для полегшення запам'ятовування паролів, може бути атакований, оскільки підказки допомагають у реалізації підбору паролів. Користувач може використовувати стійкий пароль, наприклад, "221277King" з відповідною підказкою: "д-р + улюблений письменник". Атакуючий може припустити, що для користувача пароль складається з дати народження та імені улюбленого автора користувача. Це допомагає сформувати відносно короткий словник для атаки шляхом перебору.

Таємне питання та відповідь

Припустимо, відповідь користувача "Будапешт", а секретне питання "Місце народження". Зловмисник може обмежити словник для підбору секретної відповіді назвами міст. Більш того, якщо атакуючий має в своєму розпорядженні деякою інформацією про користувача, дізнатися його місце народження не складно[5].

1.6.2 Атаки на засоби авторизації (Authorization)

Атаки цього класу спрямовані на методи, які використовуються Web-сервером для визначення того, чи має користувач, служба або програма необхідний дозвіл для вчинення дії.

Багато Web-ресурсів дозволяють доступ до деякого вмісту тільки певним користувачам. Доступ для інших має бути обмежений. Використовуючи різну техніку, зловмисник може підвищити свої привілеї і дістати доступ до захищених ресурсів.

а) Передбачуване значення ідентифікатора сесії (Credential/Session Prediction)

Передбачуване значення ідентифікатора сесії дозволяє перехоплювати сесії інших користувачів. Подібні атаки виконуються шляхом передбачення або вгадування унікального ідентифікатора сесії користувача. Ця атака також як і перехоплення сесії (Session Hijacking) у разі успіху дозволяє зловмисникові послати запит Web-сервера з правами скомпрометованого користувача. Дизайн багатьох серверів припускає аутентифікацію користувача при першому зверненні та подальше відстеження його сесії.

Для цього користувач вказує комбінацію імені та пароля. Замість повторної передачі ім'я користувача та пароль при кожній транзакції, Web-сервер генерує унікальний ідентифікатор, який присвоюється сесії користувача. Наступні запити користувача до сервера містять ідентифікатор сесії як доказ того, що аутентифікація була успішно пройдено. Якщо атакуючий може передбачити або вгадати значення ідентифікатора іншого користувача, це може бути використано для проведення атаки.

Приклад:

Багато серверів генерують ідентифікатори сесії, використовуючи алгоритми власної розробки. Подібні алгоритми можуть просто збільшувати значення ідентифікатора для кожного запиту користувача. Інший поширений варіант - використання функції від поточного часу або інших специфічних для комп'ютера даних.

Ідентифікатор сесії зберігається в cookie, прихованих полях форм або URL. Якщо атакуючий має можливість визначити алгоритм, використовуваний для генерації сесії, він може виконати наступні дії:

1) підключитися до сервера, використовуючи поточний ідентифікатор сесії;

2) обчислити або підібрати наступний ідентифікатор сесії;

3) присвоїти отримане значення ідентифікатора cookie / прихованого полю форми /URL[6,7,8].

б) Недостатня авторизація (Insufficient Authorization)

Недостатня авторизація виникає, коли Web-сервер дозволяє атакуючому отримувати доступ до важливої інформації або функцій, доступ до яких повинен бути обмежений. Те, що користувач пройшов аутентифікацію не означає, що він повинен отримати доступ до всіх функцій і вмісту сервера. Крім аутентифікації повинно бути реалізовано розмежування доступу.

Процедура авторизації визначає, які дії може здійснювати користувач, служба або додаток. Правильно побудовані правила доступу повинні обмежувати дії користувача відповідно до політики безпеки. Доступ до важливих ресурсів сайту повинен бути дозволений тільки адміністраторам.

Приклад:

У минулому багато Web-серверів зберігали важливі ресурси в "прихованих" директоріях, таких як "/ admin" або "/ log". Якщо атакуючий запитував ці ресурси напряму, він отримував до них доступ і міг переналаштувати сервер, отримати доступ до важливої інформації або повністю скомпрометувати систему.

Деякі сервери, після аутентифікації, зберігають у cookie або прихованих полях ідентифікатор "ролі" користувача в рамках Web-додатків. Якщо розмежування доступу грунтується на перевірці даного параметра без верифікації приналежності до ролі при кожному запиті, зловмисник може підвищити свої привілеї, просто модифікувавши значення cookie.

Наприклад, значення cookie

SessionId=12345678;Role=User

Замінюється на

SessionId=12345678;Role=Admin[4].

в) Відсутність тайм-ауту сесії (Insufficient Session Expiration)

Атаки цього класу виникають у випадку якщо для ідентифікатора сесії або облікових даних не передбачений таймаут або має значення дуже велике, зловмисник може скористатися старими даними для авторизації. Це підвищує уразливість сервера для атак, пов'язаних з крадіжкою ідентифікаційних даних. Оскільки протокол HTTP не передбачає контроль сесії, Web-сервери зазвичай використовують ідентифікатори сесії для визначення запитів користувача. Таким чином, конфіденційність кожного ідентифікатора повинна бути забезпечена, щоб запобігти багаторазовий доступ користувачів з одним профілем.

Викрадений ідентифікатор може використовуватися для доступу до даних користувача або здійснення шахрайських транзакцій. Відсутність таймауту сесії збільшує ймовірність успіху різних атак. Приміром, зловмисник може отримати ідентифікатор сесії, використовуючи мережевий аналізатор або вразливість типу міжсайтового виконання сценаріїв. Хоча таймаут не допоможе у випадку, якщо ідентифікатор буде використаний негайно, обмеження часу допоможе у випадку більш пізніх спроб використання ідентифікатора.

В іншій ситуації, якщо користувач отримує доступ до сервера з публічного комп'ютера (бібліотека, Internet-кафе і т.д.) відсутність таймауту сесії може дозволити зловмисникові скористатися історією браузера для перегляду сторінок користувача.

Велике значення таймауту збільшує шанси підбору чинного ідентифікатора. Крім того, збільшення цього параметра веде до збільшення одночасно відкритих сесій, що ще більше підвищує ймовірність успішного підбору.

Приклад:

При використанні публічного комп'ютера, коли кілька користувачів мають необмежений фізичний доступ до машини, відсутність таймауту сесії дозволяє зловмисникові переглядати сторінки, відвідані іншим користувачем. Якщо

функція виходу з системи просто перенаправляє на основну сторінку Web-сервера, а не завершує сесію, сторінки, відвідані користувачем, можуть бути переглянуті зловмисником.

Оскільки ідентифікатор сесії не був відзначений як недійсний, атакуючий отримає доступ до сторінок сервера без повторної аутентифікації[10].

г) Фіксація сесії (Session Fixation)

Використовуючи даний клас атак, зловмисник присвоює ідентифікатору сесії користувача задане значення. Залежно від функціональних можливостей сервера, існує декілька способів зафіксувати значення ідентифікатора сесії. Для цього можуть використовуватися атаки типу міжсайтового виконання сценаріїв або підготовка сайту з допомогою попереднього HTTP запиту. Після фіксації значення ідентифікатора сесії атакуючий очікує моменту, коли користувач увійде в систему. Після входу користувача, зловмисник використовує ідентифікатор сесії для отримання доступу до системи від імені користувача.

Можна виділити два типи систем управління сесіями на основі ідентифікаторів.

Перший з них дає змогу браузеру вказувати будь-який ідентифікатор.

Системи другого типу обробляють тільки ідентифікатори, згенеровані сервером. Якщо використовуються системи першого типу, зловмисник може вибрати будь-який ідентифікатор сесії. У другому випадку зловмисникові доводиться підтримувати встановлену сесію і періодично з'єднуватися з сервером для уникнення закриття сесії за таймаут.

Без наявності активного захисту від фіксації сесії, ця атака може бути використана проти будь-якого сервера, аутентифікує користувачів за допомогою ідентифікатора сесії.

Більшість Web-серверів зберігає ID в cookie, але це значення так само може бути присутнім в URL або прихованому полі форми.

Системи, що використовують cookie, є найбільш уразливими. Більшість відомих на даний момент варіантів фіксації сесії спрямовані саме на значення cookie.

На відміну від крадіжки ідентифікатора, фіксація сесії надає зловмисникові набагато більший можливостей. Це пов'язано з тим, що активна фаза атаки відбувається до входу користувача в систему.

Приклад:

Атаки, спрямовані на фіксацію сесії зазвичай проходять у три етапи.

1) Встановлення сесії

Зловмисник встановлює нелегальну-сесію на атакуючому сервер і отримує від сервера ідентифікатор або вибирає довільний ідентифікатор. У деяких випадках нелегальна-сесія повинна підтримуватися в активному стані шляхом періодичних звернень до сервера.

2) Фіксація сесії

Зловмисник передає значення ідентифікатора нелегальної-сесії браузеру користувача та фіксує його ідентифікатор сесії. Це можна зробити, наприклад, встановивши значення cookie в браузері за допомогою XSS.

3) Підключення до сесії

Атакуючий очікує аутентифікації користувача на сервері. Після того, як користувач зайшов на сайт, зловмисник підключається до сервера, використовуючи зафіксований ідентифікатор, і отримує доступ до сесії користувача[11,12].

1.6.3 Логічні атаки

Атаки цього класу спрямовані на експлуатацію функцій ПЗ або логіки його функціонування. Логіка ПЗ є очікуваним процесом функціонування програми при виконанні певних дій. Як приклади можна привести відновлення паролів, реєстрацію облікових записів, аукціонні торги, транзакції в системах електронної комерції. ПЗ може вимагати від користувача коректного виконання декількох

послідовних дій для отримання певного результату. Зловмисник може обійти ці механізми або використовувати їх у своїх цілях.

а) Зловживання функціональними можливостями (Abuse of Functionality)

Ця атака спрямована на використання функцій програмного Web-забезпечення з метою обходу механізмів розмежування доступу. Деякі механізми програмного Web-забезпечення включаючи функції забезпечення безпеки можуть бути використані для цих цілей. Наявність вразливості в одному з другорядних компонентів ПЗ може привести до компрометації усього ПЗ. Рівень ризику і потенційні можливості зловмисника у разі проведення атаки дуже сильно залежать від конкретного ПЗ. Зловживання функціональними можливостями дуже часто використовується спільно з іншими атаками - такими, як зворотний шлях в директоріях і так далі.

Приклади зловживання функціональними можливостями включають:

– використання функцій пошуку для отримання доступу до файлів за межами кореневої директорії WEB-сервера;

– використання функції завантаження файлів на сервер для перезапису файлів конфігурації або впровадження серверних сценаріїв;

– реалізацію відмови в обслуговуванні шляхом використання функції блокування облікового запису при багаторазовому введенні неправильного пароля[13].

б) Відмова в обслуговуванні (Denial of Service)

Цей клас атак спрямований на порушення доступності WEB-сервера. Атаки, спрямовані на відмову в обслуговуванні, реалізуються на мережевому рівні, проте вони можуть бути спрямовані і на прикладний рівень. Використовуючи функції програмного WEB забезпечення зловмисник може вичерпати критичні ресурси системи або скористатися вразливістю, що призводить до припинення функціонування системи.

Зазвичай DoS-атаки спрямовані на вичерпання критичних системних ресурсів - таких, як обчислювальні потужності, оперативна пам'ять, дисковий простір або пропускна спроможність каналів зв'язку. Якщо якийсь з ресурсів досягне максимального завантаження, ПЗ цілком буде недоступне. Атаки можуть бути спрямовані на будь-який з компонентів Web-забезпечення, наприклад, такі як сервер СКБД, сервер аутентифікації і так далі.

Приклад:

Припустимо, що сервер Health Care-генерує звіти про клінічну історії користувачів. Для генерації кожного звіту сервер запитує всі записи, що відповідають певним номером соціального страхування. Оскільки в базі містяться сотні мільйонів записів, користувачеві доводиться чекати результату декілька хвилин. У цей час завантаження процесора сервера СУБД досягає 60%.

Зловмисник може послати десять одночасних запитів на отримання звітів, що з високою ймовірністю призведе до відмови в обслуговуванні, оскільки завантаження процесора сервера баз даних досягне максимального значення. На час обробки запитів зловмисника нормальна робота сервера буде неможлива.

DoS на інший сервер

Зловмисник може розмістити на популярному Web-форумі посилання (наприклад, у вигляді зображення в повідомленні) на інший ресурс. При заході на форум, користувачі будуть автоматично завантажувати дані з атакуємого серверу на вказаний ресурс, використовуючи його ресурси. Якщо на атакуючому сервері використовується система запобігання атак з функцією блокування IP-адреси атакуючого, у посиланні може використовуватися сигнатура атаки (наприклад ../../../../../etc/passwd), що призведе до блокування користувачів, що зайшли на форум.

Атаки на сервер СУБД

Зловмисник може скористатися впровадженням коду SQL для видалення даних з таблиць, що призведе до відмови в обслуговуванні програми[4,13].

в) Недостатня протидія автоматизації (Insufficient Anti-automation)

Недостатня протидія автоматизації виникає, коли сервер дозволяє автоматично виконувати операції, які повинні проводитися вручну. Для деяких функцій програмного забезпечення необхідно реалізовувати захист від автоматичних атак. Автоматизовані програми можуть варіюватися від нешкідливих роботів пошукових систем до систем автоматизованого пошуку вразливостей і реєстрації облікових записів. Подібні роботи генерують тисячі запитів в хвилину, що може привести до падіння продуктивності усього Web-серверу. Протидія автоматизації полягає в обмеженні можливостей подібних програмних засобів.

Наприклад, файл robots може запобігати індексування деяких частин сервера, а додаткові затрати ідентифікації запобігати автоматичну реєстрацію сотень облікових записів системи електронної пошти[4].

г) Недостатня перевірка процесу (Insufficient Process Validation)

Вразливості цього класу виникають, коли сервер недостатньо перевіряє послідовність виконання операцій ПЗ. Якщо стан сесії користувача і ПЗ належним чином не контролюється, ПЗ може бути вразливий для шахрайських дій. В процесі доступу до деяких функцій ПЗ очікується, що користувач виконає ряд дій в певному порядку. Якщо деякі дії виконуються невірно або в неправильному порядку, виникає помилка, що призводить до порушення цілісності. Прикладами подібних функцій виступають переведення, відновлення паролів, підтвердження купівлі, створення облікового запису і т.д. В більшості випадків ці процеси складаються з ряду послідовних дій, здійснюваних в чіткому порядку.

Для забезпечення коректної роботи подібних функцій Web-забезпечення повинно чітко відстежувати стан сесії користувача і її відповідність поточним операціям. В більшості випадків це здійснюється шляхом збереження стану сесії в cookie або прихованому полі форми HTML. Але оскільки ці значення можуть бути модифіковані користувачем,обов'язково повинна проводитися перевірка цих значень на сервері. Якщо цього не відбувається, зловмисник дістає можливість обійти послідовність дій, тобто, логіку програми.

Приклад:

Система електронної торгівлі може пропонувати знижку на продукт В, у випадку купівлі продукту А. Користувач, який не хоче купувати продукт А, може спробувати придбати продукт В зі знижкою. Заповнивши замовлення на купівлю обох продуктів, користувач отримає знижку. Потім користувач повертається до форми підтвердження замовлення і видаляє продукт А, шляхом модифікації значень у формі. Якщо сервер повторно не перевірить можливість покупки продукту В за вказаною ціною без продукту А, буде здійснено закупівлю за низькою ціною[7,9].

1.6.4 Розголошування інформації

Атаки цього класу спрямовані на отримання додаткової інформації програмного Web-забезпечення. Використовуючи ці вразливості, зловмисник може визначити ПО, що використовується, номери версій клієнта і сервера і встановлені оновлення. У інших випадках може бути отримана інформація, котра може містити шлях розташування тимчасових файлів або резервних копій. У багатьох випадках ці дані не вимагаються для роботи користувачів.

Більшість серверів надають доступ до надмірного об'єму даних, проте необхідно мінімізувати об'єм службової інформації. Чим більшими знаннями про програму буде знати зловмисник, тим легше йому буде скомпрометувати систему.

а) Індексування директорій (Directory Indexing)

Наданням списку файлів в директорії є нормальною поведінкою Web-сервера, якщо сторінка, що відображується за умовчанням (index.html/home.html/default.htm), відсутня.

Коли користувач запрошує основну сторінку Web-сайту, він зазвичай вказує доменне ім'я сервера без імені конкретного файлу. Сервер переглядає основну директорію, знаходить в ній файл, використовуваний за умовчанням, і на його основі генерує відповідь. Якщо такий файл відсутній, як відповідь може повернутися список файлів в директорії сервера.

Ця ситуація аналогічна виконанню команди «ls» (Unix) або «dir» (Windows) на сервері і форматуванню результатів у вигляді HTML. В цьому випадку зловмисник може дістати доступ до даних, не призначених для вільного доступу.

Досить часто адміністратори покладаються на «безпеку через приховання» припускаючи, що раз гіперпосилання на документ відсутнє, то він недоступний необізнаним. Сучасні сканери вразливостей, такі як Nikto, можуть динамічно додавати файли і теки до списку сканованих, залежно від результатів запитів. Використовуючи вміст /robots.txt або отриманого списку директорій, сканер може знайти захований вміст або інші файли.

Таким чином зовні безпечне індексування директорій може привести до витоку важливої інформації, яка надалі буде використана для проведення атак на систему.

Приклад:

Використовуючи індексування директорій, можна дістати доступ до наступних даних:

- резервні копії (.bak, .old, .orig);
- тимчасові файли. Такі файли повинні видалятися сервером автоматично, але іноді залишаються доступними;
- приховані файли, назва яких починається з символу «.»;
- угода про імена. Ця інформація може допомогти передбачити імена файлів або директорій (admin або Admin, back - up або backup);
- перелік користувачів сервера. Дуже часто для кожного з користувачів створюється директорія з ім'ям, заснованим на назві облікового запису;
- імена файлів конфігурації (.conf, .cfg, .config);
- вміст серверних сценаріїв або виконуваних файлів у разі невірно вказаних розширень або дозволів.

Можуть бути використані три основні сценарії отримання списку файлів Web-сервера:

1 Помилки конфігурації. Подібні проблеми виникають, коли адміністратор помилково вказує в конфігурації сервера цю опцію. Подібні ситуації часто

виникають при налаштуванні складних конфігурацій, де деякі директорії мають бути доступні для перегляду;

2 Деякі компоненти Web-сервера дозволяють отримувати список файлів, навіть якщо це не дозволено в конфігураційних файлах. Зазвичай це виникає в результаті помилок реалізації, коли сервер генерує список файлів при отриманні певного запиту;

3 Бази цих пошукових машин (Google, Wayback machine) можуть містити кеш старих варіантів сервера, включаючи списки файлів[4].

б) Ідентифікація додатків (Web Server/Application Fingerprinting)

Визначення версій програмного забезпечення використовується зловмисником для отримання інформації про використовуваних сервером і клієнтом операційних системах, Web-серверах та Інтернет-броузерах. Також ця атака може бути спрямована на інші компоненти програмного Web-забезпечення, наприклад службу каталогу або сервер баз даних або використовувані технології програмування. Зазвичай подібні атаки здійснюються шляхом аналізу різної інформації, що надається Web-сервером.

Зазвичай подібні атаки здійснюються шляхом аналізу різної інформації, що надається Web-сервером, наприклад:

- Особливості реалізації протоколу HTTP;
- Заголовки HTTP-відповідей;
- Використовувані сервером розширення файлів (. Asp або. Jsp);
- Значення Cookie (ASPSESSION і т.д.);
- Повідомлення про помилки;
- Структура каталогів і використовуване угоду про імена (Windows / Unix);
- Інтерфейси підтримки розробки Web-додатків (Frontpage / WebPublisher);
- Інтерфейси адміністрування сервера (iPlanet / Comanche);
- Визначення версій операційної системи.

Для визначення версій клієнтського програмного забезпечення зазвичай використовується аналіз HTTP-запитів (порядок дотримання заголовків, значення User-agent і так далі). Проте для цих цілей може застосовуватися і інша техніка. Так, наприклад, аналіз заголовків поштових повідомлень, створених за допомогою клієнта Microsoft Outlook, дозволяє визначити версію встановленого на комп'ютері Інтернет-броузеру Internet Explorer. Наявність детальної і точної інформації про використовувані програмного забезпечення дуже важлива для зловмисника, оскільки реалізція багатьох атак (наприклад переповнювання буфера) специфічно для кожного варіанту операційної системи або програмного забезпечення. Крім того, детальна інформація про інфраструктуру дозволяє понизити кількість помилок[13].

в) Витік інформації (Information Leakage)

Ці вразливості виникають в ситуаціях, коли сервер публікує важливу інформацію, наприклад, коментарі розробників або повідомлення про помилки, яка може бути використана для компрометації системи. Цінні з точки зору зловмисника дані можуть міститися в коментаріях HTML, повідомленнях про помилки або просто бути присутнім у відкритому вигляді. Існує величезна кількість ситуацій, в яких може статися просочування інформації. Вона не обов'язково призводить до виникнення вразливості, але часто дає зловмиснику інформацію до подальшої побудови атаки. З просочуванням важливої інформації можуть виникати ризики різної міри, тому необхідно мінімізувати кількість службової інформації, доступної на клієнтській стороні.

Аналіз доступної інформації дозволяє зловмисникові провадити розвідку і отримати уявлення про структуру директорій сервера, використовуваних SQL-запитах, назвах ключових процесів і програм сервера.

Часто розробники залишають коментарі в HTML-сторінках і коді сценаріїв для полегшення пошуку помилок і підтримки програмного забезпечення. Ця інформація може варіюватися від простих описів деталей функціонування програми до (у гірших випадках) імен користувачів і паролів, використовуваних

при відладці. Просочування інформації може відноситися і до конфіденційних даних, оброблюваним сервером. Це можуть бути ідентифікатори користувача (ІНН, номери водійських посвідчень, паспортів і т.д.), а також поточна інформація (баланс особового рахунку або історія платежів).

Багато атак цієї категорії виходять за рамки захисту програмного Web-забезпечення і переходять в область фізичної безпеки. Просочування інформації в цьому випадку часто виникає коли, в Інтернет-броузері відображується інформація, яка не повинна виводитися у відкритому виді навіть користувачеві.

г) Зворотний шлях в директоріях (Path Traversal)

Ця техніка атак спрямована на отримання доступу до файлів, директорій і команд, що знаходяться поза основною директорією Web-сервера. Зловмисник може маніпулювати параметрами URL з метою отримати доступ до файлів або виконати команди, що розташовуються у файловій системі Web-сервера. Для подібних атак потенційно уразливий будь-який пристрій, що має Web-інтерфейс. Багато Web-серверів обмежують доступ користувача певною частиною файлової системи, зазвичай званої Web document root або CGI root. Ці директорії містять файли, призначені для користувача, і програми, необхідні для отримання доступу до функцій Web-забезпечення. Більшість базових атак, що експлуатують зворотний шлях, засновані на впровадженні в URL символів «./» для того щоб змінити розташування ресурсу, який оброблятиметься сервером. Оскільки більшість Web-серверів фільтрують цю послідовність, зловмисник може скористатися альтернативними кодуваннями для представлення символів переходу по директоріях. Популярні прийоми включають використання альтернативних кодувань, наприклад Unicode («.%u2216» або «.%c0%af»), використання зворотного слешу («.\») в Windows-серверах, символів URLEncode («%2e% 2e% 2f») або подвійного кодування URLEncode («.%255c»). Навіть якщо Web-сервер обмежує доступ до файлів певним каталогом, ця уразливість може виникати в сценаріях або CGI-програмах. Можливість використання зворотного шляху в каталогах досить часто виникає в додатках, що використовують

механізми шаблонів чи завантажують текст їх сторінок з файлів на сервері. У цьому варіанті атаки зловмисник модифікує ім'я файлу, що передається як параметр CGI - програми або серверного сценарію. В результаті зловмисник може отримати початковий код сценаріїв. Досить часто до імені файлу, що запитується, додаються спеціальні символи - такі, як «%00» - з метою обходу фільтрів.

д) Передбачуване розташування ресурсів (Predictable Resource Location)

Передбачуване розташування ресурсів дозволяє зловмисникові дістати доступ до прихованих даних або функціональних можливостей. Шляхом підбору зловмисник може дістати доступ до вмісту, не призначеного для публічного перегляду. Тимчасові файли, файли резервних копій, файли конфігурації або стандартні приклади часто є метою подібних атак. В більшості випадків перебір може бути оптимізований шляхом використання стандартної угоди про імена файлів і директорій сервера. Отримувані зловмисником файли можуть містити інформацію про дизайн додатка, інформацію з баз даних, імена машин або паролі, шляхи до директорій. Приховані файли також можуть містити вразливості, відсутні в основному застосуванні.

Приклад:

Атакуючий може створити запит до будь-якого файлу або папці на сервері.

Наявність або відсутність ресурсу визначається за кодом помилки (наприклад, 404 у разі відсутності папки або 403 в разі її наявності на сервері). Нижче наведено варіанти подібних запитів.

Сліпий пошук популярних назв директорій:

/admin/

/backup/

/logs/

/vulnerable_file.cgi

Зміна розширень існуючого файлу: (/test.asp)

/test.asp.bak

/test.bak

/test

1.6.5 Атаки на клієнтів (Client-side Attacks)

Атаки цього типу направлені на користувачів Web-сервера Під час відвідування сайту між користувачем і сервером встановлюються довірчі «стосунки», як в технологічному, так і в психологічному аспектах. Експлуатуючи цю довіру, зловмисник може використовувати різні методи для проведення атак на клієнтів сервера.

а) Підміна вмісту (Content Spoofing)

Використовуючи цю техніку, зловмисник змушує користувача повірити, що сторінка згенерована Web-сервером, а не передана з зовнішнього джерела.

Деякі Web-сторінки створюються з використанням динамічних джерел HTML- коду. Приміром, розташування фрейму (<frame src=" http://foo.example/file.html">) може передаватися у параметрі URL (http://foo.example/page?frame_src=http://foo.example /file.html). Атакуючий може замінити значення параметра "frame_src" на "frame_src =

http://attacker.example/spoof.html". Коли буде відображатися результуюча сторінка, у рядку адреси браузера користувача відображатиметься адреса сервера (foo.example), але так само на сторінці буде присутній вміст із зовнішнього джерела, завантажене з сервера атакуючого (attacker.example), замасковане під легальний контент.

Спеціально створене посилання може бути надіслана електронною поштою, системі моментального обміну повідомленнями, опублікована на дошці повідомлень або відкрита в браузері користувача з використанням міжсайтового виконання сценаріїв.

Якщо атакуючий спровокував користувача на перехід по спеціально створеному посиланню, у користувача може скластися враження, що він

переглядає дані з сервера, в той час як частина їх була сгенерірованна зловмисником.

Таким чином, станеться дефейс (груба підміна вмісту) сайту http://foo.example на стороні користувача, оскільки вміст сервера буде додано з сервера http://attacker.example. Ця атака так само може використовуватися для створення помилкових сторінок, таких як форми введення пароля, прес-релізи.

б) Міжсайтове виконання сценаріїв (Cross-site Scripting, XSS)

Наявність цієї вразливості дозволяє зловмиснику передати серверу виконуваний код, який буде перенаправлений Інтернет-броузеру користувача. Для написання подібного коду зазвичай використовуються HTML/JavaScript, але можуть бути застосовані і VBScript, ActiveX, Java, Flash та ін. Переданий код виконується в контексті безпеки (чи зоні безпеки) уразливого сервера. Використовуючи поточні привілеї, код дістає можливість читати, модифікувати або передавати важливі дані, доступні за допомогою Інтернет-броузеру. При цьому виді атаки у атакованого користувача може бути скомпрометований аккаунт (крадіжка cookie), його

Інтернет-броузер може бути перенаправлений на інший сервер або здійснена підміна вмісту сервера. В результаті ретельно спланованої атаки зловмисник може використовувати Інтернет-броузер жертви для перегляду сторінок сайту від імені користувача, що атакується. Передача коду в таких випадках здійснюється через URL, в заголовках HTTP-запиту (cookie, user-agent, refferer), значеннях полів форм і так далі. Існує два типи атак, що призводять до міжсайтового виконання сценаріїв: постійні (збережені) і непостійні (відбиті). Основною відмінністю між ними є те, що у відбитому варіанті передача коду серверу і повернення його клієнтові здійснюється у рамках одного HTTP-запиту, а в збереженому - в різних. Здійснення непостійної атаки вимагає, щоб користувач перейшов по посиланню, яке сформоване зловмисником (посилання може бути передане по e-mail, ICQ і так далі). В процесі завантаження сайту код, впроваджений в URL або заголовки запиту, буде переданий клієнтові і виконаний

в його Інтернет-броузері. Збережений різновид уразливості виникає, коли код передається серверу і зберігається на ньому на деякий проміжок часу. Найбільш популярними цілями атак в цьому випадку є форуми, пошта з Web-інтерфейсом і чати. Для атаки користувачеві не обов'язково переходити по посиланню, досить відвідати уразливий сайт.

Приклад:

Збережений варіант атаки

Багато сайтів мають дошки оголошень і форуми, які дозволяють користувачам залишати повідомлення. Зареєстрований користувач зазвичай ідентифікується за номером сесії, що зберігається в cookie. Якщо атакуючий залишить повідомлення, що містить код на мові JavaScript, він отримає доступ до ідентифікатора сесії користувача.

Приклад коду для передачі cookie:

<SCRIPT> Document.location = 'http://attackerhost.example/cgi-bin/cookiesteal.cgi?' + Document.cookie </ SCRIPT>

в) Розщеплення HTTP-запиту (HTTP Response Splitting)

При використанні даної уразливості зловмисник посилає серверу спеціальним чином сформований запит, відповідь на який інтерпретується метою атаки як дві різні відповіді. Друга відповідь повністю контролюється зловмисником, що дає йому можливість підробити відповідь сервера.

У реалізації атак з розщепленням HTTP-запиту беруть участь як мінімум три сторони:

– Web-сервер, який містить подібну уразливість.

– Мета атаки, що взаємодіють з Web-сервером під управлінням зловмисника. Типово в якості мети атаки виступає кешуючий сервер-посередник або кеш браузера.

– Атакуючий, який ініціює атаку.

Можливість здійснення атаки виникає, коли сервер повертає дані, надані користувачем в заголовках HTTP відповіді. Зазвичай це відбувається при пере

направлення користувача на іншу сторінку (коди HTTP 3xx) або коли дані, отримані від користувача, зберігаються в cookie.

У першій ситуації URL, на який відбувається перенаправлення, є частиною заголовка Location HTTP відповіді, а в другому випадку значення cookie передається в заголовку Set-Cookie.

Основою розщеплення HTTP-запиту є впровадження символів переведення рядка (CR і LF) таким чином, щоб сформувати дві HTTP транзакції, в той час як реально буде відбуватися тільки одна. Переклад рядка використовується для того, що б закрити першу (стандартну) транзакцію, і сформувати другу пару питання / відповідь, повністю контрольовану зловмисником і абсолютно непередбачувану логікою програми.

У результаті успішної реалізації цієї атаки зловмисник може виконати наступні дії:

– Міжсайтового виконання сценаріїв.

– Модифікація даних кеша сервера-посередника. Деякі кешуючий сервери-посередники (Squid 2.4, NetCache 5.2, Apache Proxy 2.0 і ряд інших), зберігають підроблений зловмисником відповідь на жорсткому диску і на подальші запити користувачів за цією адресою повертають кешовані дані.

Це призводить до заміни сторінок сервера на стороні клієнта. Крім цього, зловмисник може переправити собі значення Cookie користувача або присвоїти їм певне значення. Так само ця атака може бути спрямована на індивідуальний кеш браузера користувача.

– Міжкористувацька атака (один користувач, одна сторінка, тимчасова підміна сторінки). При реалізації цієї атаки зловмисник не посилає додатковий запит. Замість цього використовується той факт, що деякі сервери-посередники поділяють одне TCP-з'єднання до сервера між декількома користувачами. У результаті другий користувач отримує у відповідь сторінку, сформовану зловмисником. Крім підміни сторінки зловмисник може також виконати різні операції з cookie користувача.

– Перехоплення сторінок, що містять дані користувача. У цьому випадку зловмисник отримує відповідь сервера замість самого користувача. Таким чином, він може отримати доступ до важливої або конфіденційної інформації[4,8,13].

1.6.6 Виконання коду (Command Execution)

Атаки цього класу спрямовані на виконання коду на Web-сервері Всі сервери використовують дані, передані користувачем при обробці запитів. Часто ці дані використовуються при складанні команд, вживаних для генерації динамічного вмісту. Якщо при розробці не враховуються вимоги безпеки, зловмисник дістає можливість модифікувати виконавчі команди.

а) Переповнення буфера (Buffer Overflow)

Експлуатація переповнення буфера дозволяє зловмисникові змінити шлях виконання програми шляхом перезапису даних у пам'яті системи. Переповнення буфера є найбільш поширеною причиною помилок у програмах. Воно виникає, коли обсяг даних перевищує розмір виділеної під них буфера. Коли буфер переповнюється, дані переписують інші області пам'яті, що призводить до виникнення помилки. Якщо зловмисник має можливість управляти процесом переповнення, це може викликати ряд серйозних проблем.

Переповнення буфера може викликати відмови в обслуговуванні, приводячи до пошкодження пам'яті і викликаючи помилки в програмах. Більш серйозні ситуації дозволяють змінити шлях виконання програми і виконати в її контексті різні дії. Це може відбуватися в кількох випадках.

Використовуючи переповнення буферу, можна перезаписувати службові області пам'яті, наприклад, адресу повернення з функцій у стеці. Також, при переповненні можуть бути переписані значення змінних у програмі.

Переповнення буфера є найбільш поширеною проблемою в безпеці і нерідко зачіпає Web-сервери. Проте атаки, що експлуатують цю уразливість,

використовуються проти Web-додатків не дуже часто. Причиною цього є те, що атакуючому, як правило, необхідно проаналізувати вихідний код або образ програми. Оскільки атакуючому доводиться експлуатувати нестандартну програму на віддаленому сервері, йому доводиться атакувати "всліпу", що знижує шанси на успіх.

На рисунку 1.3 зображена схема атаки “ переповнення буфера”.

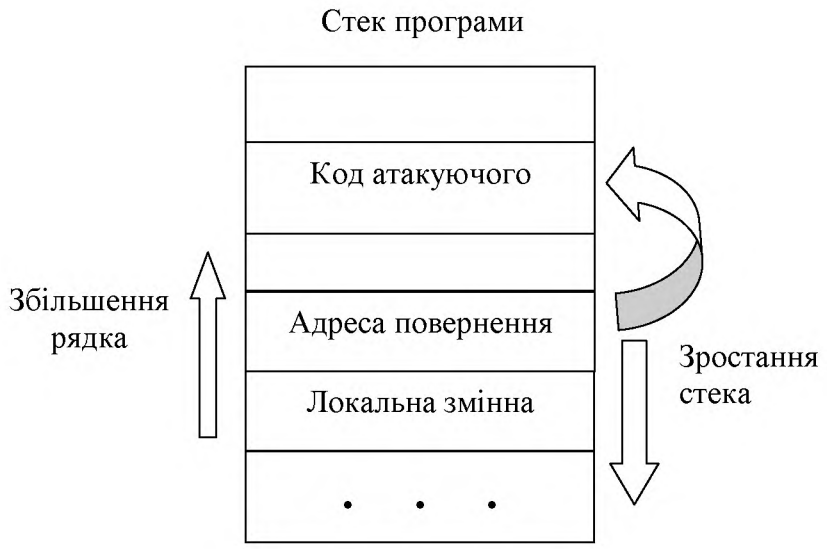


Рисунок 1.3 — Схема атаки “ переповнення буфера”

б) Атака на функції форматування рядків (Format String Attack)

При використанні цих атак шлях виконання програми модифікується методом перезапису областей пам'яті за допомогою функцій форматування символьних змінних. Уразливість виникає, коли призначені для користувача дані застосовуються як аргументи функцій форматування рядків - таких, як fprintf, printf, sprintf, setproctitle, syslog і так далі. Якщо зловмисник передає додатку рядок, що містить символи форматування ("%f", "%p", "%n" і так далі), то у нього з'являється можливість:

- виконати довільний код на сервері;
- прочитати значення із стека;
- викликати помилки в програмі/відмову в обслуговуванні.

в) Впровадження операторів LDAP (LDAP Injection)

Атаки цього типу спрямовані на Web-сервери, які створюють запити до служби LDAP на основі даних, що вводяться користувачем. Спрощений протокол доступу до служби каталогу (Lightweight Directory Access Protocol, LDAP) - відкритий протокол для створення запитів і управління службами каталогу сумісними зі стандартом X.500.

Протокол LDAP працює поверх транспортних протоколів Internet (TCP/UDP). Web-забезпечення може використовувати дані, надані користувачем для створення запитів по протоколу LDAP при генерації динамічних Web-сторінок. Якщо інформація отримана від клієнта, належним чином не веріфікується, то зловмисник дістає можливість модифікувати LDAP-запит. Слід зауважити, що запит виконуватиметься з тим же рівнем привілеїв, з яким працює компонент ПЗ, виконуючий запит (сервер СКБД, Web-сервер і т. д). Якщо цей компонент має права на читання або модифікацію даних в структурі каталогу, зловмисник дістає ті ж можливості[7,8].

г) Виконання команд ОС (OS Commanding)

Атаки цього класу спрямовані на виконання команд операційної системи на Web-сервері шляхом маніпуляції вхідними даними. Якщо інформація, отримана від клієнта, належним чином не веріфікується, то зловмисник дістає можливість виконати команди ОС. Вони виконуватимуться з тим же рівнем привілегій, з яким працює компонент ПЗ, виконуючий запит (сервер СКБД, WEB-сервер і т. д).

Програмні Web-забезпечення часто використовують параметри, які вказують на те, який файл відображувати або використовувати як шаблон. Якщо цей параметр не перевіряється досить ретельно, то зловмисник може підставити свої команди ОС до запиту.

Більшість мов сценаріїв дозволяють запускати команди ОС під час виконання, використовуючи варіанти функції exec. Якщо дані, отримані від користувача передаються цій функції без перевірки, зловмисник може виконати команди ОС на відстані.

д) Впровадження операторів SQL (SQL Injection)

Ці атаки спрямовані на Web-сервери, які створюють SQL-запити до серверів СКБД на основі даних, що вводяться користувачем.

Мова запитів SQL є спеціалізованою мовою програмування, що дозволяє створювати запити до серверів СКБД. Більшість серверів підтримують цю мову у варіантах, стандартизованих ISO і ANSI. У більшості сучасних СКБД присутні розширення діалекту SQL, специфічні для цієї реалізації (T-SQL в Microsoft SQL Server, -PL SQL в Oracle і т. д.). Багато програмного Web-забезпечення використовує дані, передані користувачем, для створення динамічних Web-сторінок.

Якщо інформація, отримана від клієнта, належним чином не веріфікується, то зловмисник дістає можливість модифікувати запит до SQL-серверу, що відправляється ПЗ. Запит виконуватиметься з тим же рівнем привілегій, з яким працює компонент ПЗ, виконуючий запит (сервер СКБД, Web-сервер і т. д). В результаті зловмисник може отримати повний контроль над сервером СКБД і навіть його операційною системою.

Зазвичай виділяють два методи експлуатації впровадження операторів SQL: звичайна атака і атака всліпу. У першому випадку зловмисник підбирає параметри запиту, використовуючи інформацію про помилки, які згенеровані програмним Web-забезпеченням. У другому випадку стандартні повідомлення про помилки модифіковані, і сервер повертає зрозумілу для користувача інформацію про неправильне введення. Здійснення SQL Injection можливо і в цій ситуації, проте вивлення уразливості ускладнене. Найбільш поширений метод перевірки наявності проблеми - додавання виразів, що повертають істинне і помилкове значення[10-13].

е) Впровадження серверних розширень (SSI Injection)

Атаки цього класу дозволяють зловмисникові передати виконавчий код, який надалі буде виконаний на Web-сервері. Вразливості, що призводять до можливості здійснення цих атак, полягають у відсутності перевірки даних, наданих користувачем перед збереженням їх у файлі, що інтерпретується

сервером. Перед генерацією HTML-сторінки сервер може виконувати сценарії, наприклад SSI. У деяких ситуаціях початковий код сторінок генерується на основі даних, наданих користувачем. Якщо зловмисник передає серверу оператори SSI, він може дістати можливість виконання команд операційної системи або включити в неї заборонений вміст при наступному відображенні.

є) Впровадження операторів XPath (XPath Injection)

Ці атаки спрямовані на Web-сервера, які створюють запити на мові XPath на основі даних, що вводяться користувачем. Мова XPath 1.0 розроблена для на-

дання можливості звернення до частин документу на мові XML. Він може бути використаний безпосередньо або як складова частина XSLT-перетворення XML-документів, або як виконання запитів XQuery. Синтаксис XPath близький до мови SQL-запитів.

Якщо запити XPath генеруються під час виконання на основі введених даних користувача, то у зловмисника з'являється можливість модифікувати запит з метою обходу логіки роботи програми[10,12].

1.7 Загальні відомості про браузери

Браузер – програмне забезпечення для комп'ютера або іншого електронного пристрою, як правило, під'єднаного до Інтернету, що дає можливість користувачеві взаємодіяти з текстом, малюнками або іншою інформацією на гіпертекстовій веб-сторінці. Тексти та малюнки можуть містити посилання на інші веб-сторінки, розташовані на тому ж веб-сайті або на інших веб-сайтах. Веб-переглядач з допомогою посилань дозволяє користувачеві швидко та просто отримувати інформацію, розміщену на багатьох веб-сторінках.

1.7.1 Принцип роботи браузера

Веб-переглядач під'єднується до сервера HTTP, отримує з нього документ і форматує його для представлення користувачеві або намагається викликати зовнішню програму, яка це зробить, залежно від формату документа. Формати документа, які веб-переглядач повинен представляти без допомоги зовнішніх програм, визначає World Wide Web Consortium. До них належать формати

текстових документів HTML та XHTML, а також найпоширеніші формати растрової графіки GIF, JPEG та PNG.

Адресування сторінок відбувається за допомогою URL (Uniform Resource Locator, RFC 1738), який інтерпретується, як адреса, що починається з http: для протоколу HTTP. Багато навігаторів також підтримують інші типи URL та їх відповідні протоколи, як, наприклад, gopher: для Gopher (ієрархічний протокол гіперпосилань), ftp: для протоколу перенесення файлів FTP, rtsp: для Протоколу потоків реального часу RTSP, та https: для HTTPS (HTTP Secure, що розширює HTTP за допомогою Secure Sockets Layer SSL або Transport Layer Security TLS).

1.8 Порівняння безпеки популярних Інтернет - браузерів

Кожен виробник інтернет-браузерів має свої широко декларовані переваги, які знаходять відгук у очах певної групи користувачів, і забезпечують того чи іншого браузера затятих прихильників і широку популярність. Індустрія браузерів існує, в основному, за рахунок непрямих джерел доходу. Всі популярні браузери або можна встановити безкоштовно, або ж вбудовані в ту або іншу операційну систему. Internet Explorer вбудований в Microsoft Windows, починаючи з Windows 98, а Safari інтегрований в Mac OS. Відповідно, конкурувати виробникам інтернет-браузерів з використанням економічних важелів впливу - неможливо.

Часто користувачі віддають перевагу того чи іншого браузера з-за гарного інтерфейсу, швидкості і зручності в роботі або наявності якихось розширень. Отже, в хід ідуть інші методи боротьби «за серце користувача» - з використанням таких, вельми претензійних, гасел як «найшвидший браузер», «найбільш зручний браузер», «найбільш функціональний браузер», «найбільш настроюється браузер» та інші. Часто вибір вашого браузера для повсякденної роботи - це справа багаторічної звички або сліпа віра рекламі виробника браузера, або віра в ідеали в області розвитку вільного Інтернету, які ставлять перед собою виробники браузеров, або авторитетну думку знайомих фахівців, або навіть бажання постійно пробувати щось нове. Але при цьому часто забувається або спеціально замовчується про ступінь безпеки самого браузера. Адже безпосередньо через

браузер ми переглядаємо вміст веб-сайтів. Через браузер ми заходимо на сайти інтернет-банків, виробляємо оплату товарів і послуг, користуємося онлайн-сервісами або обмінюємося конфіденційною інформацією. Саме на браузер лягає первинна відповідальність за безпеку в мережі.

Розглянемо з точки зору безпеки п'ятірку найбільш популярних браузерів для платформи Microsoft Windows (в алфавітному порядку):

1 Apple Safari:

2 Google Chrome;

3 Microsoft Internet Explorer;

4 Mozilla Firefox;

5 Opera.

1.8.1 Існуючі технології безпеки браузера Apple Safari

Компанія Apple, кажучи про безпеку, в першу чергу, акцентує увагу на функції Безпечного перегляду. У даному режимі Safari не записує історію відвідуваних сайтів, завантаженого ПЗ і документів, не зберігає пошукові запити, cookies, і дані веб-форм. У Safari також присутні функції блокування спливаючих вікон. Схема налаштування безпеки браузера Apple Safari наведена на рисунку 1.4.

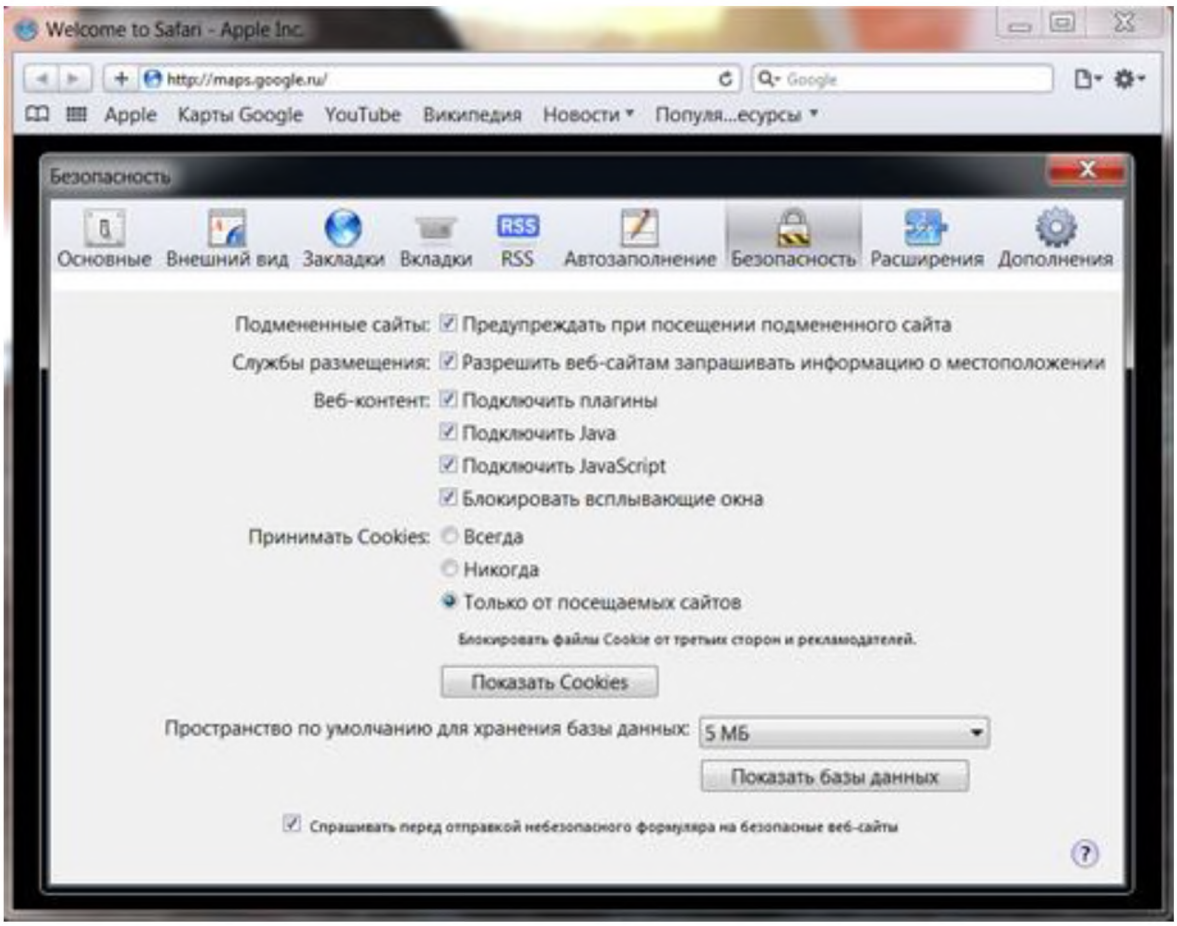


Рисунок 1.4 – Налаштування безпеки браузера Apple Safari

У Safari вбудовані і продовжують розвиватися технології, які протидіють атак з використанням міжсайтового скриптингу (XSS, Cross Site Scripting, така абревіатура використовується для виключення плутанини з абревіатурою Cascading Style Sheets (CSS) - каскадні таблиці стилів). Також вбудовані в браузер репутаційні технології блокування шкідливих сайтів: фішингових, шахрайських, а також сайтів, які розповсюджують шкідливі програми. Також вбудована підтримка EV-сертифікатів (Extended Validation), що дозволяє легко виділяти легітимні сайти.

Safari підтримує технології безпечного шифрування для запобігання перехоплення сеансів зв'язку, шахрайства при роботі в Інтернеті. Також підтримується аутентифікація на основі реєстрації на безпечних веб сайтах і найбільш популярні проксі-протоколи. Цікава також функція Безпечні завантаження, завдяки якій при першому відкритті кожного сайту відображається джерело, з якого була взята та чи інша сторінка.

1.8.2 Існуючі технології безпеки браузера Google Chrome

Схема налаштування безпеки браузера Google Chrome наведена на рисунку 1.5.

У цьому браузері існує захист від шахрайських і фішингових сайтів, зосереджена в технології «Безпечний перегляд».

Також виділяється функціональна можливість під назвою «пісочниця», за допомогою якого браузер може запобігти установку в систему шкідливих програм, а також має можливість відслідковувати вплив коду, який виконується в однієї вкладки браузера на вміст інших відкритих вкладок. У Chrome 12 з'явився фільтр шкідливих файлів на основі репутаційних технологій, що при подальшому розвитку може скласти конкуренцію технології Application Reputation від Microsoft.

Зокрема, у Google Chrome існує технологія забезпечення безперервності HTTPS-з'єднання і захисту його від компрометації, захист від XSS-атак і інші корисні функції.

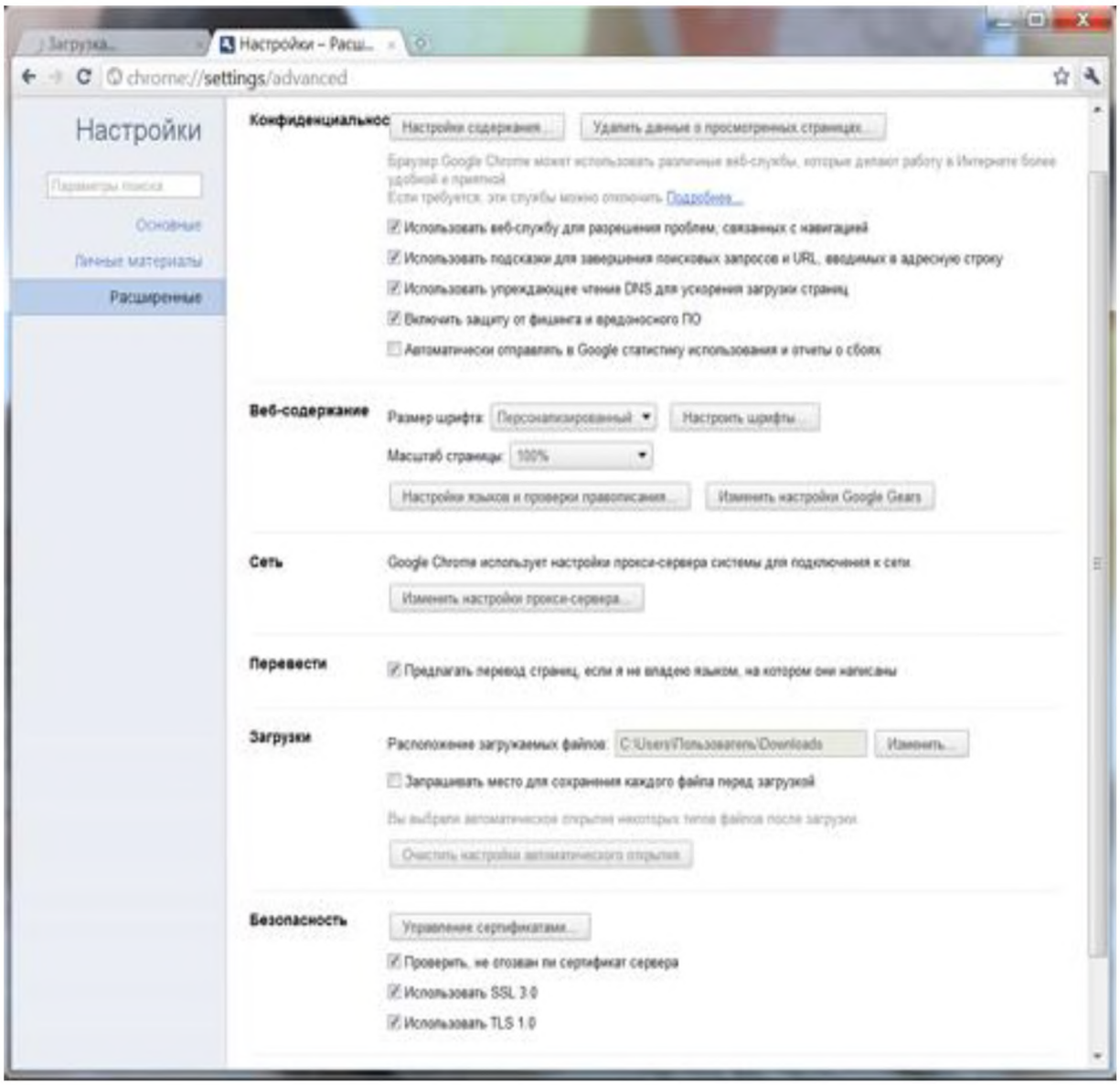


Рисунок 1.5 – Налаштування безпеки браузера Google Chrome

1.8.3 Існуючі технології безпеки браузера Microsoft Internet Explorer

Схема налаштування безпеки браузера Internet Explorer наведена на рисунку 1.6.

Компанія Microsoft, кажучи про безпеку свого браузера, в першу чергу робить упор на фільтрацію ActiveX-вмісту. У загальному-те, проблема небезпечного ActiveX-умісту актуальна саме для даного браузера, оскільки без додаткових плагінів в конкуруючих браузерах взаємодія з активним вмістом, розташованим на інтернет-сторінках, проводиться за допомогою інших технологій.

Також акцент робиться на протидію XSS-атакам, перегляд в приватному режимі InPrivate і функція захисту від стеження. Також реалізовано виділення доменів другого рівня в адресному рядку браузера, жирним кольором, що дозволяє легко визначити, чи користувач знаходиться на цьому сайті, на який хотів зайти, або ж на шахрайському, адреса якого сильно схожий на адресу цього сайту.

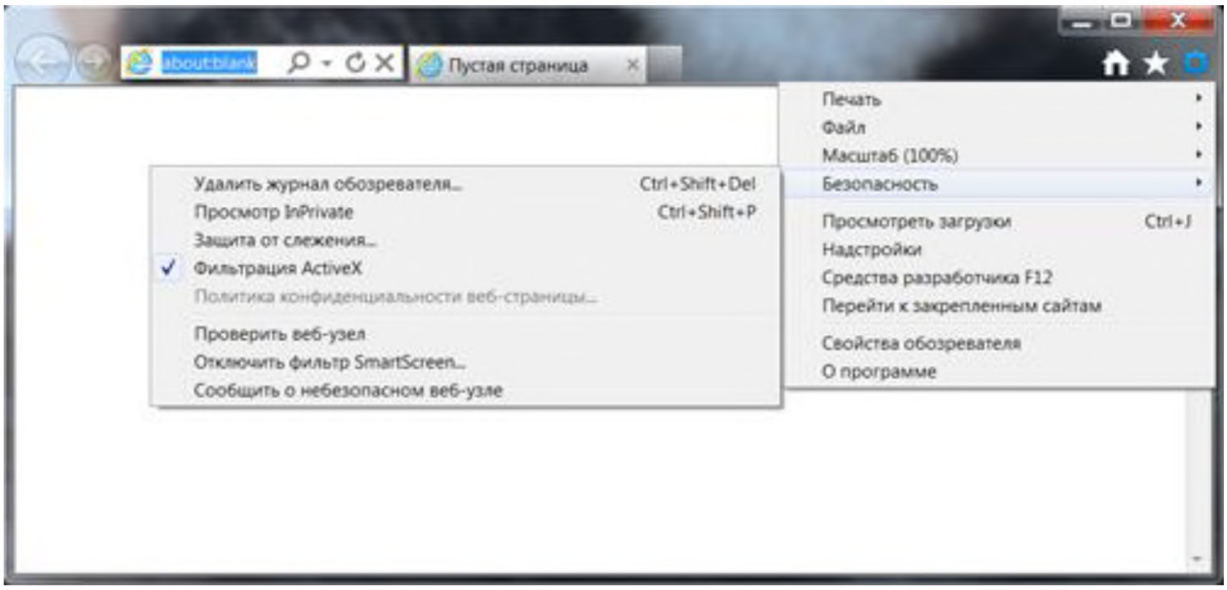


Рисунок 1.6 – Налаштування безпеки браузера Internet Explorer

Як унікальної функціональної особливості безпеки можна вказати широко рекламується фільтр SmartScreen, який в 9-ой версії Internet Explorer має можливість фільтрувати не тільки шкідливі сайти з URL, але і, власне, шкідливі файли за допомогою технології Application Reputation, яка заснована на репутаційних технологіях.

Слід зауважити, що актуальна версія Internet Explorer значно удосконалилася в плані підвищення стандартів інформаційної безпеки в порівнянні з попередніми версіями даного браузера, і його цілком можна рекомендувати для використання початківцям інтернет-користувачам для здійснення операцій інтернет банкінгу та інших потенційно-небезпечних операцій при чіткому дотриманні рекомендацій виробника.

1.8.4 Існуючі технології безпеки браузера Mozilla Firefox

Розробники браузера Firefox традиційно приділяють безпеки свого браузера пильну увагу.

Зокрема, це підтримка розширених EV-сертифікатів, захист від XSS-атак, інтеграція з Батьківським контролем Windows 7, функції «Приватний перегляд», інтеграція з антивірусними продуктами, фільтр шкідливих сайтів, захист від стеження за діями користувача в Інтернеті за допомогою спеціальних скриптів, що розміщуються на інтернет-сторінках, і підтримка HTTPS-сполук.

Схема налаштування безпеки браузера Mozilla Firefox наведена на рисунку 1.7.

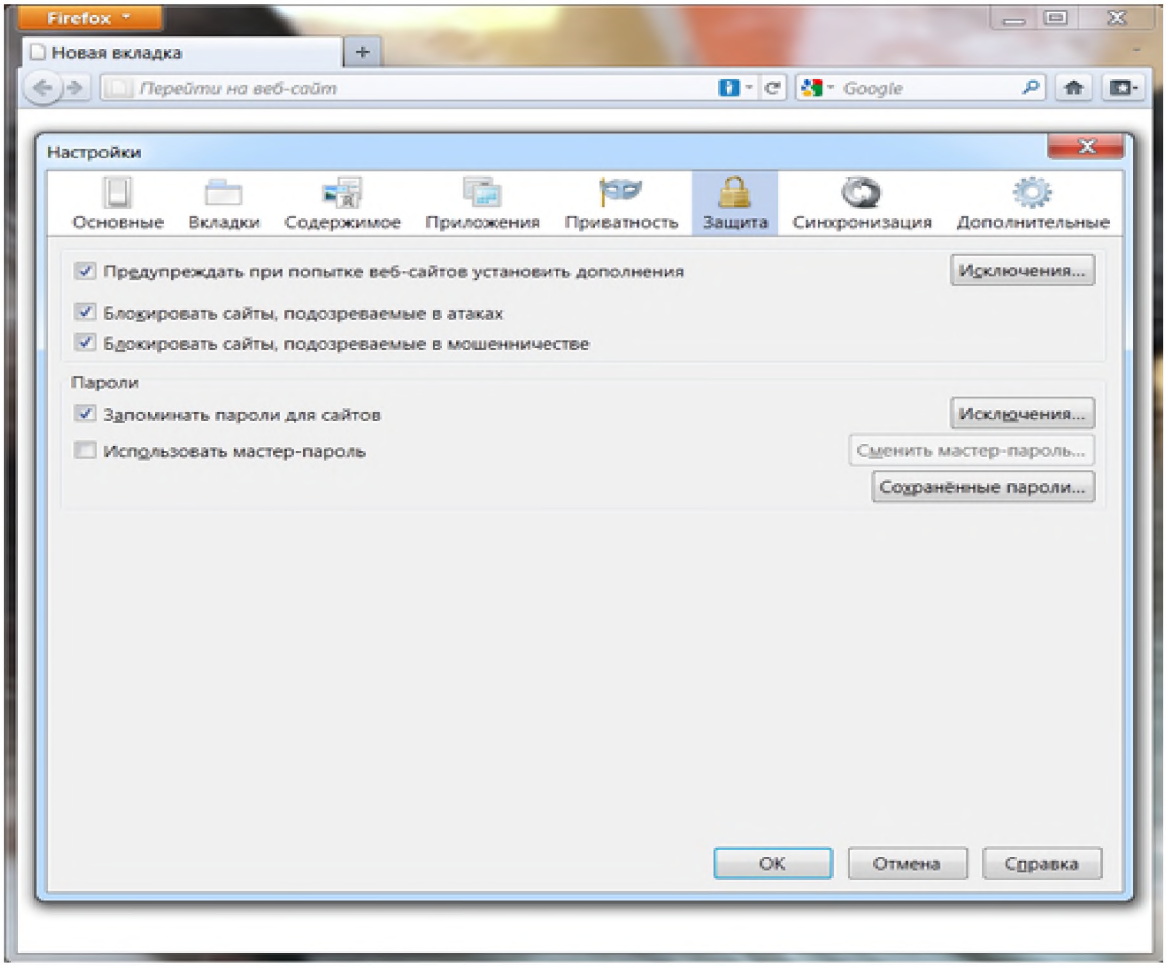


Рисунок 1.7 – Налаштування безпеки браузера Mozilla Firefox

При розгляді даного браузера слід також звернути увагу на те, що розробники роблять ставку на широке використання функціональних доповнень (розширень), які створюються сторонніми розробниками. За допомогою цих доповнень можна значно підвищити безпеку використання даного браузера.

Таким чином, браузер є ідеальним конструктором для користувачів, які володіють достатніми знаннями з інформаційної безпеки і точно знають, що хочуть отримати від браузера.

1.8.5 Існуючі технології безпеки браузера Opera

Розробники браузера Opera гранично лаконічні відносно питання безпеки браузера. Зокрема, заявляється про існування фільтру від шкідливих інтернет-сайтів, режим приватного перегляду, підтримки розширених сертифікатів сайтів, і управлінні завантажуваними cookies.

Схема налаштування безпеки браузера Opera наведена на рисунку 1.8.

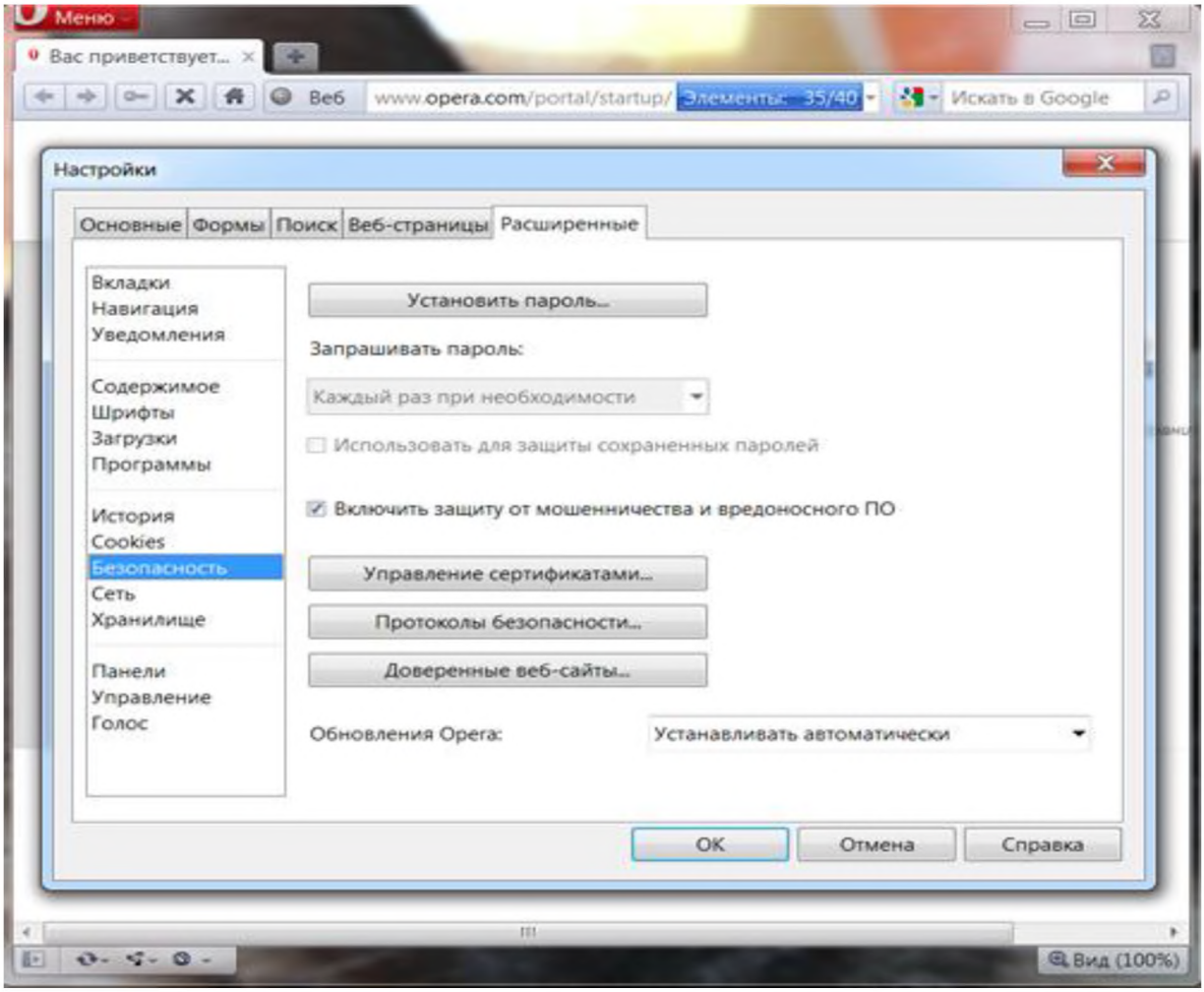


Рисунок 1.8 — Налаштування безпеки браузера Opera

1.8.6 Порівняння безпеки браузерів

Узагальнена схема наявності найбільш важливих на сьогоднішній день технологій безпеки веб-браузерів у наведена у таблиці 1.1. Дані, представлені в таблиці наводяться на основі інформації, наданої виробниками браузерів.

Таблиця 1.1– Наявність технологій безпеки в популярних веб-браузерах

Технології безпеки	Apple Safari	Google Chrome	Microsoft Internet Explorer	Mozilla Firefox	Opera
Автоматичне оновлення браузера	+	+	+	+	+
Підтримка HTTPS-з'єднань і візуалізація наявності безпечного з'єднання	+	+	+	+	+
Захист від компрометації HTTPS-з'єднання	-	+	Частково (pinned sites)	-	-
Підтримка EV-сертифікатів	+	+	+	+	+
Механізми захисту від XSS-атак	+	+	+	+	-
Фільтр шкідливих сайтів по URL	+	+	+	+	+
Фільтр шкідливого програмного забезпечення	-	+ (sandboxing і репутаційні технології перевірки чистоти файлів)	+ (SmartScreen - Application Reputation)	Частково (при використанні додатків, інтеграція з антивірусним пз)	-
Режим приватного перегляду	- (Режим «Приватний доступ»)	- (Режим анонімності)	- (Режим InPrivate)	- (Режим «Приватний перегляд»)	- (Режим приватності)
Захист від стеження	-	-	+ (Tracking Protection)	+	-
Використання	Виконуючий	Виконуючий	Виконуючий	Виконуючий	Виконуючий

ASLR	файл і DLL-бібліотеки	файл і DLL-бібліотеки	файл і DLL-бібліотеки	файл і DLL-бібліотеки	файл і DLL-бібліотеки

Розглянемо деякі нюанси приведеної таблиці.

Підтримка роботи з EV-сертифікатами, наявності режим приватного перегляду, а також можливості з'єднань з веб-вузлами через протокол HTTPS - все це реалізовано у всіх порівнюваних браузерах.

С захистом від компрометації HTTPS-з'єднання ситуація дещо гірше. З відомих технологій з цього приводу можна згадати лише можливість спостереження за безперервністю HTTPS-сполук у Google Chrome і закріплені сайти (pinned sites) у Internet Explorer 9 при використанні разом з Windows 7.

Ця функція заснована на тому, що користувачі в більшості випадків набирають в адресному рядку сайту лише його домен (наприклад, domain.com), без вказівки протоколу, за яким необхідно з'єднуватися (http:// або https://). У цьому випадку браузер спочатку з'єднується з веб-сервером по протоколу HTTP. Якщо сервер при цьому підтримує HTTPS-протокол, і на ньому налаштований автоматичний редірект на цей безпечний протокол, то лише тоді відбувається редірект з HTTP-протоколу HTTPS. На думку фахівців Microsoft, цього часу, що витрачається на редірект між HTTP і HTTPS-протоколу може бути достатньо для проведення атаки. Використання закріплених сайтів зручно лише для невеликого набору найбільш важливих сайтів. Тому в таблиці дана технологія для IE9 відзначена як підтримувана частково.

Фільтр шкідливих сайтів по URL до теперішнього часу також присутній в кожному поважному браузері, але подібний стан речей виник відносно недавно. Технологія фільтрації небезпечних сайтів по URL є у всіх браузерах, але якість роботи такого функціоналу безпосередньо залежить від використовуваних баз і якості фидбэка з користувачами, які беруть безпосередню участь у наповненні відповідних баз, розташованих в хмарах вендорів або їх партнерів. Наприклад, Firefox для блокування шкідливих сайтів користується хмарами Google. Фактично, інформація про нових шкідливих сайтах, які можна надіслати за допомогою Firefox, відправляється в компанію Google, і бразуер користується

відповідними репутаційними технологіями. Інші браузери використовують власні репутаційні технології, ефективність яких може істотно відрізнятися.

Окремо варто сказати про захист від установки в систему шкідливих програм, по суті, про антивірусному функціоналі на рівні браузера. Вона реалізована тільки в Internet Explorer 9. Фільтр SmartScreen, вбудований в цей браузер, оцінює репутацію для закачуваних файлів з інтернету. У Google Chrome реалізована пісочниця, яка отримала підтримку у вигляді нового компонента, що відповідає за перевірку файлів, що завантажуються на шкідливість з допомогою репутаційних технологій і вже виглядає як повноцінна технологія фільтрації шкідливих програм, які завантажуються засобом браузера. Можливості Mozilla Firefox заслуговує лише половину результату.

Варто сказати, що репутаційні технології перевірки файлів, реалізовані в Google Chrome на поточний момент виглядають сирими. Проведемо невеликий експеримент, задавши пошуковій системі запит «аватар завантажити на великій швидкості» і через кілька кликів знайдемо свіжу модифікацію лже-архіву, що вимагає за «розпаковування» надіслати гроші зловмисникам. IE9 вивів повідомлення про те, що файл «завантажується незвичайним чином». Chrome теж кілька секунд перевіряв файл у своєму «хмарі», але повідомлення про те, що файл не підписаний і у нього немає видавця, вивела операційна система, а не браузер.

Що стосується автоматичного оновлення браузерів, то зазвичай під цим мається на увазі установка нових мінорних версій, які закривають знайдені уразливості і підвищують стабільність веб-клієнтів. Автоматичний перехід на нову версію браузерів звичайно пов'язаний з явною вказівкою на подібне бажання користувача. У зв'язку з цим інтернет-користувачам можна порадити погоджуватися на такі пропозиції, хоча це і може призвести до того, що доведеться звикати до нового зовнішнього вигляду та функціоналу популярного браузера. Також не завадить стежити за новинами (або підписатися на них), які публікуються на офіційному сайті виробника браузера.

Позитивним моментом є підтримка технології ASLR (Address Space Layout Randomization, рандомізація розміщення адресного простору) всій п'ятіркою

популярних браузерів в реалізації для платформи Windows як для виконуваного файлу, так і для довантажує DLL-бібліотек. Це говорить, що розробники веб-браузерів для Windows дотримуються рекомендацій Microsoft при створенні безпечних додатків.

Що стосується нової функціональної можливості, що входить до підсистеми безпеки інтернет-браузера, під назвою «захист від стеження», то вона заявлена всього в двох описуваних продуктах – Internet Explorer і Mozilla Firefox. Подібні функції дозволяють припиняти передачу даних про відвідування користувачем сайтів в різні рекламні агентства і маркетингові відділи компаній, яка здійснюється за допомогою спеціальних скриптів, впроваджуваних у рекламні оголошення і просто в код веб сторінок. В Mozilla Firefox функція включається у розділі «Конфіденційність», настройок браузера, відповідна налаштування називається «Повідомляти веб-сайтів, що я не хочу, щоб стежили за мною». В цілому в таблиці виділяється Microsoft Internet Explorer, несильно від них відстають Mozilla Firefox і Google Chrome. Safari і Opera замикають список.

1.9 Висновок. Постановка задачі

На основі проаналізованих у першому розділі загроз на Web-сервера, які загрожують конфіденційності, доступності та цілісності інформації вибрати профіль захищенності. На основі проаналізованих вразливостей Інтернет-браузерів, на основі яких хакери будують атаки з застосуванням java-аплетів розробити необхідні засоби захисту. Розробка нових засобів захисту повинна підвищити рівень захисту Web-додатків від атак реалізованих за допомогою java-аплетів, виключаючи нині існуючі способи обходу.

РОЗДІЛ 2. СПЕЦІАЛЬНА ЧАСТИНА

2.1 Принцип роботи Java

В основі технології Java лежить клієнт-серверна модель, а Java-програма складається з декількох блоків, кожен з яких виконує певну частину загальної задачі. На стороні клієнта присутні тільки ті блоки, які необхідні в даний момент. Найбільш часто використовувані блоки зберігаються в кеші на жорсткому диску або в оперативній пам'яті комп'ютера користувача. Оскільки блок завантажується з сервера, то й керувати такою системою можна з сервера, тобто централізовано. Це також гарантує, що користувач завжди буде використовувати найостаннішу версію програми.

Основний компонент цієї технології – віртуальний Java-процесор, який являє собою середовище для виконання Java-команд, або так званих байт-кодів. Будь-яка Java-програма повинна відповідати специфікації віртуального Java-процесора, яка повністю визначає систему команд Java, типи даних, оброблених Java-процесором, і його регістри. Крім того, Java-процесор виконує деякі допоміжні функції, наприклад "збору сміття", тобто звільнення невживаної пам'яті.

Байт-коди розроблялися так, щоб максимально скоротити середню довжину команди. Java-процесор має мінімум регістрів, стековую архітектуру і часто використовує непряму адресацію. Тому більшість з команд займає всього один байт, до якого додається (якщо необхідно) номер операнда - 0, 1, 2, 3 і так далі. Крім того, для обробки кожного типу даних Java-процесор має свій набір команд. В результаті середня довжина Java-команди складає всього 1,8 байта (при довжині команди класичних RISC-процесорів в середньому чотири байти).

Крім віртуального процесора, технологія Java включає в себе (як необов'язкової елемента) об'єктно-орієнтована мова програмування, побудований на основі мови C++, з якого прибрали все зайве і додали нові механізми для забезпечення безпеки і розподілених обчислень. Однак мова Java можна замінити будь-яким іншим досить досконалим мовою програмування, додавши в нього всі

необхідні елементи. Наприклад, вже існує компілятор мови Пекла, який генерує програми байт-кодах Java.

Мова Java дає програмістам можливість не просто розробляти нові програми, але і використовувати елементи вже написаних і перевірених програм. Такий модульний принцип дозволяє швидко писати нові програмні продукти та ефективно модернізувати старі. Крім того, у стандарт мови входить безліч корисних бібліотек, на основі яких можна будувати обчислювальні системи будь-якої складності. Причому цей стандартний набір постійно поповнюється новими важливими функціями.

Ще однією особливістю Java є аплети. Аплет - це невелика програма, в якій повинно бути визначено декілька обов'язкових функцій. Аплет завантаження по мережі і може виконуватися на Web-браузері, який підтримує Java. Саме ця частина Java-технології призначена для використання у всесвітній мережі Internet, і тому захист повинна поширюватися як на сам аплет, так і на клієнта мережі, який використовує цей аплет.

2.1.1. Захист Java-технології

Найбільш уразливими з точки зору безпеки компонентом Java-технології є аплети, оскільки їх може використовувати будь-який клієнт, який зовсім не зобов'язаний знати правила "техніки безпеки при роботі з цими невеликими програмами. Саме тому аплетів для передбачені самі жорсткі методи захисту. Хоча різні браузери та програми перегляду аплетів можуть по-різному захищати інформацію користувача від нападу, але в загальному випадку аплету має бути заборонено наступне:

- читати, змінювати, видаляти і перейменовувати локальні файли;
- створювати локальні директорії і читати їх вміст;
- перевіряти існування і параметри певного файлу;
- здійснювати доступ по мережі до віддаленого комп ’ ютера;
- отримувати список мережевих сеансів зв'язку, які встановлює локальний комп'ютер з іншими комп'ютерами;

– відкривати нові вікна без повідомлення користувача (це необхідно для запобігання "емуляції" аплет інших програм);

– отримувати відомості про користувача або його домашньої директорії;

– визначати свої системні змінні;

– запускати локальні програми;

– виходити з інтерпретатора Java;

– завантажувати локальні бібліотеки;

– створювати потоки, які не перераховані в ThreadGroup (клас, керуючий виконанням потоків різних частин програми) цього аплету, і керувати ними;

– отримувати доступ до ThreadGroup іншого аплету;

– визначати свої об'єкти Class-Loader (Завантажувач Java-об'єктів) і SecurityManager (Диспетчер безпеки для аплетів);

– переобозначать системні об'єкти ContentHandlerFactory, SocketImplFactory і URLStreamHandler-Factory (ці класи управляють мережевий роботою Java);

– отримувати доступ до будь-якої упаковці, що відрізняється від стандартних;

– визначати класи, які входять до локальну упаковку. Ці правила забезпечують наступні компоненти Java-технології.

– Віртуальний Java-процесор, який постійно контролює свій стан.

– Завантажувач аплетів і Java-програм, який контролює завантажувані коди.

– Диспетчер безпеки (SecurityManager), контролює і блокуючий небезпечні дії аплетів.

У класі SecurityManager перелічені методи, які використовуються системою для контролю дій аплету в залежності від характеристик навколишнього середовища. Програма, яка застосовується для перегляду аплету, створює підклас SecurityManager, який і реалізує необхідну політику безпеки. Посилання на цей SecurityManager записується в об'єкті System.

Ще один механізм безпеки вбудований в завантажувач аплетів і програм (ClassLoader). Браузер перевизначають цей клас і реалізує свої власні правила роботи з мережевими протоколами. Одна з основних функцій завантажувача об'єктів - розділення простору імен різних аплетів і операційної системи, що дозволяє уникнути їх взаємного впливу.

Інша, не менш важлива функція завантажувача – верифікація байт-кодів, тобто перевірка правильності отриманого елемента Java-програми і його цілісності. У процесі верифікації з'ясовується наступне:

– чи відповідає версія отриманого блоку версіями інших елементів системи;

– збережений чи формат байт виконуваного-коду;

– чи відповідає програма специфікації конкретного віртуального Java-процесора;

– чи може виникнути переповнення або вичерпання стеку;

– чи всі регістри Java-процесора використовуються правильно;

– чи немає некоректних перетворень типів.

Метою такої перевірки є виявлення неправильного використання непрямої адресації, яке може призвести до порушення в роботі віртуального процесора, і перевірка цілісності аплету. Цей механізм забезпечує захист і надійну роботу розподіленої програми, що дозволяє не завантажувати в браузер всю Java-програму цілком, а довантажувати її невеликими блоками по мірі необхідності. Сам віртуальний Java-процесор також має вбудовані механізми захисту від нападу. Наприклад, оскільки байт-коди Java інтерпретуються, то можна контролювати індекси масивів, що дозволяє уникнути переповнення буфера - найпоширенішою і небезпечною помилки. Вбудовані прилади обробки виняткових ситуацій дозволяють ефективно вирішувати конфлікти, а "збирач сміття", який очищає невикористану пам'ять, не дає можливості "нападаючому" переглянути "відходи", які містять корисну інформацію.

2.2.2 Принцип роботи java-аплетів

Аплети використовуються для надання динамічного характеру Web-документа. Класичним прикладом використання java-аплетів є надання статичним картинкам певних ефектів (падаючого снігу, руху хвиль по поверхні води і т.п.), а також різні способи анімації динамічно задаються текстових написів. Інші мови компілюються в об'єктний код для конкретної операційної системи і процесора. Тому, наприклад, скомпилированное для Windows програма не може працювати на Macintosh. На відміну від них, Java не залежить від платформи, оскільки він створює проміжний код (bytecode - байт-код, який не залежить від конкретного процесора. Віртуальна машина Java (JVM - Java Virtual Machine) потім конвертує байт-код в машинний код, який розуміє процесор на даній конкретній системі.

Процес створення і виконання java-аплетів:

1 Програміст створює java-аплет і виконує його компіляцію.

2 Компілятор Java перетворює вихідний код у байт-індекс (не залежить від конкретного процесора).

3 Користувач завантажує java-аплет.

4 JVM конвертує байт-код в машинний код (для відповідного процесора, встановленого на комп'ютері користувача).

5 Аплет запускається при зверненні до нього.

Для запуску аплету, JVM створює віртуальну машину в рамках користувача середовища, звану пісочницею (sandbox). Ця віртуальна машина є замкнутою середовищем, в якій аплет виконує свої дії. Аплети зазвичай відправляються по запитам від веб сторінок, тому аплет виконується відразу, як тільки він приходить. Такий аплет може виконувати шкідливу діяльність навмисно або випадково, якщо розробник аплету зробив щось неправильно. Тому пісочниця строго обмежує доступ аплету до будь-яких системних ресурсів. JVM є посередником між аплетом і ресурсами системи, перехоплюючи, перевіряючи і виконуючи запити аплету до системних ресурсів і залишаючи при цьому сам аплет всередині пісочниці. Компоненти цього процесу показані на рисунку 2.1.

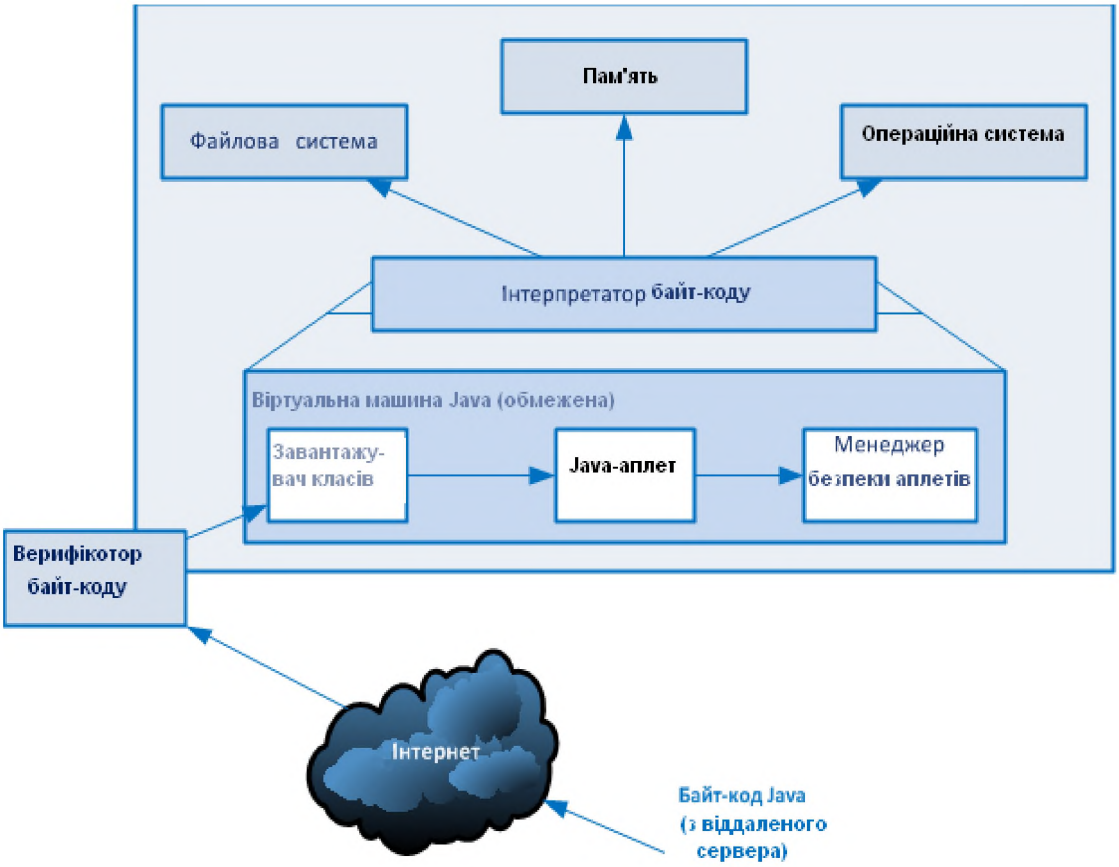


Рисунок 2.1 – Модель безпеки Java

Мова програмування Java має ряд особливостей, які спрощують його використання в Інтернет. До їх числа слід віднести наступні два:

1 Вихідний текст програми перетворюється не в машинні команди, а в спеціальний код, який не може безпосередньо здійсняться процесором. Це забезпечує апаратну незалежність програми і дозволяє використовувати її на комп'ютерах різних типів. Однак такий підхід вимагає наявності на комп'ютері, де виконується-Java аплет, спеціального модуля (так званої "Java-машини"), що забезпечує перетворення незалежного коду в машинні команди і їх виконання процесором.

2 У мовою відсутні засоби, що дозволяють організувати пряме взаємодія з пристроями і, перш за все, з оперативною пам'яттю. Це виключає можливість появи в тексті програм великої кількості помилок, які можуть призвести до збою комп'ютерної системи, а, отже, забезпечує підвищену стабільність роботи програм.

Java-аплети реалізуються у вигляді окремих файлів і зберігаються на Web-серверах. Можливість використання аплетів в Web-документах забезпечується тим, що: на клієнтському комп'ютері є програмний компонент "Java-машина", що забезпечує виконання аплету (Java-машина може включатися до складу Web-клієнта, або до складу операційної системи, яка керує роботою Web-клієнта); до складу мови HTML включений спеціальний тег, що дозволяє підключити аплет до Web-документа і вказати його адресу в Мережі і вхідні параметри. Аплети хоста можуть використовувати різні способи, не призначені для цього системного ресурсу, чи можуть змусити користувача виконати різні небажані дії. Приклади небажаних дій аплетів на хості включають DoS-атаки, підробку e-mail адрес, вторгнення в приватне життя (наприклад, експорт ідентифікації e-mail адреси та інформації про платформі) та інсталювання люків (backdoors) в систему. Так як модель безпеки Java є складною, користувачеві може бути важко зрозуміти її і керувати нею. Така ситуація може збільшити ризик.

2.2 Вибір профілю захищеності WEB-додатків

Згідно з рекомендаціями НД ТЗІ 2.5-010-03 та з урахуванням особливостей надання доступу до інформації WEB-додатків, типових характеристик середовища функціонування та особливостей технологічних процесів оброблення інформації, а також зважаючи на те, що в АС класу «3» WEB-сервер розміщується у оператора, а робочі станції – на іншій території, взаємодія яких з WEB-сервером здійснюється з використанням мереж передачі даних (технологія Т2), був обраний наступний профіль захищеності:

{ КА-2, КВ-1, ЦА-1, ЦО-1, ЦВ-1, ДВ-1, ДР-1, НР-2, НИ-2, НК-1, НО-1,НЦ-1, НТ-1, НВ-1 }

КА-2. Базова адміністративна конфіденційність;

КВ-1. Мінімальна конфіденційність при обміні;

ЦА-1. Мінімальна адміністративна цілісність;

ЦО-1. Обмежений відкат;

ЦВ-1: Мінімальна цілісність при обміні;

ДВ-1. Ручне відновлення;

ДР-1. Квоти;

НР-2. Захищений журнал;

НИ-2. Одиночна ідентифікація і автентифікація;

НК-1. Однонаправлений достовірний канал;

НО-1. Виділення адміністратора;

НЦ-1. КЗЗ з контролем цілісності;

НТ-1. Самотестування за запитом;

НВ-1: Автентифікація вузла.

2.2.1 Критерії конфіденційності

В будь-якій КС інформація може переміщуватись в одному з двох напрямів: від користувача до об'єкта або від об'єкта до користувача. Конфіденційність забезпечується через додержання вимог політики безпеки щодо переміщення інформації від об'єкта до користувача або процесу. Правильне (допустиме) переміщення визначається як переміщення інформації до авторизованого користувача, можливо, через авторизований процес.

КЗЗ оцінюваної КС надає послуги з захисту об'єктів від несанкціонованого ознайомлення з їх змістом (компрометації). Конфіденційність забезпечується такими послугами: довірча конфіденційність, адміністративна конфіденційність, повторне використання об'єктів, аналіз прихованих каналів, конфіденційність при обміні.

КА-2. Базова адміністративна конфіденційність

Ця послуга дозволяє адміністратору або спеціально авторизованому користувачу керувати потоками інформації від захищених об'єктів до користувачів. Рівні даної послуги ранжируються на підставі повноти захисту і вибірковості управління.

Політика адміністративної конфіденційності, що реалізується КЗЗ, визначає множину об'єктів КС, до яких вона відноситься. КЗЗ здійснює розмежування доступу на підставі атрибутів доступу користувача і захищеного об'єкта. Запити

на зміну прав доступу обробляються КЗЗ тільки в тому випадку, якщо вони надходять від адміністраторів або від користувачів, яким надані відповідні повноваження. КЗЗ надає можливість адміністратору або користувачу, що має відповідні повноваження, для кожного захищеного об'єкта шляхом керування належністю користувачів, процесів і об'єктів до відповідних доменів визначити конкретних користувачів і/або групи користувачів, які мають право одержувати інформацію від об'єкта. КЗЗ надавє можливість адміністратору або користувачу, що має відповідні повноваження, для кожного процесу через керування належністю користувачів і процесів до відповідних доменів визначити конкретних користувачів і/або групи користувачів, які мають право ініціювати процес.

Права доступу до кожного захищеного об'єкта встановлюються в момент його створення або ініціалізації.

КВ-1. Мінімальна конфіденційність при обміні

Ця послуга дозволяє забезпечити захист об'єктів від несанкціонованого ознайомлення з інформацією, що міститься в них, під час їх експорту/імпорту через незахищене середовище. Рівні даної послуги ранжируються на підставі повноти захисту і вибірковості керування.

Політика конфіденційності при обміні, що реалізується КЗЗ, визначає множину об'єктів і інтерфейсних процесів, до яких вона відноситься. Політика конфіденційності при обміні, що реалізується КЗЗ, визначає рівень захищеності, який забезпечується механізмами, що використовуються, і спроможність користувачів і/або процесів керувати рівнем захищеності. КЗЗ забезпечує захист від безпосереднього ознайомлення з інформацією, що міститься в об'єкті, який передається.

Ця послуга забезпечується використанням протоколу HTTPS, що є модифікацією базового протоку передачі даних HTTP, який використовує криптографічні протоколи TSL и SSL та TCP-порт 443.

2.2.2 Критерії цілісності

КЗЗ оцінюваної КС надає послуги з захисту оброблюваної інформації від несанкціонованої модифікації. Цілісність забезпечується такими послугами: довірча цілісність, адміністративна цілісність, відкат, цілісність при обміні.

ЦА-1. Мінімальна адміністративна цілісність

Ця послуга дозволяє адміністратору або спеціально авторизованому користувачу керувати потоками інформації від користувачів до захищених об'єктів Web-додатків. Рівні даної послуги ранжируються на підставі повноти захисту і вибірковості керування.

Політика адміністративної цілісності, що реалізується КЗЗ, визначає множину об'єктів КС, до яких вона відноситься. КЗЗ здійснює розмежування доступу на підставі атрибутів доступу користувача і захищеного об'єкта. Запити на зміну прав доступу повинні оброблятися КЗЗ тільки в тому випадку, якщо вони надходять від адміністраторів або від користувачів, яким надані відповідні повноваження. КЗЗ надає можливість адміністратору або користувачу, який має відповідні повноваження, для кожного захищеного об'єкта шляхом керування належністю користувачів, процесів і об'єктів до відповідних доменів визначити конкретних користувачів і/або групи користувачів, які мають право модифікувати об'єкт.

Права доступу до кожного захищеного об'єкта встановлюватися в момент його створення або ініціалізації. Представлені правила збереження атрибутів доступу об'єктів під час їх експорту і імпорту

ЦО-1. Обмежений відкат

Відкат є багатосторонньою послугою, що дозволяє відновлюватися після помилок користувача, збоїв програмного забезпечення або апаратури і підтримувати цілісність баз даних, додатків, побудованих на транзакціях. Дана послуга забезпечує можливість відмінити операцію або послідовність операцій і повернути захищений об’єкт до попереднього стану. Рівні даної послуги ранжируються на підставі множини операцій, для яких забезпечується відкат.

Використання відкладеного резервування, що вимагає втручання користувача для завантаження резервного носія, не є реалізацією відкату. Якщо система реалізує дану послугу, то її використання має фіксуватись в журналі. Відміна операції не повинна приводити до видалення з журналу запису про операцію, яка пізніше була відмінена.

Ця послуга забезпечується завдяки використанню резервних копій даних.

ЦВ-1: Мінімальна цілісність при обміні

Ця послуга дозволяє забезпечити захист Web-додатків від несанкціонованої модифікації інформації, що міститься в них, під час їх експорту/імпорту через незахищене середовище. Рівні даної послуги ранжируються на підставі повноти захисту і вибірковості керування.

Політика цілісності при обміні, що реалізується КЗЗ, визначає множину об'єктів КС і інтерфейсних процесів, до яких вона відноситься, рівень захищеності, що забезпечується використовуваними механізмами, і спроможність користувачів і/або процесів керувати рівнем захищеності. КЗЗ забезпечує можливість виявлення порушення цілісності інформації, що міститься в об'єкті, який передається.

Ця послуга забезпечується використанням протоколу HTTPS, що є модифікацією базового протоку передачі даних HTTP, який використовує криптографічні протоколи TSL и SSL та TCP-порт 443.

2.2.3 Критерії доступності

КЗЗ оцінюваної КС надаває послуги щодо забезпечення можливості використання КС в цілому, окремих функцій або оброблюваної інформації на певному проміжку часу і гарантувати спроможність КС функціонувати у випадку відмови її компонентів. Доступність може забезпечуватися в КС такими послугами: використання ресурсів, стійкість до відмов, гаряча заміна, відновлення після збоїв.

ДВ-1. Ручне відновлення

Ця послуга забезпечує повернення КС у відомий захищений стан після відмови або переривання обслуговування. Рівні даної послуги ранжируються на підставі міри автоматизації процесу відновлення.

Політика відновлення, що реалізується КЗЗ, визначає множину типів відмов КС і переривань обслуговування, після яких можливе повернення у відомий захищений стан без порушення політики безпеки. Чітко вказані рівні відмов, у разі перевищення яких необхідна повторна інсталяція КС. Після відмови КС або переривання обслуговування КЗЗ повинен перевести КС до стану, із якого повернути її до нормального функціонування може тільки адміністратор або користувачі, яким надані відповідні повноваження. Існувати ручні процедури, за допомогою яких можна безпечним чином повернути КС до нормального функціонування.

Ця послуга реалізується за допомогою включення до АС додаткових резервних WEB-додатків, що є копіями основних WEB- додатків, за допомогою яких можна відновити нормальну роботу системи.

ДР-1. Квоти

Ця послуга дозволяє користувачам керувати використанням послуг і ресурсів. Рівні даної послуги ранжируються на підставі повноти захисту і вибірковості керування доступністю послуг КС.

Політика використання ресурсів, що реалізується КЗЗ, визначає множину об'єктів КС, до яких вона відноситься. Політика використання ресурсів визначає обмеження, які можна накладати, на кількість даних об'єктів (обсяг ресурсів), що виділяються окремому користувачу. Запити на зміну встановлених обмежень обробляються КЗЗ тільки в тому випадку, якщо вони надходять від адміністраторів або від користувачів, яким надані відповідні повноваження.

2.2.4 Критерії спостереженості

КЗЗ оцінюваної КС надає послуги з забезпечення відповідальності користувача за свої дії і з підтримки спроможності КЗЗ виконувати свої функції. Спостереженість забезпечується в КС такими послугами: реєстрація (аудит),

ідентифікація і автентифікація, достовірний канал, розподіл обов'язків, цілісність КЗЗ, самотестування, ідентифікація і автентифікація при обміні, автентифікація відправника, автентифікація отримувача.

НР-2. Захищений журнал

Реєстрація дозволяє контролювати небезпечні для КС дії. Рівні даної послуги ранжируються залежно від повноти і вибірковості контролю, складності засобів аналізу даних журналів реєстрації і спроможності вияву потенційних порушень.

Політика реєстрації, що реалізується КЗЗ, визначає перелік подій, що реєструються. КЗЗ здійснює реєстрацію подій, що мають безпосереднє відношення до безпеки. Журнал реєстрації містить інформацію про дату, час, місце, тип і успішність чи неуспішність кожної зареєстрованої події. Журнал реєстрації містить інформацію, достатню для встановлення користувача, процесу і/або об'єкта, що мали відношення до кожної зареєстрованої події. КЗЗ забезпечує захист журналу реєстрації від несанкціонованого доступу, модифікації або руйнування.

Адміністратори і користувачі, яким надані відповідні повноваження, мають в своєму розпорядженні засоби перегляду і аналізу журналу реєстрації

НИ-2. Одиночна ідентифікація і аутентифікація

Ідентифікація і автентифікація дозволяють КЗЗ визначити і перевірити особистість користувача, що намагається одержати доступ до КС. Рівні даної послуги ранжируються залежно від числа задіяних механізмів автентифікації.

Політика ідентифікації і автентифікації, що реалізується КЗЗ, визначає атрибути, якими характеризується користувач, і послуги, для використання яких необхідні ці атрибути. Кожний користувач повинен однозначно ідентифікуватися КЗЗ . Перш ніж дозволити будь-якому користувачу виконувати будь-які інші, контрольовані КЗЗ дії, КЗЗ автентифікує цього користувача з використанням захищеного механізму. КЗЗ забезпечує захист даних автентифікації від несанкціонованого доступу, модифікації або руйнування.

НК-1. Однонаправлений достовірний канал

Ця послуга дозволяє гарантувати користувачу можливість безпосередньої взаємодії з КЗЗ. Рівні даної послуги ранжируються залежно від гнучкості надання можливості КЗЗ або користувачу ініціювати захищений обмін.

Політика достовірного каналу, що реалізується КЗЗ, визначає механізми встановлення достовірного зв'язку між користувачем і КЗЗ. Достовірний канал використовується для початкової ідентифікації і автентифікації. Зв'язок з використанням даного каналу ініціюється виключно користувачем.

НО-1. Виділення адміністратора

Ця послуга дозволяє зменшити потенційні збитки від навмисних або помилкових дій користувача і обмежити авторитарність керування. Рівні даної послуги ранжируються на підставі вибірковості керування можливостями користувачів і адміністраторів.

Політика розподілу обов'язків, що реалізується КЗЗ, визначає ролі адміністратора і звичайного користувача і притаманні їм функції. Користувач повинен мати можливість виступати в певній ролі тільки після того, як він виконає певні дії, що підтверджують прийняття їм цієї ролі.

НЦ-1. КЗЗ з контролем цілісності

Ця послуга визначає міру здатності КЗЗ захищати себе і гарантувати свою спроможність керувати захищеними об'єктами.

Політика цілісності КЗЗ визначає склад КЗЗ і механізми контролю цілісності компонентів, що входять до складу КЗЗ. В разі виявлення порушення цілісності будь-якого із своїх компонентів КЗЗ необхідно повідомити адміністратора і/або автоматично відновити відповідність компонента еталону або перевести КС до стану, з якого повернути її до нормального функціонування може тільки адміністратор або користувачі, яким надані відповідні повноваження. Описані обмеження, дотримання яких дозволяє гарантувати, що послуги безпеки доступні тільки через інтерфейс КЗЗ і всі запити на доступ до захищених об'єктів контролюються КЗЗ.

НТ-1. Самотестування за запитом

Самотестування дозволяє КЗЗ перевірити і на підставі цього гарантувати правильність функціонування і цілісність певної множини функцій КС. Рівні даної послуги ранжируються на підставі можливості виконання тестів у процесі запуску або штатної роботи.

Політика самотестування, що реалізується КЗЗ, описує властивості КС і реалізовані процедури, які можуть бути використані для оцінки правильності функціонування КЗЗ. КЗЗ здатен виконувати набір тестів з метою оцінки правильності функціонування своїх критичних функцій. Тести виконуються за запитом користувача, що має відповідні повноваження.

Ця послуга реалізується завдяки ПЗ Nmap, яке має у своїй базі даних велику кількість тестових запитів для тестування WEB-додатків через користувача.

НВ-1: Автентифікація вузла

Ця послуга дозволяє одному КЗЗ ідентифікувати інший КЗЗ (встановити і перевірити його ідентичність) і забезпечити іншому КЗЗ можливість ідентифікувати перший, перш ніж почати взаємодію. Рівні даної послуги ранжируються на підставі повноти реалізації.

Політика ідентифікації і автентифікація при обміні, що реалізується КЗЗ, визначає множину атрибутів КЗЗ і процедури, які необхідні для взаємної ідентифікації при ініціалізації обміну даними з іншим КЗЗ. КЗЗ, перш ніж почати обмін даними з іншим КЗЗ, ідентифікував і автентифікує цей КЗЗ з використанням захищеного механізму

Підтвердження ідентичності виконується на підставі затвердженого протоколу аутентифікації.

2.3 Аналіз загроз WEB-додатків

По відношенню до інформації, що знаходиться на WEB-додатках, та до користувачів WEB-сторінок існує ряд загроз відображених в таблиці 2.1.

Таблиця 2.1 – Загрози безпеки інформації WEB-додатків

*[1)]	Тип загрози		Можливі наслідки
К	Аналіз мережевого трафіку		Дослідження характеристик мережевого трафіку, перехоплення даних, що передаються, у тому числі ідентифікаторів користувачів і паролів
К	Загроза виявлення пароля		Виконання будь-якої дії, пов'язаного з отриманням несанкціонованого доступу
Д	Відмова в обслуговуванні	Часткове вичерпання ресурсів	Зниження пропускної здатності каналів зв'язку, продуктивності мережевих пристроїв. Зниження продуктивності серверних додатків
		Повне вичерпання ресурсів	Неможливість передачі повідомлень через відсутність доступу до середовища передачі, відмова у встановленні з'єднання. Відмова у наданні сервісу.
		Порушення логічної пов'язаності між атрибутами, даними, об'єктами	Неможливість передачі повідомлень через відсутність коректних маршрутно-адресних даних. Неможливість отримання послуг через несанкціоновану модифікацію ідентифікаторів, паролів
		Використання помилок у програмах	Порушення працездатності мережевих пристроїв
К, Ц, Д	Віддалений запуск додатків	Шляхом розсилки файлів, що містять деструктивний виконуваний код вірусне зараження	Порушення конфіденційності, цілісності, доступності інформації
		Шляхом переповнення буфера серверного додатку	
		Шляхом використання можливостей віддаленого управління системою, що надаються прихованими програмними і апаратними закладками або використовуваними штатними засобами	Приховане управління системою

*[1)] *Властивість інформації, яка порушується*

2.3.1 Загроза "Аналіз мережевого трафіку"

Ця загроза реалізується за допомогою спеціальної програми аналізатора пакетів (sniffer), перехоплює всі пакети, що передаються по сегменту мережі, і виділяє серед них ті, в яких передаються ідентифікатор користувача і пароль. У ході реалізації загрози порушник:

– вивчає логіку роботи ІС – тобто прагне отримати однозначну відповідність подій, що відбуваються в системі, і команд, які пересилаються при цьому хостами, у момент появи даних подій. Надалі це дозволить встановити зловмисникові на основі завданих відповідних команд отримати, наприклад, розширені права на дії в системі або розширити свої повноваження в ній;

– перехоплює потік даних, якими обмінюються компоненти мережевої операційної системи, для отримання конфіденційної або ідентифікаційної інформації (наприклад, статичних паролів користувачів для доступу до віддалених хостів по протоколам FTP і TELNET, які не передбачають шифрування) її підміни, модифікації і т.п.

2.3.2 Загроза виявлення паролів

Мета реалізації загрози полягає в отриманні НСД шляхом подолання парольного захисту. Зловмисник може реалізувати загрозу з допомогою цілого ряду методів, таких як простий перебір, перебір з використанням спеціальних словників, установкою шкідливої програми для паролю, підміною довіреного об'єкта мережі (IP-spoofing) і перехоплення пакетів (sniffing). В основному для реалізації загрози використовуються спеціальні програми для паролю, які намагаються отримати доступ до комп'ютера шляхом послідовного підбору паролів. У разі успіху, зловмисник може створити для себе "прохід" для майбутнього доступу, який буде діяти, навіть якщо на хості змінити пароль доступу.

2.3.3 Загрози типу "Відмова в обслуговуванні"

Ці загрози засновані на недоліки мережевого програмного забезпечення, його вразливості, що дозволяють порушникові створювати умови, коли операційна система виявляється не в змозі обробляти вхідні пакети.

Можуть бути виділені кілька різновидів таких загроз:

– прихована відмова в обслуговуванні, викликана залученням частини ресурсів ІС на обробку пакетів, що передаються зловмисником зі зниженням пропускної здатності каналів зв'язку, продуктивності мережевих пристроїв, порушенням вимог до часу обробки запитів. Прикладами реалізації загроз подібного роду може служити: спрямований шторм эхо-запитів по протоколу ОСМР (Pingflooding), шторм запитів на встановлення TCP-сполук (SYN-flooding), шторм запитів до FTP-серверу;

– явна відмова в обслуговуванні, викликана вичерпанням ресурсів ІС при обробці пакетів, що передаються зловмисником (заняття всієї пропускної смуги каналів зв'язку, переповнення черг запитів на обслуговування), при якому легальні запити не можуть бути передані через мережу з-за недоступності середовища передачі, або отримують відмову в обслуговуванні через переповнення черг запитів, дискового простору пам'яті і т.д. Прикладами загроз даного типу можуть служити шторм широкомовних ICPM-эхо-запитів (Smurf), спрямований шторм (SYN-flooding), шторм повідомлень поштового сервера (Spam);

– явна відмова в обслуговуванні, викликана порушенням логічної пов'язаності між технічними засобами ІС при передачі порушником керуючих повідомлень від імені мережевих пристроїв, що призводять до зміни маршрутно-адресних даних (наприклад, ICMPRedirectHost, DNS-flooding) або ідентифікаційної і аутентификационной інформації;

– явна відмова в обслуговуванні, викликана передачею зловмисником пакетів з нестандартними атрибутами (загрози типу "Land", "TearDrop", "Bonk", "Nuke", "UDP-bomb") або, які мають довжину, що перевищує максимально допустимий розмір (загроза типу "PingDeath" (, що може призвести до збою мережевих пристроїв, які беруть участь в обробці запитів, за умови наявності

помилок в програмах, які реалізують протоколи мережевого обміну. Результатом реалізації цієї загрози може стати порушення працездатності відповідної служби надання спільного доступу до даних в ІС, передача з однієї адреси точної кількості запитів на підключення технічних засобів у складі ІС, яке максимально може вмістити трафік (спрямований "шторм запитів"), що тягне за собою переповнення черги запитів і відмову однієї з мережевих служб або повну зупинку ІС з-за неможливості системи займатися нічим, крім обробки запитів.

2.3.4 Загрози віддаленого запуску додатків

Загроза полягає в прагненні запустити на хості ІС різні попередньо впроваджені шкідливі програми: програми-закладки, віруси, "мережеві шпигуни", основна мета яких – порушення конфіденційності, цілісності, доступності інформації і повний контроль за роботою машини. Крім того, можливий несанкціонований запуск прикладних програм користувачів для несанкціонованого отримання необхідних порушникові даних, для запуску керованих прикладною програмою прикладних процесів та ін.

Виділяються три підкласу даних загроз:

– розповсюдження файлів, що містять несанкціонований виконуваний код;

– віддалений запуск програми шляхом переповнення буфера додатків-серверів;

– віддалений запуск програми шляхом використання можливостей віддаленого управління системою, що надаються прихованими програмними і апаратними закладками, або використовуваними штатними засобами.

Типові загрози першого з цих підкласів ґрунтуються на активізації поширюваних файлів при випадковому зверненні до них. Прикладами таких файлів можуть служити: файли, що містять виконуваний код у вигляді макрокоманд (документи MicrosoftWord, Excel і т.п.); виконувані файли, що містять коди програм. Для розповсюдження файлів можуть використовуватися служби електронної пошти, передачі файлів, мережний файлової системи.

При загрозі другого підкласу використовуються недоліки програм, які реалізують мережеві сервіси (зокрема, відсутність контролю переповнення буфера). Налаштуванням системних регістрів іноді вдається перемикати процесор після переривання, викликаного переповненням буфера, на виконання коду, який перебуває за кордоном буфера.

При загрозі третього підкласу порушник використовує можливості віддаленого управління системою, що надаються прихованими компонентами (наприклад, "троянськими програмами" типу BackOrifice, NetBus), або штатними засобами управління і адмініструванням комп'ютерних мереж (LandeskManagementSuite, Managewise, BackOrifice і т.п.). В результаті їх використання вдається домогтися віддаленого контролю над станціями мережі.

2.4 Розробка засобів захисту WEB-додатків від атак реалізованих за допомогою java-аплетів

Розроблені засобів захисту WEB-додатків відноситься до засобу протидій від атак реалізованих за допомогою java-аплетів. Засоби захисту запропоновані для браузера Mozilla Firefox, яка згідно статистики займає приблизно одну четверту частку світового ринку браузерів.

2.4.1 Опис запропонованих засобів захисту

Для більшої безпеки при роботі з Інтернет – браузером були запропоновані наступні проектні рішення:

1 Миттєва ідентифікація Web-сайту;
2 Політика безпеки вмісту браузера;
3 Безпечні оновлення;
4 Захист від шкідливих сайтів;
5 Приватний перегляд;
6 Захист від стеження;
7 Інтеграція з антивірусом

2.4.1.1 Миттєва ідентифікація Web-сайту

Для переконання в достовірності сайта, перш ніж зробити покупку, перевести кошти, забронювати готель та інших дії необхідно нажати по значку сайту для його миттєвої ідентифікації. Нажати ще раз, щоб отримати більше інформації: скільки разів ви відвідали цей сайт, збережені чи ваші паролі. Перевіряйте підозрілі сайти і переконуйтеся в тому, що сайт є саме тим, за що він себе видає. Приклад миттєвої ідентифікації наведений на рисунку 2.2.

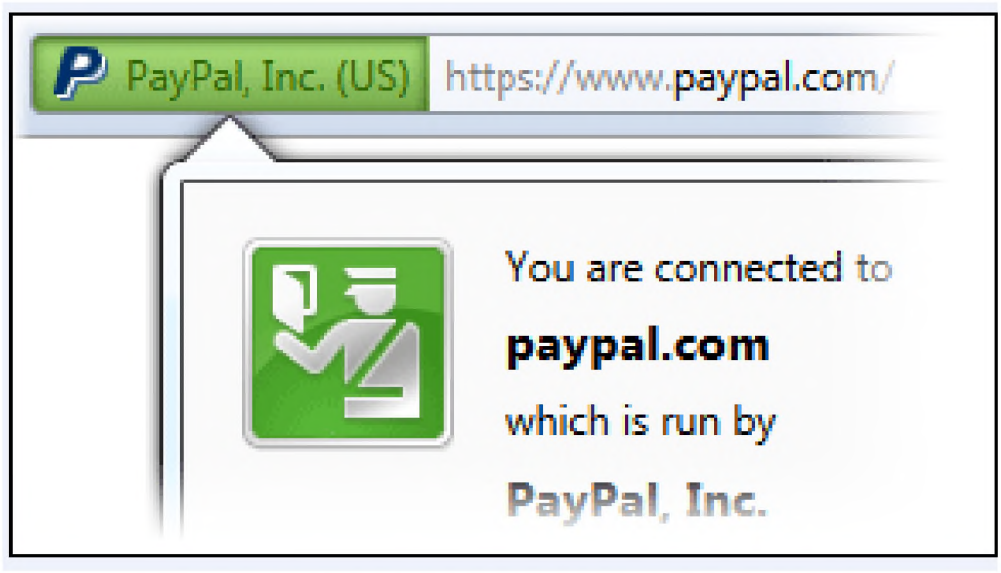


Рисунок 2.2 – Рівень безпеки веб-ресурсів

2.4.1.2 Політика безпеки вмісту браузера

Політика безпеки вмісту браузера розроблена для запобігання атак, здійснюваних через міжсайтового скриптінг, шляхом надання сайтів механізму, що дозволяє явно повідомити браузеру, яке вміст є легітимним. Браузер може ігнорувати будь-який не схвалене сайтом вміст, що підвищує вашу безпеку.

Для запуску JavaScript, Java і активного вмісту тільки для довірених доменів за вибором користувача, наприклад, особистий банківський веб-сайт користувача запропоновано використати доповнення NoScript. Воно забезпечує безпеку роботи в "зоні довіри", захищає від атак з використанням міжсайтових сценаріїв (XSS), крос-зони DNS підміною / CSRF-хакерських атак (маршрутизаторів), і атаки ClickJacking, завдяки своїй унікальній технології ClearClick.

Такий запобіжний підхід запобігає використання вразливостей без втрати функціональності[4]. У NoScript є можливість дозволити виконання скриптів тільки на довірених сайтах (білий список), а на всіх інших заборонити або дозволити одноразово. Створення білого списку наведено на рисунку 2.3.

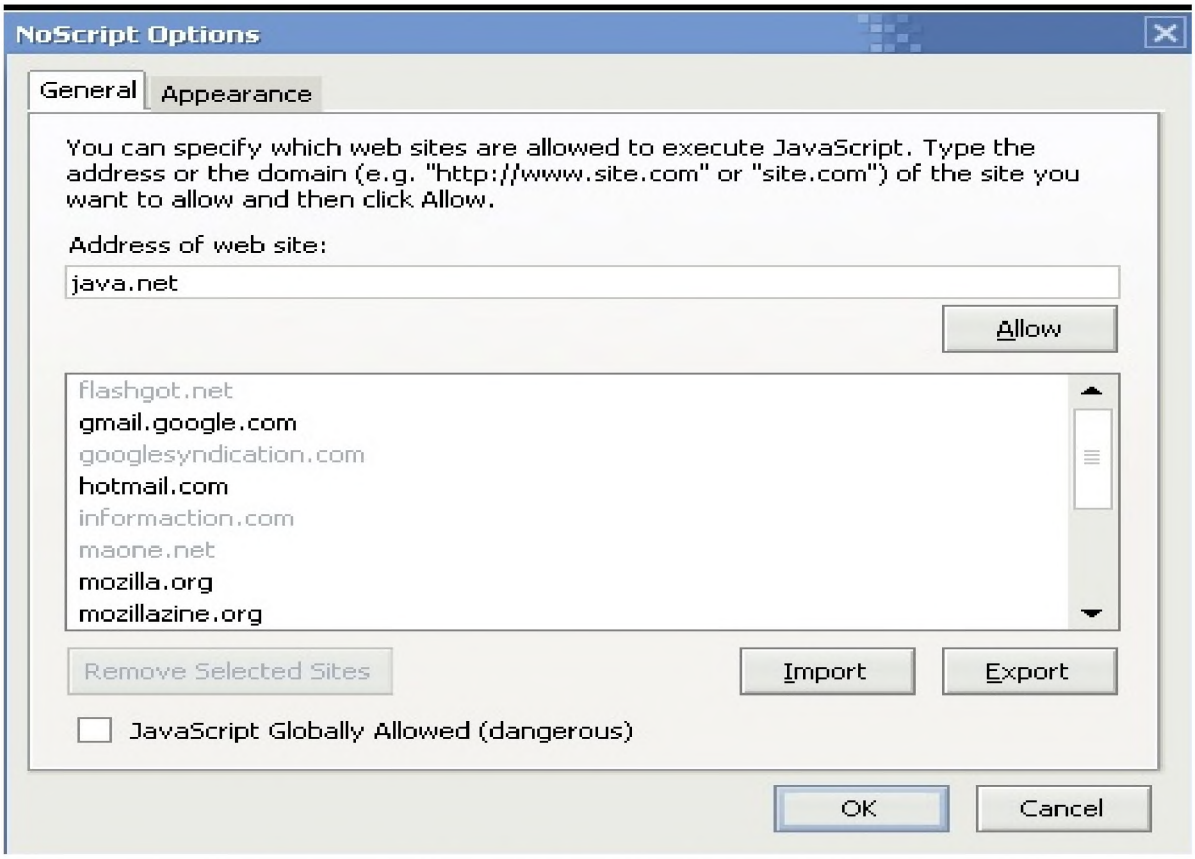


Рисунок 2.3 – Створення «білого списку»

2.4.1.3 Безпечні оновлення

Використовувати захищене з'єднання перед установкою або оновленням доповнень, стороннього програмного забезпечення і антивірусів.

2.4.1.4 Захист від шкідливих сайтів

Для захисту Firefox від вірусів, троянів і шпигунського програмного забезпечення було запропоновано використання доповнення WOT. Якщо ви випадково потрапили на сайт, атакуючий користувачів, то WOT попередить вас, що вам не варто відвідувати цей сайт і повідомить, чому цей сайт є безпечним.

Як світлофор, WOT показує, яким сайтам можна довіряти при пошуку і покупці в Інтернеті. Колір значка WOT показує рівень безпеки веб-ресурсу: «червоний» відвідувати сайти не рекомендується, перед відвідуванням «жовтих» сайтів слід добре подумати, а на «зелені» можна сміливо заходити. Рівень безпеки веб-ресурсів наведений на рисунку 2.4.

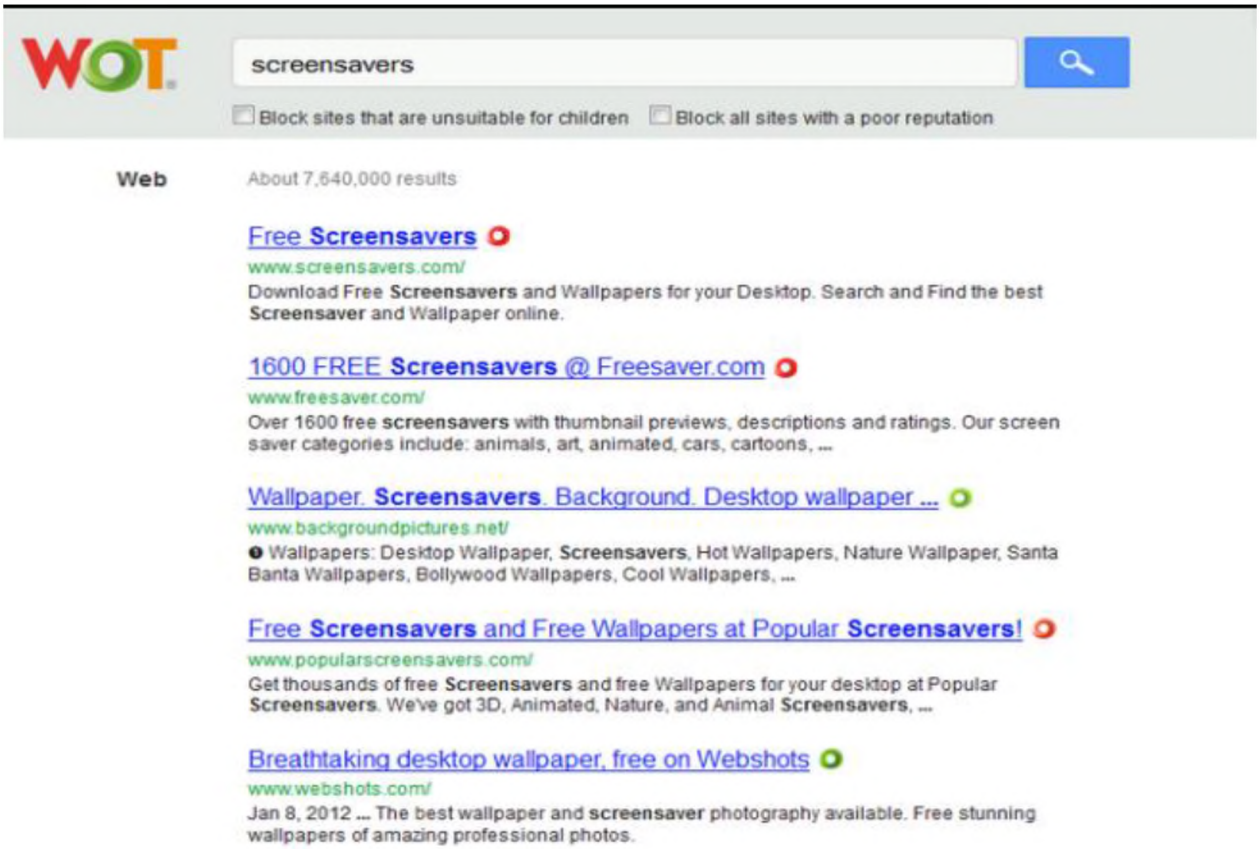


Рисунок 2.4 – Рівень безпеки веб-ресурсів

WOT користується підтримкою багатомільйонного спільноти користувачів, які надають інформацію про мільйони сайтів, ґрунтуючись на власному досвіді.

Вони виставляють рейтинг сайтів, оцінюючи їх за чотирма критеріями: чи можна довіряти; надійність продавця; конфіденційність; безпека для дітей.

Як приклад, рейтинг сайту www.mozilla.org наведено на рисунку 2.5.

Можливо побачити зелений, жовтий і червоний значки поруч із результатами пошуку в системі Google, біля посилань сайтів соціальних мереж Facebook і Twitter, у поштових повідомленнях Gmail, в Wikipedia, а також на багатьох інших популярних сайтах.

Безпечний пошук WOT дасть вам безпечні відповіді. Можливо фільтрувати посилання з поганим рейтингом, контролюючи відвідування сайтів вами і членами вашої родини.

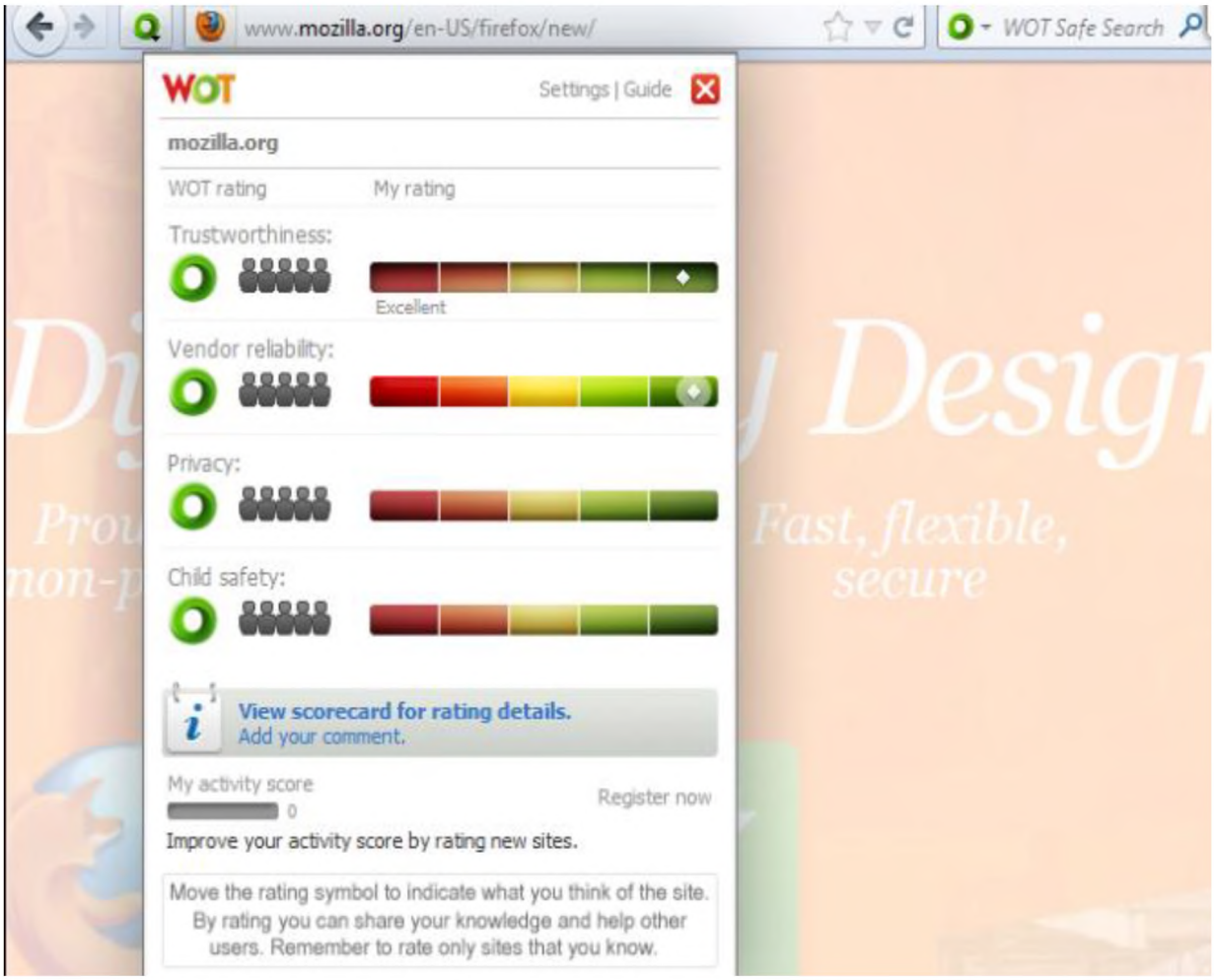


Рисунок 2.5 – Рейтинг сайту mozilla.org

2.4.1.5 Приватний перегляд

При роботі з Інтернетом, браузер запам'ятовує безліч інформації: сайти, які ви відвідували, файли, які ви завантажили, і багато іншого. Однак, інколи, необхідно, щоб інші користувачі цього комп'ютера не побачили цю інформацію. Наприклад, якщо ви працюєте зі своїм банком на загальному комп'ютері або перевіряєте пошту в Інтернет-кафе. Цього можна досягнути використовуючи режим приватного перегляду.

Режим приватного перегляду дозволяє відвідувати веб-сайти без збереження інформації про те, які сторінки і сайти ви відвідали.

В режимі приватного перегляду не зберігаються (наведений на рисунку 2.6):

– Відвідані веб-сайти: Сторінки не будуть додаватися до списку історії веб-сайтів в Журналі історії і в списку автозаповнення в адресному рядку.

– Дані з форм та пошуку: Нічого з того, що ви вводили в різні форми на веб-сторінках або в пошукову панель не буде записано в історію.

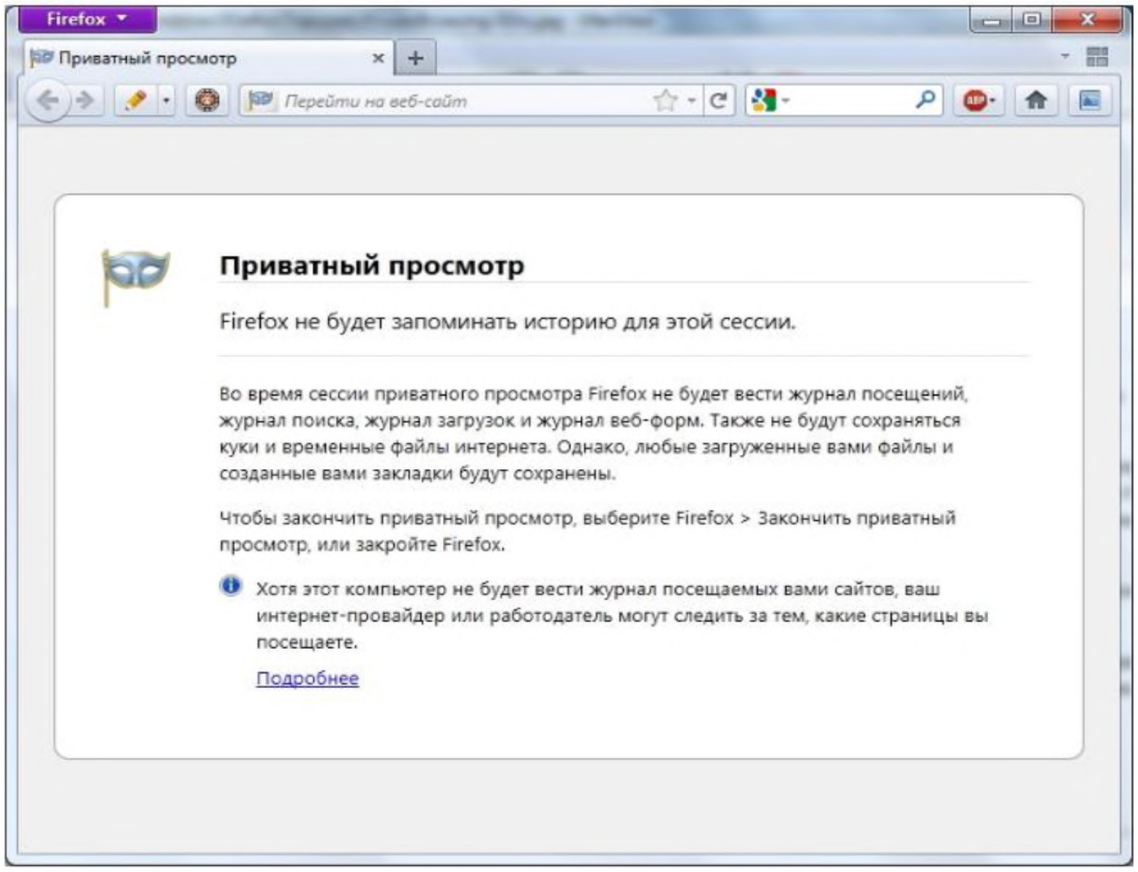


Рисунок 2.6 – Режим приватного перегляду

– Паролі: Нові паролі зберігатися не будуть.

– Список завантажень: Після вимкнення Режиму приватного перегляду завантажені під час Режиму файли не будуть відображатися у вікні завантажень.

– Cookie: Cookie зберігають інформацію про відвідані веб-сайти, наприклад, інформацію про налаштування сайту або статус реєстрації. До cookie відноситься інформація і налаштування сайту, збережена такими засобами, як Adobe Flash. Cookie також можуть бути використані третьою стороною для відстеження користувача по сайтам.

– Кеш веб-файлів: Тимчасові файли інтернету або кешовані файли веб-сторінок не будуть збережуться на комп'ютері, поки включений Режим приватного перегляду.

2.4.1.6 Захист від стеження

Більшість великих Web-сайтів відстежують поведінку своїх відвідувачів, а потім продають або надають цю інформацію іншим компаніям (наприклад рекламодавцям).

Відстеження – це поняття, яке включає в себе безліч різних способів, за допомогою яких сайти, рекламодавці та інші дізнаються про поведінку користувачів при перегляді сайтів. Це включає в себе інформацію про те, які сайти відвідує користувач, що йому подобається, не подобається і що він купуєте. Ця інформація використовується для того, щоб показати вам ті продукти, сервіси і рекламу, які швидше за все зацікавлять саме вас. Для запобігання відстеження рекомендується застосовувати опцію Не-стеж-за мною.

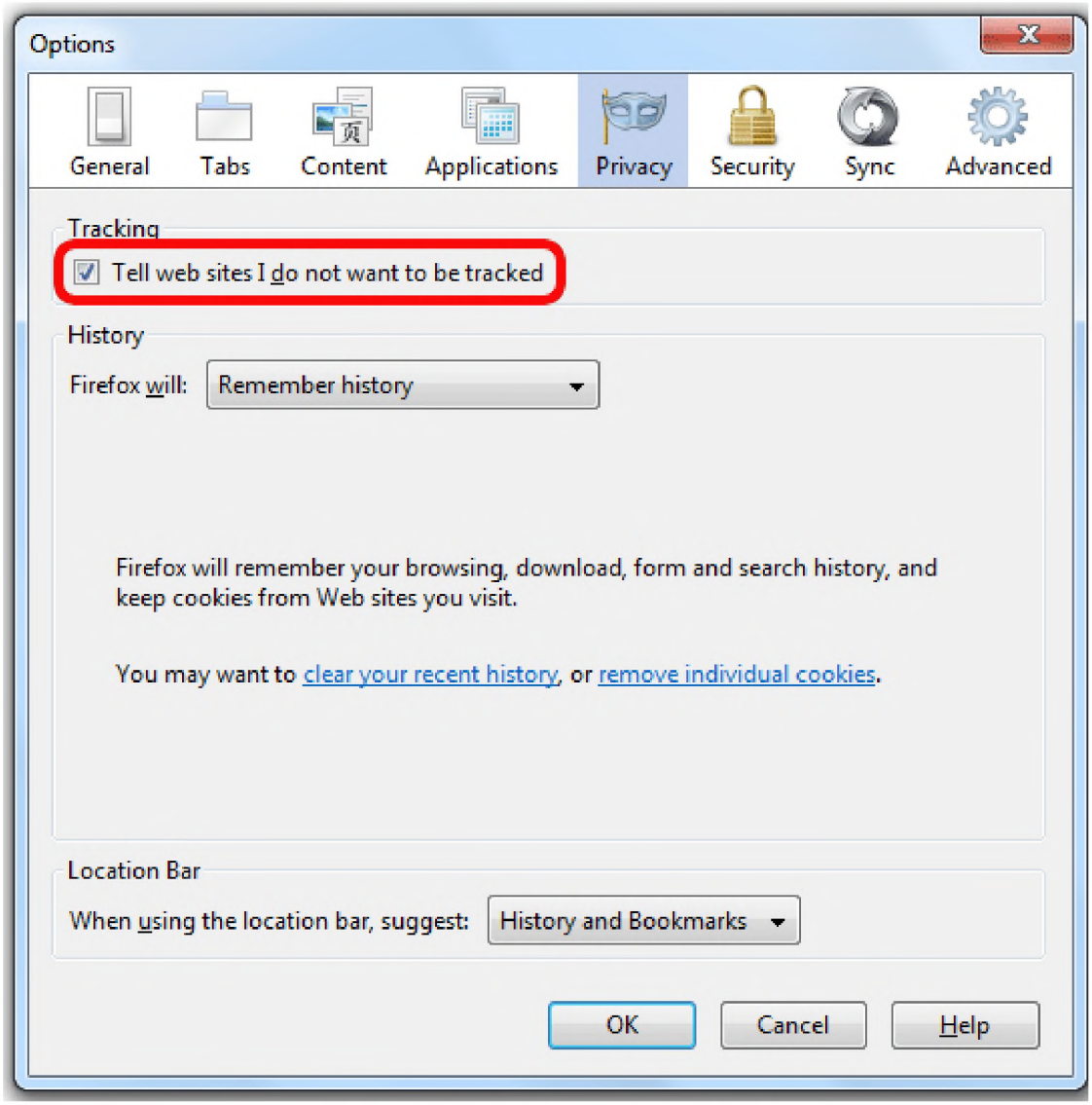


Рисунок 2.7 – Захист від стеження

Опцію. Не стеж за мною говорить всім сайтам, які відвідуєте користувач (так само, як і їх рекламодавцям і інших постачальників контента), що ви не хочете, щоб ваша поведінка в Інтернеті відстежувалася. Дотримання цієї опції добровільно – окремі сайти не зобов'язані дотримуватися її. Сайти, дотримуються цю функцію, повинні автоматично припиняти відстежувати вашу поведінку без будь-яких додаткових дій з вашого боку. Включення опції захисту від стеження наведено на рисунку 2.7.

2.4.1.7 Інтеграція з антивірусом

Для перевірки при завантаженні файлу, необхідно щоб антивірусна програма автоматично перевіряла його, захищаючи комп’ютер від вірусів і

шкідливого програмного забезпечення, яке інакше може атакувати його. Для уникненні цієї загрози рекомендується застосовувати додаток від розробників відомого антивіруса Dr.Web, яке вміє перевіряти посилання на предмет вірусів – Dr.Web antivirus link checker. Перевірка посилання на предмет вірусів наведена на рисунку 2.8.

Рисунок 2.8 – Перевірка посилання на предмет вірусів

2.5 Висновок

Результатом проведеної роботи в даному розділі став вибір профілю захищеності для Web-додатків, аналіз загроз Web-додатків, та їх детальний опис.

Створений шкідливий java-аплет (додаток Б), який призводить до переповнення буфера і виходу системи зі строю. Проведена його реалізація до і після проектних рішень. У першому випадку атака пройшла успішно, у другому – була зупинена із-за неможливості виходу роботи java-аплета з "зони довіри".

РОЗДІЛ 3. ЕКОНОМІЧНИЙ РОЗДІЛ

Метою розділу є обґрунтування економічної доцільності розробки заходів захисту web-додатків від інформаційних атак на основі java-аплетів. Обґрунтування економічної доцільності здійснюється, виходячи з розрахунку капітальних та експлуатаційних витрат, визначення величини економічного ефекту, а також показників економічної ефективності щодо запропонованих рішень із забезпечення інформаційної безпеки.

3.1 Розрахунок (фіксованих) капітальних витрат

Капітальні (фіксовані) витрати визначаються величиною коштів, призначених для створення і придбання основних фондів і нематеріальних активів, що підлягають амортизації.

Капітальні (фіксовані) витрати на проектування та впровадження проектного варіанта системи інформаційної безпеки складають:

$$К = К_{пр} + К_{зпз} + К_{пз} + К_{аз} + К_{навч} + К_{н}$$

де $К_{пр}$ – вартість розробки проекту інформаційної безпеки та залучення для цього зовнішніх консультантів;

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

$К_{пз}$ – вартість створення основного й додаткового програмного забезпечення;

$К_{аз}$ – вартість закупівлі апаратного забезпечення та допоміжних матеріалів;

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

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

3.1.1. Визначення витрат на підвищення рівня інформаційної безпеки підприємства шляхом розробки заходів захисту web-додатків від інформаційних атак на основі java-аплетів

3.1.1.1 Визначення трудомісткості розробки заходів захисту web-додатків від інформаційних атак на основі java-аплетів

Трудомісткість розробки визначається тривалістю кожної робочої операції:

$$t = t_{тз} + t_в + t_а + t_р + t_д, \quad \text{годин},$$

де $t_{тз}$ – тривалість складання технічного завдання на розробку заходів захисту web-додатків від інформаційних атак на основі java-аплетів, $t_{тз}=22$;

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

$t_а$ – тривалість аналізу існуючих загроз безпеки інформації, $t_а=20$;

$t_р$ – тривалість розробки засобів захисту WEB-додатків від атак реалізованих за допомогою java-аплетів (миттєвоє ідентифікація Web-сайту; політика безпеки вмісту браузера; безпечних оновленнь; захисту від шкідливих сайтів; приватного перегляду; захисту від стеження; інтеграції з антивірусом), $t_м=140$;

$t_д$ – тривалість підготовки технічної документації, $t_д=20$.

Отже,

$$t = 22+24+20+140+20 = 226 \text{ годин}.$$

3.1.1.2. Розрахунок витрат на підвищення рівня інформаційної безпеки підприємства шляхом розробки заходів захисту web-додатків від інформаційних атак на основі java-аплетів

Витрати на розробку заходів безпеки Кпз складаються з витрат на заробітну плату спеціаліста з інформаційної безпеки *Ззп* і вартості витрат машинного часу *Змч*:

$$К_{пз} = З_{зп} + З_{мч} = 89496 + 3509{,}78 = 93005{,}78 \text{ грн.}$$

$$З_{зп} = t\, З_{пр} = 226*396 = 89496 \text{ грн.}$$

де t – загальна тривалість операцій, годин;

Зпр – средньогодинна заробітна плата спеціаліста с інформаційної безпеки з нарахуваннями, грн/годину.

Вартість машинного часу для налагодження програми на ПК визначається за формулою:

$$З_{Мч} = t \cdot С_{Мч} = 226 \cdot 15{,}53 = 3509{,}78 \text{ грн.}$$

де t – трудомісткість операцій із побудови ефективної системи доступу персоналу, годин;

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

Вартість 1 години машинного часу ПК визначається за формулою:

$$С_{Мч} = 0{,}8 \cdot 11 \cdot 1{,}55 + \frac{4700 \cdot 0{,}5}{1920} + \frac{3200 \cdot 0{,}4}{1920} = 15{,}53 \text{ грн.}$$

При розробці заходів із забезпечення кібербезпеки через захист web-додатків від інформаційних атак на основі java-аплетів планується використовувати вже наявне апаратне забезпечення, тому капітальні витрати у зв’язку з його закупівлею не виникають.

Оскільки аплет є невеликою програмою, в якій повинно бути визначено декілька обов'язкових функцій, також аплет завантаження по мережі і

можуть виконуватися на Web-браузері, який підтримує Java, тому витрати на закупівлю додаткового програмного забезпечення також не виникають.

Але планується здійснення витрат на витрати на навчання технічних фахівців і обслуговуючого персоналу, які складуть 3000 грн. ($К_{навч}$=3000 грн.), а також витрати на налагодження системи інформаційної безпеки в розмірі 10000 грн. ($К_н$=10000 грн.)

Отже, капітальні (фіксовані) витрати на підвищення рівня інформаційної безпеки підприємства шляхом розробки заходів захисту web-додатків від інформаційних атак на основі java-аплетів складуть:

$$К = 93005,78 + 3000 + 10000 = 106005,8 \text{ грн.}$$

3.1.1 Розрахунок поточних витрат

Річні поточні витрати на функціонування системи інформаційної безпеки складають:

$$С = С_в + С_к + С_{ак}, \quad \text{грн.}$$

де $С_в$ - вартість відновлення й модернізації системи;

$С_к$ - витрати на керування системою в цілому;

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

При розробці заходів захисту web-додатків від інформаційних атак на основі java-аплетів витрати на відновлення й модернізацю системи не виникають у зв'язку з використанням з цією метою налаштувань Web-браузерів.

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 становлять 20000 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 20000 грн. Додаткова заробітна плата – 8% від основної заробітної плати. Отже,

$$С_з = 20000*12 + 20000*12*0,08= 259200 \text{ грн.}$$

Ставка ЄСВ для всіх категорій платників з 01.09.2020 р. складає 22%.

$$С_{єв} = 259200*0,22 = 57024 \text{ грн.}$$

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=6,4 кВт);

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

$Ц_е$ – тариф на електроенергію, ($Ц_е$ = 1,55 грн./кВт за годину).

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

$$С_{ел} = 6,4*1920*1,55 = 19046,4 \text{ грн.}$$

Витрати на технічне й організаційне адміністрування та сервіс системи інформаційної безпеки визначаються у відсотках від вартості капітальних витрат - 2% (Стос = 106005,8*0,02=2120,1 грн).

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

$$С_к = 259200 + 57024 + 19046,4 + 2120,1 = 337390,5 \text{ грн.}$$

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

$$С = 337390,5 \text{ грн.}$$

3.2 Оцінка можливого збитку

3.2.1 Оцінка величини збитку

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

Необхідні *вихідні дані* для розрахунку:

$t_п$ – час простою вузла або сегмента корпоративної мережі внаслідок атаки, 5 години;

$t_в$ – час відновлення після атаки персоналом, що обслуговує корпоративну мережу, 3 години;

$t_{ви}$ – час повторного введення загубленої інформації співробітниками атакованого вузла або сегмента корпоративної мережі, 12 годин;

$З_о$ – заробітна плата обслуговуючого персоналу (адміністраторів та ін.), 16000 грн./міс.;

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

$Ч_о$ – чисельність обслуговуючого персоналу (адміністраторів та ін.), 10 осіб.;

$Ч_с$ – чисельність співробітників атакованого вузла або сегмента корпоративної мережі, 30 осіб.;

О – обсяг прибутку атакованого вузла або сегмента корпоративної мережі, 800 тис. грн. у рік;

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

I – число атакованих сегментів корпоративної мережі, 3;

N – середнє число атак на рік, 32.

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:

$$П_П = \frac{\sum Зс}{F} \cdot t_п = \frac{18000 \cdot 30}{176} \cdot 5 = 15340{,}91 \quad \text{грн},$$

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 $t_{ви}$:

$$П_{ви} = \frac{\sum Зс}{F} \cdot t_{ви} = \frac{18000 \cdot 30}{176} \cdot 12 = 36818{,}18 \text{ грн.}$$

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

$$П_{пв} = \frac{\sum Зо}{F} \cdot t_в = \frac{16000 \cdot 10}{176} \cdot 3 = 2727{,}27 \text{ грн.}$$

Таким чином, витрати на відновлення працездатності вузла або сегмента корпоративної мережі складають:

$$П_в = 36818,18+2727,27 = 39545,45 \text{ грн.}$$

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:

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

$$V = \frac{800000}{2080} \cdot (5+3+12) = 7692,31 \text{ грн.}$$

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.

$$U = 15340,91 + 39545,45 + 7692,31 = 62578,67 \text{ грн.}$$

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

$$B = \Sigma_i \Sigma_n U = \Sigma_3 \Sigma_{32} 35222,03 = 6007552,32 \text{ грн.}$$

3.2.2 Загальний ефект від впровадження системи інформаційної безпеки

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

$$E = B \cdot R - C \text{ грн.},$$

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

R – вірогідність успішної реалізації атаки на сегмент мережі, частки одиниці (R=0,15);

C – щорічні витрати на експлуатацію системи інформаційної безпеки, тис. грн.

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

$$E = 6007552,32*0,15 – 337390,5 = 864120 \text{ грн.}$$

3.3 Визначення та аналіз показників економічної ефективності системи інформаційної безпеки

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:

$$ROSI = \frac{563742,3}{106005,8} = 5,31, \quad \text{частки одиниці,}$$

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, (5,5 %);

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

Розрахункове значення коефіцієнта повернення інвестицій:

$$5,31 > (5,5 – 5)/100 = 5,31 > 0,005.$$

Термін окупності капітальних інвестицій T_o показує, за скільки років капітальні інвестиції окупляться за рахунок загального ефекту від впровадження системи інформаційної безпеки. Відповідно термін окупності розробки заходів захисту web-додатків від інформаційних атак на основі java-аплетів:

$$T_O = \frac{K}{E} = \frac{1}{ROSI} = \frac{1}{5,31} = 0,19,$$ років (2,26 місяця).

3.4 Висновок

Відповідно до проведених розрахунків можна зробити висновок, що розробка заходів захисту web-додатків від інформаційних атак на основі java-аплетів є економічно доцільною, оскільки передбачає отримання економічного ефекту у розмірі 5,31 грн. на 1 грн. капітальних вкладень (ROSI=5,31). Термін окупності при цьому складе 0,19 року або 2,26 місяця. При капітальних витратах на рівні 106005,8 грн. передбачаються щорічні експлуатаційні витрати 337390,5 грн.

ВИСНОВКИ

Мова Java дає програмістам можливість не просто розробляти нові програми, але і використовувати елементи вже написаних і перевірених програм. Такий модульний принцип дозволяє швидко писати нові програмні продукти та ефективно модернізувати старі. Крім того, у стандарт мови входить безліч корисних бібліотек, на основі яких можна будувати обчислювальні системи будь-якої складності.

На сьогоднішній день більшість атак через Інтернет здійснюється за допомогою експлойтів, які використовують помилки в ПЗ, щоб пробити захист комп'ютера і отримати можливість виконання шкідливого коду без відома користувача. Проаналізувавши динаміку розвитку вірусів можна зробити висновок, що для розповсюдження шкідливих програм кіберзлочинці все частіше використовують незакриті вразливості в платформі Java.

У даній магістерської роботі була розглянута класифікація атаки на Web-додатки. Проведений аналіз загроз WEB-серверів та обраний профіль захищеності відповідно до НД ТЗІ 2.5-010-03.

На основі результатів дослідження були розроблені проектні рішення, що забезпечують конфіденційності, доступності та цілісності інформації.

Був проведений розрахунок витрат на впровадження засобів захисту та розрахунок передбачених збитків від атак на WEB-додатки.

Розроблено інженерно-технічні та організаційні заходи щодо охорони праці на робочому місці користувача. Здійснено розрахунок освітлення робочого місця та робочого приміщення користувача.

ПЕРЛІК ПОСИЛАНЬ

1. Развитие информационных угроз в первом квартале 2012 года (Електроний ресурс) http://www.securelist.com/ru/analysis/208050757 /razvitie_informatsionnykh_ugroz_v_pervom_kvartale_2012_goda#22 – Загол. з екрана.

2. Плично раскрытый Java-эксплойт резко повышает уровень угрозы (Електроний ресурс)/Спосіб доступу:URL: http://www.rusecurity.com/2011 /11/30/publichno-raskryityiy-java-eksployt-rezko-povyishaet-uroven-ugrozyi. – Загол. з екрана.

3. Киберпреступники продолжают активно использовать Java-эксплойты (Електроний ресурс) / http://www.3dnews.ru/software-news/609200.–Загол. з екрана.

4. "Brute Force Attack", Imperva Glossary (Електроний ресурс)/Спосіб доступу: URL: http://www.imperva.com/application_defense _center//brute_force.html.–Загол. з екрана.

5. "Protecting Secret Keys with Personal Entropy", By Carl Ellison, C. Hall, R. Milbert, and B. Schneier (Електроний ресурс)/ Спосіб доступу: URL: http://www.schneier.com/paper-personal-entropy.html. – Загол. з екрана.

6. "Defense: Brute-Force Exploitation of Web Application Session ID's", By David Endler – DEFENSE Labs (Електроний ресурс)/ Спосіб доступу: URL: http://www.cgisecurity.com/lib/SessionIDs.pdf. – Загол. з екрана.

7. "Best Practices in Managing HTTP-Based Client Sessions", Gunter Ollmann - X-Force Security Assessment Services EMEA (Електроний ресурс)/ Спосіб доступу: URL: http://www.technicalinfo.net/papers/ WebBasedSessionManagement.html. – Загол. з екрана.

8. "A Guide to Web Authentication Alternatives", Jan Wolter (Електроний ресурс)/ Спосіб доступу: URL: http://www.unixpapa.com/auth/homebuilt.html . – Загол. з екрана.

9. "Brute Force Attack", Imperva Glossary (Електроний ресурс)/Спосіб доступу: URL: http://www.imperva.com/application_defense _center/glossary/brute_force.html.– Загол. з екрана.

10. "Dos and Don'ts of Client Authentication on the Web", Kevin Fu, Emil Sit, Kendra Smith, Nick Feamster - MIT Laboratory for Computer Science (Електроний ресурс) /Спосіб доступу: URL: http://cookies.lcs.mit.edu/pubs/webauth:tr.pdf . – Загол. з екрана.

11. "Session Fixation Vulnerability in Web-based Applications", By Mitja Kolsek – Acros Security (Електроний ресурс) / Спосіб доступу: URL: http://www.acrossecurity.com/papers/session_fixation.pdf . – Загол. з екрана.

12. "Divide and Conquer", By Amit Klein – Sanctum(Електроний ресурс)/ Спосіб доступу: URL: http://www.sanctuminc.com/pdf/ whitepaper_httpresponse.pdf . – Загол. з екрана;

13. Классификация угроз безопасности Web-приложений (Електроний ресурс)/ Спосіб доступу: URL: http://www.infosecurity.ru/iprotect/ websec/classification/.– Загол. з екрана.

14. Java (Електроний ресурс) Спосіб доступу: URL http://ru.wikipedia.org/wiki/Java /.– Загол. з екрана.

15. Какие можно предпринять действия по повышению безопасности Java. (Електроний ресурс)/ Спосіб доступу: URL: http://www.java.com/ru/ download/faq/tips.xml /.– Загол. з екрана.

16. С. Воронов “ВИРТУАЛЬНАЯ БЕЗОПАСНОСТЬ “ Chip, январь 2002.

17. М. Мамаев, С. Петренко “WORLD WILD WEB ИЛИ ДИКАЯ ПАУТИНА” Chip,январь 2002.

18. Статистика зароботної плати/ Спосіб доступу: URL: http://trud.dp.ua/vacancyview.php?pagenumber=4&branch=7&area=3/.– Загол. з екрана.

19. НД ТЗІ 2.5-010-03. Вимоги до захисту інформації WEB-сторінки від несанкціонованого доступу.

20. НД ТЗІ 2.5-005-99. Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу.

21. НД ТЗІ 2.5-004-99. Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу.

ДОДАТОК А. Відомість матеріалів кваліфікаційної роботи

№	Формат	Найменування	Кількість листів	Примітка
1	А4	Реферат	3	
2	А4	Список умовних скорочень	1	
3	А4	Зміст	3	
4	А4	Вступ	1	
5	А4	1 Розділ	40	
6	А4	2 Розділ	28	
7	А4	3 Розділ	11	
8	А4	Висновки	1	
9	А4	Перелік посилань	3	
10	А4	Додаток А	1	
11	А4	Додаток Б	5	
12	А4	Додаток В	1	
13	А4	Додаток Г	1	
14	А4	Додаток Д	1	

ДОДАТОК Б. Зловмисний java-аплет

Файл NewJApplet1.java

```
package helloapplet;

import java.awt.AWTException;
import java.awt.Color;
import java.awt.Robot;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyEvent;
import java.awt.event.KeyListener;
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JApplet;
import javax.swing.JButton;
import javax.swing.JOptionPane;

public class NewJApplet1 extends JApplet {

    /**
     * Initialization method that will be called after the applet is loaded
     * into the browser.
     */
    public boolean b;
    public void init() {
        try {
            // TODO start asynchronous download of heavy resources
```

```
final Robot r = new Robot();
final JButton jb = new JButton("Нажми на меня ;)");
b=true;
jb.addMouseListener(new MouseListener() {

    @Override
    public void mouseClicked(MouseEvent e) {
        b=!b;
    }
   @Override
    public void mousePressed(MouseEvent e) {

    }

    @Override
    public void mouseReleased(MouseEvent e) {
    }

    @Override
    public void mouseEntered(MouseEvent e) {
    }

    @Override
    public void mouseExited(MouseEvent e) {
    }
});
jb.addMouseMotionListener(new MouseMotionListener() {

    @Override
    public void mouseDragged(MouseEvent e) {
```

```
}

            @Override
            public void mouseMoved(MouseEvent e) {
               if (b==true)
               {
                  if (e.getLocationOnScreen().getY()>jb.getHeight()-100 ||
                        e.getLocationOnScreen().getY()<100 ||
                        e.getLocationOnScreen().getX()>jb.getWidth()-100 ||
                        e.getLocationOnScreen().getX()<100)
                     r.mouseMove((jb.getWidth()/2)-100, (jb.getHeight()/2)-100);
               }
            }
         });
         add(jb);
      }
      // TODO overwrite start(), stop() and destroy() methods
      catch (AWTException ex) {
         Logger.getLogger(NewJApplet1.class.getName()).log(Level.SEVERE, null,
ex);
      }
   }
   // TODO overwrite start(), stop() and destroy() methods
}
```

Файл NewApplet1.java

```
package helloapplet;

import java.applet.Applet;
import java.awt.Label;
```

```
public class NewApplet1 extends Applet {

    public void init() {
    // TODO start asynchronous download of heavy resources
    Label l = new Label("Click");
    l.setAlignment(Label.CENTER);
    add(l);
  }
  // TODO overwrite start(), stop() and destroy() methods
}
```

Файл HelloApplet.java

```
package helloapplet;
public class HelloApplet {
  public static void main(String[] args) {
    // TODO code application logic here
  }
}
```

Файл NewApplet.java

```
package helloapplet;

import java.applet.Applet;
import java.awt.AWTException;
import java.awt.Graphics;
import java.awt.Robot;
import java.awt.event.MouseEvent;
```

```
import java.awt.event.MouseListener;
import java.awt.event.MouseMotionListener;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;

public class NewApplet extends Applet {
    boolean b;
    public void init() {
        // TODO start asynchronous download of heavy resources
        try {
            // TODO start asynchronous download of heavy resources
            final Robot r = new Robot();


        }
        // TODO overwrite start(), stop() and destroy() methods
        catch (AWTException ex) {
            Logger.getLogger(NewJApplet1.class.getName()).log(Level.SEVERE, null,
ex);
        }
    }
    // TODO overwrite start(), stop() and destroy() methods
}
```

ДОДАТОК В. 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

16 Додаток Д.doc

Презентація.pptx

ДОДАТОК Г. Відгуки керівників розділів

Відгук керівника економічного розділу:

Керівник розділу _____ _____

(підпис) (ініціали, прізвище)

ДОДАТОК Д. ВІДГУК

на кваліфікаційну роботу магістра на тему:
Методи захисту web-додатків від інформаційних атак на основі java-аплетів
студента групи 125м-19-1
Бєлоха Германа Костянтиновича

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, __ джерела та __ додатка.

Мета роботи: підвищення захисту WEB-додатків від атак реалізованих за допомогою java-аплетів, що можуть загрожувати конфіденційності, доступності та цілісності інформації

Зміст та структура роботи дозволяють розкрити поставлену тему повністю.

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

Робота виконана самостійно. У роботі було приведена класифікація атак на WEB-сервера, проаналізований принцип роботи java-аплетів та існуючи технології безпеки в Java. Проведений поглиблений аналіз існуючих засобів безпеки Інтернет-браузерів.

У спеціальній частині був проведений аналіз загроз WEB-додатків та розроблена модель загроз, обґрунтовано вибір профілю захищеності. Узагальнено переваги та недоліки кожного з видів Інтернет-браузерів. Запропоновано засоби захисту WEB-додатків від атак, які реалізуються за допомогою java-аплетів.

Це підтверджує самостійність обробки даних, практичні рекомендації та висновки.

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 Бєлох Герман Костянтинович заслуговує на оцінку «_____».

Керівник роботи,
д.т.н., проф. В.І. Корнієнко

Керівник спец. част.

ас. кафедри БІТ Ю.В. Ковальова