Навчальне видання

В.Д. КОЗЮРА, В.О. ХОРОШКО,
М.Є. ШЕЛЕСТ, Ю.М. ТКАЧ, О.О.БАЛЮНОВ

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХСИСТЕМАХ

Підручник

В авторській редакції

Відповідальний за випуск – *Лук'яненко В.В.*

Підписано до друку 18.01.2020 р.
Формат 60х 84/16. Папір офсетний. Друк числовий.
Гарнітура Times New Roman. Обл.-вид. арк. 18,01.
Ум. друк. арк. 13,72. Тираж 300 прим.
Зам. № 583.

Віддруковано з оригінал-макету замовника

Видавець - ФОП Лук'яненко В.В. ТПК «Орхідея»

Свідоцтво про внесення суб'єкта видавничої справи
до державного реєстру видавців, виготівників
і розповсюджувачів видавничої продукції
серія ДК № 3020 від 02.11.2007 р.

16600, Чернігівська обл., м. Ніжин, вул. Небесної сотні, 13 а.
Тел.: 068 815 06 60
E-mail: holdingvv@ gmail.com

В.Д. КОЗЮРА, В.О. ХОРОШКО,
М.Є. ШЕЛЕСТ, Ю.М. ТКАЧ, О.О. БАЛЮНОВ

ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Підручник

Чернігів
2020

УДК 681.3(075)

З-38

Витяг з протоколу засідань вченої ради №10 від 25 листопада 2019р.

РЕЦЕНЗЕНТИ:

Професор кафедри захисту інформації Національного університету "Львівська політехніка" д.т.н., доцент ***Опірський І.Р.***
Директор Центру інформаційних технологій і захисту інформації Вінницького національного технічного університету д.т.н., професор ***Яремчук Ю.Є.***
Лауреат Державної премії України в галузі науки і техніки, д.т.н., професор, професор Національного авіаційного університету ***Щербак Л.М.***
Д.т.н., доцент кафедри інформаційних та комп'ютерних систем Чернігівського національного технологічного університету ***С.В. Зайцев***

Козюра В. Д., Хорошко В. О., Шелест М. Є., Ткач Ю. М., Балюнов О.О.

З-38 Захист інформації в комп'ютерних системах: підручник. – Ніжин: ФОП Лук'яненко В.В., ТПК «Орхідея», 2020. – 236с.

ISBN **978-617-7609-44-4**

У підручнику розглянуто питання загроз інформації в комп'ютерних системах, зокрема висвітлено питання аналізу трафіку в мережі Internet, розкрито суть атак на інформацію в комп'ютерних системах, що доволі часто реалізуються, описано механізми безпеки в ОС MS Windows, криптографічні й стеганографічні методи захисту інформації в комп'ютерних системах, а також запропоновано класифікацію шкідливого програмного забезпечення та засоби захисту від них.Виклад теоретичного матеріалу супроводжується наочними матеріалами (рисунками, блок-схемами, діаграмами тощо), кожен розділ завершено висновками, питаннями для самоконтролю та літературою.

Підручник призначено для студентів спеціальності 125 «Кібербезпека» першого освітнього рівня підготовки (бакалавр, а також буде корисним магістрам (другий освітній рівень) відповідної спеціальності, аспірантам, викладачам, науковцям та фахівцям у галузі інформаційної безпеки.

УДК 681.3(075)

ISBN 978-617-7609-44-4

Для нотаток

Література

1. Конахович Г.Ф., Пузыренко А.Ю. Компьютерная стеганография. Теория и практика. – К.: «МК-Пресс», 2006. – 288 с.
2. Хорошко В.А. Введение в компьютерную стеганографию / В.А. Хорошко, М.Е. Шелест – К.: 2002. – 140 с.
3. Хорошко В.О. Комп'ютерна стеганографія / В.О. Хорошко, Ю.Є. Яремчук, В.В. Карпінець – Вінниця: ВНТУ, 2014. – 155 с.
4. Ленков С.В. Методы и средства защиты информации. В 2-х томах / С.В. Ленков, Д.А. Перегудов, В.А. Хорошко – К: Арий, 2008.
5. Хорошко В.О. Основи комп'ютерної стеганографії / В.О. Хорошко, О. Д. Азаров, М. Є. Шелест, Ю. Є. Яремчук – Вінниця: ВДТУ, 2003. – 143 с.
6. Шенон К. Теория связи в секретных системах. В «Работы по теории информации и кибернетике», с. 333 – 402. – М: Изд. ИЛ, 1963.
7. Перепелицын Е.Г. Нестандартные методы математической статистики и их применение к технической диагностике и анализу изображений / Е.Г. Перепелицын – М.: Омега – Л, 2006, – 312 с.

ЗМІСТ

рівнів формується нотне відображення розширеного музичного твору з впровадженим в нього прихованим повідомленням. На підставі нотного відображення розширення здійснюється його музична реалізація за допомогою сучасних комп'ютерних систем, що представляють собою програмно-апаратні синтезатори звуку.

Подальша звукова обробка музичних записів, оброблених стегосистемою, не обов'язкова. Оскільки основна область застосування музичних стегосистем - це середовище Internet, в якому музичні записи розміщуються в цифровому форматі на Web-сторінках, то досить, щоб розширений музичний твір сприймався сторонніми особами не як шум, а як деяка музика, яка містить мелодію або сукупність мелодій, які допускають ту чи іншу тематичну інтерпретацію.

Висновки

1. Стеганографія – міждисциплінарна наука і мистецтво про приховану передачу або зберігання інформації з урахуванням збереження в таємниці самого факту такої передачі (зберігання). Метою стеганографії є створення прихованої передачі даних, цифрового відбитку і стеганографічного водяного знаку.
2. Комп'ютерна стеганографія – це використовування особливостей комп'ютерної платформи, а цифрова стеганографія – це напрям, заснований на прихованні або впровадженні секретної інформації в цифрові мультимедіа-об'єкти (зображення, відео, аудіо, текстури 3D-об'єктів і т.д.), викликаючи при цьому деякі спотворення цих об'єктів.

Питання для самоконтролю

1. Що є комп'ютерна стеганографія?
2. Які основні елементи включає модель стеганографічної системи?
3. Визначите основні напрями розвитку стеганографії.
4. Дайте класифікацію методів приховання інформації в стегосистемах.
5. Які методи текстової стеганографії застосовуються?
6. Методи потворення формату текстового документа.
7. Які методи приховування інформації у забраженні застосовуються?
8. У чому особливості методів приховування інформації у частотній зоні?
9. У чому полягають структурні методи та методи викривлення?
10. Особливості стеганографічних методів приховування інформації у звуковому середовищі.
11. Що таке музикальна стеганографія?

в радіомережах загального призначення і поширюється на електронних носіях інформації, які, в зв'язку з розвитком комп'ютерної техніки, набули широкого поширення. У зв'язку з цим використання музичного середовища для приховування інформаційних повідомлень є досить перспективним. Крім методів, описаних вище, для приховування даних можна застосовувати методи, засновані на модифікації тих параметрів музичного середовища, які в теорії музики можна описати якісно. Музичне середовище має своє текстове відображення у вигляді нот і інших знаків, які дозволяють досить адекватно відображати музичний твір і його внутрішню структуру такими елементами, як ноти, гами, періоди, такти, каденції, акорди, мотиви, модуляції, тональності, різні види розвитку, секвенції та ін. Побудови музичних фрагментів підкоряються синтаксичним правилам, які можна описати, що дозволяє будувати логічні взаємовідносини і, відповідно, опис структур музичних творів.

Музичні стегосистеми забезпечують приховування інформації в музичному середовищі за аналогією з імпровізацією музичних творів. По суті, імпровізація - це така зміна музичного твору або його фрагментів, яка зберігає основні теми первісного твору у вигляді мелодій, але при цьому розширює образ музичної теми іншими рисами, які доповнюють основний образ і яких не було в основному музичному творі. Основна відмінність музичної стеганографії від імпровізації полягає в тому, що метою є не розширення образів базового музичного твору, а внесення змін, які зберігають мелодію основного твору, відповідають всім правилам побудови даного твору і при цьому кодують приховуване повідомлення, не спотворюючи головної теми твору.

Фрагмент музичного твору може бути описаний у вигляді деякої логічної структури. Аналогом слова текстового речення в музичному творі буде один такт мелодії, а аналогом речення в музиці вважатимемо фрагменти, що розділяються цензурами. Як правило, музичний твір складається з ряду фраз, які складаються з тактів. Нехай є фрагмент мелодії, який представляє слово тексту у вигляді співвідношення $\beta(i,\ j)+...+\beta(i+k,j+r)=x_i(t)$, а також фрагмент мелодії, записаний у вигляді співвідношення $\alpha(\eta,\xi)+...+\alpha(\eta+e,\xi+q)=x_n(m)$. Впровадження тексту в музичний твір здійснюється окремими реченнями, кожне з яких може зіставити з окремою мелодією.

На першому етапі роботи стегосистеми аналізується кількість мелодій (кількість її модифікацій) в рамках музичного твору в зіставленні з кількістю речень повідомлення. На другому етапі здійснюється аналіз допустимості розширення деякого речення музичного твору реченнями тексту повідомлення. Цей аналіз проводиться на основі дослідження логічних формул тексту речення $L(t)$ і музичногоречення $L(m)$. На наступному етапі, в разі вибору відповідної пари $L(m)$ і $L(t)$, здійснюється аналіз наслідування фраз мелодій, окремих слів тексту і слів мелодії, що відповідає погодженню пар на рівні опису $x_i(t)$ і $x_\eta(m)$. Після позитивного вирішення завдань перерахованих

ВСТУП

Розвиток глобального інформаційного суспільства, швидкий темп розвитку сучасних IT-технологій у всіх сферах діяльності (як в державному так і в приватному секторі), а також їх стрімке розповсюдження серед широких мас населення, обумовлюють необхідність забезпечення кібернетичного захисту інформації, тобто забезпечення захисту інтересів особи, суспільства та держави від зовнішніх і внутрішніх загроз, що стосуються питань кібербезпеки. Тому серед основних завдань, визначених Указом Президента України № 96/2016 від 27.01.2016 року у "Стратегії кібербезпеки України", є створення в Україні національної системи кібербезпеки.

Принцип комплексного рішення завдань захисту інформації припускає застосування разом з традиційними апаратно-програмними засобами і способами, організаційними і нормативно-правовими заходами захисту, також сучасних засобів, зокрема, криптографічних і стеганографічних. Унікальність цих методів і засобів захисту інформації як безпосередньо в КС, так і в зовнішніх каналах зв'язку, полягає в тому, що вони забезпечують найнадійніший шлях захисту, бо охороняють безпосередньо саму інформацію, а не доступ до неї. Криптографічні методи дозволяють, окрім конфіденційності інформації, забезпечити її цілісність і достовірність, організувати процедуру автентифікації абонентів, що обмінюються інформацією. На відміну від криптографії, стеганографія включає сукупність методів, що ґрунтуються на різних принципах, які забезпечують приховання самого факту існування секретної інформації в тому або іншому середовищі, а також засобів реалізації цих методів.Метою стеганографії є створення прихованої передачі даних(ППД), цифрового відбитку (ЦВ) та стеганографічного водяного знаку (СВЗ).

Проблема забезпечення безпеки інформаційних ресурсів та систем також вирішується шляхом застосування вбудованих механізмів в операційну систему (ОС). Засоби безпеки ОС включають такі механізми, як облікові записи, паролі і захист файлів. Вони також включають такі менш помітні механізми, як захист ОС від ушкодження, недопущення здійснення ряду дій (наприклад, перезавантаження комп'ютера) з боку менш привілейованих користувачів і заборона несприятливої дії призначених для користувача програм на програми інших користувачів або на операційну систему. Таким чином, операційна система, разом з окремими користувачами, має можливість захистити файли, пам'ять і налаштування конфігурації від небажаного перегляду і зміни інформації.

Але незважаючи на існуючі засоби та механізми захисту інформації в тому числі потужні антивірусні пакети, небезпека зараження комп'ютерів не лише не зменшується, але продовжує зростати. Шкідливі програми свідомо призначені для несанкціонованого знищення, блокування, модифікації, копіювання комп'ютерної інформації або нейтралізації засобів захисту комп'ютерної інформації. Вони завдають значної шкода, а саме починаючи від відкриття-закриття піддону CD/DVD-ROM і закінчуючи знищенням даних і поломкою апаратного забезпечення (поломками відомий, зокрема, Win32.CIH). Шкідливі програми можуть інсталювати інше шкідливе ПЗ, здійснювати крадіжку, займатись шахрайством, здирництвом, шпигунством за

ввести відповідний сигнал-відлуння, в залежності від приховуваного біта: $c(t) = v(t) + \alpha v(t - \Delta)$.

У базовій схемі передбачено приховування в аудіосигналах одного біта, але сигнал можна розбити випадковим чином на l відрізків і в кожен з них вставити по біту. Для виділення сигналу-відлуння і відновлення прихованих даних застосовується автокореляційний аналіз. Як стегоключ тут зазвичай використовуються значення величин Δ_1 і Δ_0 з урахуванням обраних меж відрізків.

Фазові методи приховування застосовуються як для аналогового, так і для цифрового сигналу. Вони використовують той факт, що плавну зміну фази на слух визначити не можна. В таких методах дані, що захищаються, кодуються або певним значенням фази, або зміною фаз в діапазоні. Якщо розбити звуковий сигнал на сегменти, то дані зазвичай приховують тільки в першому сегменті при дотриманні двох умов:

— збереження відносних фаз між послідовними сегментами;

— результуючий фазовий спектр стегосигналу повинен бути плавним, оскільки різкі стрибки фази є демаскуючим фактором.

Розглянемо приховування даних шляхом зсуву фази. Сигнал контейнера розбивається на N коротких сегментів $c_i(n)$ довжиною $I(m)$, і за допомогою ШКФ будується матриця фаз $\varphi_i(k)$ і амплітудний спектр $A_i(k)$:

$$\varphi_i(k) = \arctan\frac{\mathrm{Im}\left[F\{c_i\}(k)^2\right]}{\mathrm{Re}\left[F\{c_i\}(k)^2\right]} \quad \text{и} \quad A_i(k) = \sqrt{\mathrm{Re}\left[F\{c_i\}(k)\right]^2 + \mathrm{Im}\left[F\{c_i\}(k)\right]^2}\,.$$

У зв'язку з тим, що фазові зрушення між двома сусідніми сегментами можуть бути легко виявлені, в стегосигналі повинні бути збережені різниці фаз. Тому секретне повідомлення вбудовується тільки в фазу першого сегмента:

$$\overline{\varphi}_0(k) = \begin{cases} \pi/2, & \text{якщо} \quad m_k = 0; \\ -\pi/2, & \text{якщо} \quad m_k = 1. \end{cases}$$

Крім того, створюється нова матриця фаз:

$$\overline{\varphi}_i(k) = \overline{\varphi}_0(k) + \left[\overline{\varphi}_1(k) - \overline{\varphi}_0(k)\right]$$

. . .

$$\overline{\varphi}_N(k) = \overline{\varphi}_{N-1}(k) + \left[\overline{\varphi}_N(k) - \overline{\varphi}_{N-1}(k)\right].$$

Після цього за допомогою ОБПФ створюється стегосигнал з використанням нової матриці фаз і амплітудного спектра $A_i(k)$. Таким чином, зі зміною початкової фази $\varphi_0(k)$ фази всіх наступних сегментів будуть змінені на відповідну величину. Під час вилучення прихованого значення одержувач секретної інформації, знаючи довжину послідовності $c(m)$, зможе обчислити БПФ і виявити фази $\varphi_0(k)$.

5.4.2 Музичні стеганосистеми

Музична форма звукового середовища займає більшу частину інформаційного простору Internet. Крім цього вона широко використовується

ровому форматі через будь-яку мережу передачі даних. Відомо, що слуховий апарат людини функціонує в широкому динамічному діапазоні; він дуже чутливий до випадкових адитивних перешкод, здатний розрізняти відносну фазу, зовсім нечутливий до абсолютної фази. Ці особливості слухового апарату дозволяють успішно застосовувати стеганографічні методи в аудіосередовищі.

5.4.1 Стеганографічні методи захисту данних у звуковому середовищі

Метод найменших важливих бітів застосовується при цифровому поданні аудіосигналу і придатний для використання за будь-яких швидкостей зв'язку. При перетворенні звукового сигналу в цифрову форму завжди присутній шум дискретизації, який не вносить суттєвих спотворень. "Шумовим" бітам відповідають молодші біти цифрового представлення сигналу, які можна замінити приховуваними даними. Наприклад, якщо звуковий сигнал представлений в 16-бітовому вигляді, то зміна чотирьох молодших бітів не приведе до помітних на слух спотворень. Як стегоключ зазвичай використовується покажчик місця розташування бітів, в яких містяться приховувані дані.

Методи широкосмугового кодування використовують ті ж принципи, що й методи приховування даних в зображеннях. Їх суть полягає в незначній одночасній модифікації цілого ряду певних бітів контейнера при приховуванні одного біта інформації. Існує кілька різновидів методу. У найбільш поширеному варіанті вихідний сигнал модулюється високошвидкісною псевдовипадковою послідовністю $w(t)$, яка визначена на області значень $\{-1, 1\}$. Внаслідок цього для передачі результату необхідна велика (іноді більша ніж в 100 разів) смуга пропускання. Зазвичай послідовності $w(t)$ вибирають ортогональними до сигналу контейнера. Результуючий стегосигнал $s(t)$ являє собою сумарний сигнал контейнера $c(t)$ і приховуваних даних $d(t)$:

$$s(t) = v(t) + \alpha \times d(t) \times w(t),$$

де коефіцієнт загасання призначений для вибору оптимального рівня шуму, який вноситься вставленими даними.

Для вилучення прихованих даних $d(t)$ на стороні, яка приймає дані, необхідно мати ту саму псевдовипадкову імпульсну послідовність $w(t)$, забезпечивши при цьому її синхронізацію зі стегосигналом: $s(t) \times w(t) = v(t) \times w(t) + \alpha \times d(t)$. У зв'язку з цим дану псевдовипадкову бітову послідовність зазвичай використовують в якості стегоключа.

Метод приховування в сигналі-відлунні. Приховувати дані можна також шляхом впровадження в звуковий сигнал відлуння. Відомо, що при невеликих часових зрушеннях сигнал-відлуння практично не відрізнити на слух. Тому, якщо ввести певні тимчасові затримки (наприклад, Δ_1 для одиничного біта даних і Δ_0 – для нульового), величина яких не перевищує поріг виявлення, то, розбиваючи вихідний звуковий сигнал $v(t)$ на сегменти, в кожен з них можна

користувачем тощо. Шахраї через шкідливе ПЗ можуть використовувати ресурси зараженого комп'ютера в злочинних цілях: отримання несанкціонованого (і/або дармового) доступу до ресурсів самого комп'ютера або третіх ресурсів, доступних через нього, у тому числі пряме управління комп'ютером (*backdoor*);організація на комп'ютері відкритих релеєв і загальнодоступних проксі-серверів;використання заражених комп'ютерів (у складі ботнету) для проведення DDoS-атак; збір адрес електронної пошти і поширення спаму, у тому числі у складі ботнета;накрутка електронних голосувань, клацань по рекламних банерах;генерація монет платіжної системи Bitcoin; поширення інших шкідливих програм (наприклад, «троянських коней», що поширюють віруси);пряме управління комп'ютером;дезактивація антивірусів і брандмауерів тощо.

Отже, питання підготовки сучасних фахівців з кібербезпеки для органів та формувань, що входять до складу сектору безпеки і оборони України, а також забезпечення загальної кіберосвіти широких верств населення, стає одним із найбільш пріоритетних завдань, як однієї із важливих складових сфер національної безпеки і оборони держави в умовах ведення гібридної війни проти України. Таким чином, навчання підходів та методів захисту інформації в комп'ютерних системах (КС) є однією зі складових у системі навчання і виховання кадрів у сфері кібербезпеки у вищих навчальних закладах України.

Автори висловлюють глибоку подяку за уважне та доброзичливе рецензування, за висловлені зауваження і поради, які сприяли значному покращенню та поглибленню навчального підручника.

1. ЗАГРОЗИ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Інформаційна безпека – багатогранна, багатовимірна область діяльності, в якій успіх може принести тільки *системний*, *комплексний* підхід до побудови систем захисту інформації.

Система захисту інформації повинна уміти *протистояти різноманітним атакам*, як зовнішнім, так і внутрішнім, атакам автоматизованим і скоординированным.

1.1 Поняття безпеки інформації в комп'ютерних системах

1.1.1 Зміст і основні поняття комп'ютерної безпеки

Захист інформації – проблема з прадавніх часів.

Специфіка комп'ютерної форми представлення інформації (провокує на посягання):

- ❒ можливість діставання доступу до великих об'ємів інформації в локальному фізичному зосередженні;
- ❒ можливість швидкого або миттєвого копіювання величезних об'ємів інформації і, як правило, без слідів;
- ❒ можливість швидкого або миттєвого руйнування або спотворення величезних об'ємів інформації.

В результаті – комп'ютерні системи і інформаційна безпека – невід'ємні поняття.

Захист (забезпечення) безпеки інформації – не просто допоміжний, але одна з головних (основних) функцій КС при їх створенні і експлуатації.

Основні етапи розвитку теорії і практики комп'ютерної безпеки:

Етап	Роки	Основні чинники	Зміст
Початковий	1960 - 1970-і роки	• Поява ЕОМ 3-го покоління • Початок застосування ЕОМ для інформаційного забезпечення великих підприємств і організацій	• Початок теоретичних досліджень проблем захисту комп'ютерної інформації • Дослідження і перші реалізації технологічних аспектів захисту інформації (парольні системи автентифікації) • «Відкриття» криптографії у внедержавної сфері
Другий	1970 - 1980-і роки	• Широке впровадження ЕОМ в інформаційне забезпечення не лише великих, але і середніх під-	• Інтенсивні теоретичні дослідження формальних моделей безпеки (Хоффман (1970-1974), Хартсон (1975), Харрисон, Рузо, Ульман

кілька інваріантів, які описуються у вигляді многочлена. Секретний ключ при такому підході - це спосіб нумерації графа. Відомо, що можлива кількість перенумерованих графів для довільного графа досить велика. Ця обставина робить запропонований спосіб приховування повідомлень досить стійким проти атак розтину.

У структурних методах можна виділити окремі етапи стеганографічного перетворення.

Першим етапом є перетворення секретного повідомлення, яке захищається, m в цифрову форму CH. Це перетворення може бути, наприклад, будь-яким криптографічним перетворенням. Воно являє собою шифрування тексту з усіма відповідними атрибутами, включаючи ключі шифрування.

Другий етап являє собою перетворення послідовності чисел CH в графічну структуру GS. Як графічні структури найчастіше використовуються графи. Крім графів, можна використовувати різні піктограми або інші структури, які піддаються формальному опису у той чи інший спосіб..

На третьому етапі здійснюється перетворення графічної структури в візуальне інформаційне середовище WS. У загальному випадку, у якості такого середовища може використовуватися, наприклад, будь-яке мультимедійне або програмне середовище.

Четвертий етап являє собою сукупність методів і відповідних процедур, за допомогою яких формується сюжет із візуальних образів із впровадженими в них таємними повідомленнями.

В рамках даного підходу візуальний образ складається з графічних елементів, які ідентифікуються з елементами GS. Дані елементи являють собою позначені вершини, позначені або не позначені ребра і інші елементи, що ідентифікують компоненти з CH. Необхідним етапом функціонування такої стегосистеми є формування деякого сюжету з окремих графічних образів для фрагмента інформаційного середовища.

Таким чином, весь ланцюжок перетворень, які реалізуються стегосистемою на рівні окремих етапів перетворення, може бути записаний у вигляді: $S \Rightarrow CH \Rightarrow GS \Rightarrow WS \Rightarrow SJ$, де SJ - опис сюжету, який складається з окремих графічних образів. Слід зазначити, що розглянутий підхід можна застосувати як для перетворення зображення з метою розміщення в ньому приховуваного повідомлення, так і для генерування візуального зображення по секретному повідомленню.

5.4 Приховування інформації у звуковому середовищі

Особливий розвиток методи цифрової стеганографії знайшли в аудіосередовищі. З їх допомогою забезпечується пересилання великих обсягів прихованих даних в звукових повідомленнях, які транслюються по телевізійній, радіо або телефонній мережах. Сучасні засоби телекомунікації дозволяють передавати звукові сигнали не тільки в реальному часі, а й у циф-

приховуванні 1 до кольору пікселя додається випадкове значення Δx. Хоча цей підхід подібний до методу заміни, є одна істотна відмінність: в методі LSB значення обраного кольору не обов'язково дорівнює секретному біту повідомлення, а в методах спотворення при приховуванні нульового біта не відбувається ніяких змін. Крім цього, значення Δx може бути вибрано так, що будуть зберігатися статистичні властивості контейнера.Для вилучення прихованих даних необхідно провести порівняння всіх $I(m)$ обраних пікселів стеганограми з відповідними пікселями вихідного контейнера. Якщоi-й піксель буде відрізнятися, то це свідчить про те, що в прихованому повідомленні був одиничний біт, інакше – нульовий.

Існує ще один підхід до реалізації методу спотворення зображення при приховуванні даних. Відповідно до даного методу, при вставці приховуваних даних робиться швидше спроба змінити порядок появи надлишкової інформації в контейнері, ніж змінити його вміст. При приховуванні даних складається певний "список пар" пікселів, для яких відмінність буде менше порогової. Цей список грає роль стегоключа - без нього не можна відновити секретне повідомлення. Якщо абонент має доступ до "списку пар", він завжди зможе провести зворотний процедуру.

Структурні методи

Розглянуті вище методи в основному використовували інформаційну надмірність на рівні пікселів або ж проводили перетворення в частотній області зображення. Нижче наводиться метод, в якому приховування інформації проводиться на змістовному рівні з використанням структурних та інформаційних параметрів зображення. По суті, він є розвитком відомої стеганографічної технології - семаграми. Суть методу полягає в проведенні послідовних перетворень фрагментів графічного зображення, які в кінці призводять до формування приховуваного тексту.

В даний час з'явилося безліч графічних пакетів програм і баз даних, за допомогою яких можна створювати різні графічні зображення, презентації, мультиплікації тощо. У кожному графічному зображенні можна виділити окремі компоненти, які відповідно до його області інтерпретації мають своє інформаційне навантаження. Візуальний образ S можна представити у вигляді цифрової послідовності, яка потім легко перетворюється в текстове повідомлення. Це можливо, наприклад, в процесі покриття образу деяким графом, використовуючи інформаційну інтерпретацію окремих його компонентів. При першому наближенні вершинами такого графа можуть слугувати окремі компоненти малюнка, а ребрами - їх сполуки. При кодуванні інформації, що приховується, отриманий граф можна перетворювати досить широким спектром відомих в теорії графів перетвореннями. В кінці такий граф може бути розмічено відповідно до певного алгоритму і представлено у вигляді його числового інваріанта. Найпростішим інваріантом є матриця суміжності графа (послідовність нумерації вершин). Можна використовувати

		приємств і організацій • Персоналізація засобів обчислювальної техніки • Впровадження ПК в офісну, фінансово-господарську і економічну діяльність • Поява на базі ПК систем локальної інформаційної комунікації	(1975), Белл, ЛаПадула (1975-1976) • Публікація в США стандарту DES (1977) • Інтенсивні теоретичні дослідження у сфері несиметричної криптографії (У.Диффи, М.Хеллман (1976), стандарт RSA Р.Райвест, А.Шамир, А.Адлеман (1978) • «Помаранчева книга» (1983) • MMS-модель (1984) • ГОСТ 28147-89
Третій	Кінець 1980-х – теперішній час	• Повна комп'ютеризація усіх сфер діяльності • Повсюдне використання ПК, у тому числі і як засіб інформаційної комунікації • Виникнення і стрімкий розвиток глобальної інформаційно-комп'ютерної інфраструктури (мережі Інтернет) • Виникнення і розвиток «Інформаційного» законодавства	• Подальший розвиток формальних моделей і технологій захисту інформації • Перехід на «захищеність» при розробці комерційних КС: ОС, СУБД • Поява спеціальної проблеми комп'ютерної безпеки – комп'ютерних вірусів (термін ввів Ф.Коен, 1984) • Розвиток національних і міжнародних стандартів захищеності КС • Широке впровадження криптографічних засобів захисту інформації для зберігання і передачі комп'ютерної інформації, в архітектуру КС, в процедури автентифікації (поява криптографічних протоколів) • Теоретичні дослідження і реалізація практичних систем забезпечення цілісності комп'ютерної інформації (поява стандартів і систем електронного цифрового підпису) • Поява «комп'ютерної» злочинності

Об’єкти захисту інформації

Захист інформації — сукупність методів і засобів, що забезпечують цілісність, конфіденційність і доступність інформації за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації.

Основними **об'єктами захисту інформації** є:

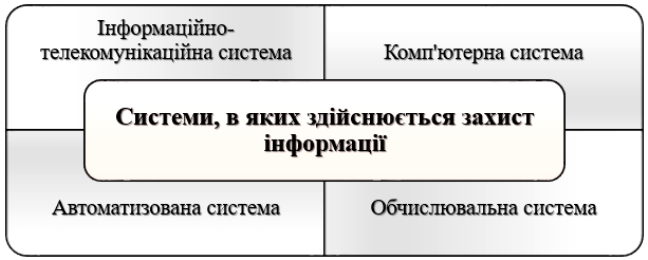
❒ *Інформація з обмеженим доступом* (ІзОД), тобто інформаційні ресурси, зокрема, ті, що містять відомості, як іналежат або до таємної, або до конфіденційної інформації;

❒ *Технічні засоби приймання, обробки, зберігання та передання інформації* (ТЗПІ), а саме: системи та засоби інформатизації (обчислювальна техніка, інформаційно-обчислювальні комплекси, мережі та системи); програмні засоби (операційні системи, системи керування базами даних та інше загально-системне і прикладне програмне забезпечення); автоматизовані системи керування; системи зв'язку; технічні засоби отримання, передавання та оброб-

ки ІзОД (звукозапис, звукопідсилення, звукосупроводження, переговорні та телевізійні пристрої;

❒ Засоби тиражування і виготовлення документів та інші технічні засоби обробки графічної, алфавітно-цифрової та текстової інформації), їх інформативні фізичні поля;

❒ *Допоміжні технічні засоби і системи* (ДТЗС), тобто технічні засобиі системи, якіне належать до ТЗПІ, але розташовані в приміщеннях, де обробляється ІзОД; до них відносять технічні засоби відкритого телефонного абогучномовного зв'язку, системи пожежної та охоронної сигналізації, система енергопостачання, радіотрансляційна мережа, система часофікації, енергопобутові прилади тощо, а також самі приміщення, де циркулює ІзОД.



Закон України **«Про захист інформації в інформаційно-телекомунікаційних системах»** визначає:

❒ **Інформаційна (автоматизована) система (ІС)** – організаційно-технічна система, в якій реалізується технологія обробки інформації з використанням технічних і програмних засобів.

❒ **Телекомунікаційна система (ТС)** – сукупність технічних і програмних засобів, призначених для обміну інформацією шляхом передавання,випромінювання або приймання її у вигляді сигналів, знаків, звуків, рухомих або нерухомих зображень чи в інший спосіб.

❒ **Інформаційно-телекомунікаційна система (ІТС)** – сукупність інформаційних та телекомунікаційних систем, які у процесі обробки інформації діють як єдине ціле.

Нормативний документ технічного захисту **інформації НД ТЗІ 1.1-003-99 «Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу»** визначає:

❒ **Обчислювальна система** — сукупність програмно-апаратних засобів, призначених для обробки інформації. Вона поєднує в собі технічні засоби оброблення і передавання даних (засоби обчислювальної техніки і зв'язку), а також методи і алгоритми оброблення даних, реалізовані у вигляді відповідного програмного забезпечення.

❒ **Комп'ютерна система (КС)** — сукупність програмно-апаратних засобів, яка подана для оцінки. Під *оцінюванням* розуміють експертне оціню-

Розглянемо приклад статистичного методу. Припустимо, що кожний блок контейнера B_i являє собою прямокутник пікселів $p^{(i)}{}_{n,m}$. Нехай є псевдовипадкова двійкова модель того ж розміру $S=\{S^{(i)}{}_{n,m}\}$, в якій кількість одиниць і нулів збігається. Модель S в даному випадку є стегоключем. Для приховування інформації кожен блок зображення B_i ділиться на дві рівні підмножини C_i і D_i, де $C_i=\{p^{(1)}{}_{n,m}\in B_i | S_{n,m}=1\}$ і $D_i=\{p^{(i)}{}_{n,m}\in B_i | S_{n,m}=0\}$. Потім до всіх пікселівмножини C_i додається значення $k>0$. Для вилучення повідомлення необхідно реконструювати підмножини C_i і D_i і знайти відмінність між ними. Якщо блок містить повідомлення, то всі значення підмножини C_i будуть більшими, ніж відповідні значення на етапі вбудовування повідомлення. Якщо припустити, що всі пікселі C_i і D_i незалежні, випадково розподілені величини, то можна застосувати статистичний тест:

$$q_i=\frac{\overline{C}_i-\overline{D}_i}{\hat{\sigma}_i}, \text{ де } \hat{\sigma}_i=\sqrt{\frac{Var[C_i]-Var[D_i]}{|S|/2}},$$

де $\overline{C}_i$ — середнє значення всіх пікселів множини C_i, а $Var[C_i]$ — оцінка дисперсії випадкових змінних в C_i. Відповідно до центральної граничної теореми, статистика q буде асимптотично прагнути до нормального розподілу $N(0,1)$. Якщо повідомлення вбудовано в блок зображення B_i, то математичне сподівання q буде більше нуля. Таким чином, і-й біт секретного повідомлення відновлюється шляхом перевірки статистики q_i блоку B_i на рівність нулю.

5.3.5 Структурні та методи спотворення інформації

Методи спотворення

Методи спотворення, на відміну від попередніх методів, вимагають знання про первісний вигляд контейнера. Схема приховування полягає в послідовному проведенні ряду модифікацій контейнера, які вибираються відповідно до секретного повідомлення. Для вилучення прихованих даних необхідно визначити всі відмінності між стеганограмою і вихідним контейнером. За цими розбіжностями відновлюється послідовність модифікацій, які виконувалися при приховуванні секретної інформації. Для більшості додатків такі системи не приносять користі, оскільки для вилучення даних необхідно мати доступ до набору первинних контейнерів: якщо противник також буде мати доступ до цього набору, то він зможе легко виявити модифікації контейнера і отримати докази прихованого листування. Таким чином, основною вимогою при використанні таких методів є необхідність поширення набору вихідних контейнерів між абонентами мережі через секретний канал доставки.

Методи спотворення легко застосовувати до цифрових зображень. Як і в методах заміни, для приховування даних вибирається $I(m)$ різних пікселів контейнера, які використовуються для приховування інформації. Такий вибір можна зробити, використовуючи датчик випадкових чисел (або перестановок). При приховуванні біта 0 значення пікселя не змінюється, а при

Якщо $m_i = 0$, то прихована інформація буде загублена. При деяких умовах значення $|\Delta C_i|$ може зрости настільки (хоча його математичне сподівання дорівнює нулю), що витяг відповідного біта стане неможливим.Однак це відбувається рідко, а можливі помилки можна виправляти, застосовуючи коригувальні коди.

Основна перевага широкосмугових стеганометодів - це порівняно висока стійкість до спотворення зображень і різного виду атак, так як прихована інформація розподілена в широкій смузі частот, отже її важко видалити без повного руйнування контейнера.Спотворення стегозображень збільшують значення ΔC_i і, якщо $|\Delta C_i| > |\Delta G_i m_i|$, то приховане повідомлення не постраждає.

5.3.4 Статистичні методи

Статистичні методи приховують інформацію шляхом зміни деяких статистичних властивостей зображення. Вони засновані на перевірці статистичних гіпотез. Суть методу полягає в такій зміні деяких статистичних характеристик контейнера, при якому одержувач зможе відрізнити модифіковане зображення від немодифікованого.

Дані методи відносяться до "однобітових" схем, тобто орієнтовані на приховування одного біта секретної інформації. $I(m)$-розрядна статистична стегосистема утворюється з безлічі однорозрядних шляхом розбиття зображення на $I(m)$ непересічних блоків $B_1,\ldots,B_{1(m)}$. При цьому секретний біт повідомлення m_i вбудовується в і-й блок контейнера. Виявлення захованого біта в блоці проводиться за допомогою функції перевірки, яка відрізняє модифікований блок від немодифікованого:

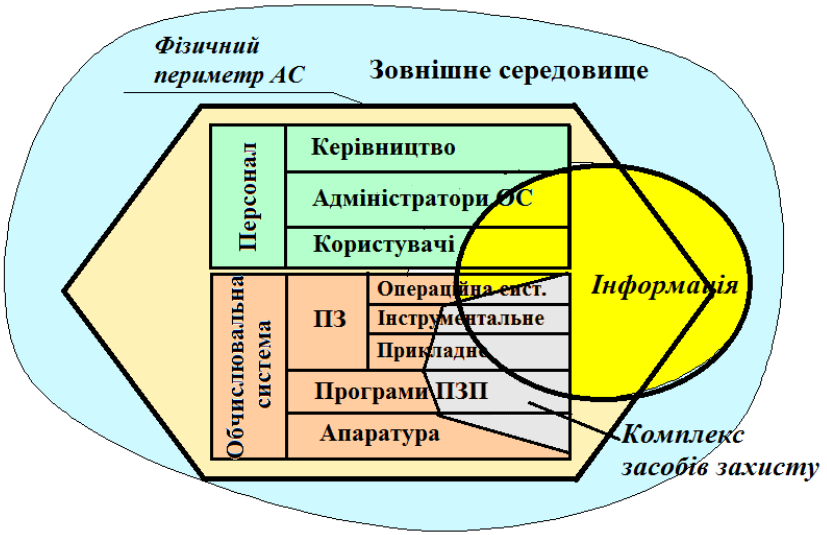
$$f(B_i) = \begin{cases} 1, & \text{якщо блок } B_i \text{ було модифіковано;} \\ 0, & \text{в іншому випадку.} \end{cases}$$

Основне завдання при розробці статистичного методу - це створення відповідної функції f. Побудова функції f робиться на основі теорії перевірки статистичних гіпотез (наприклад: основної гіпотези "блок B_i не змінений "та альтернативної -"блок B_i змінений "). Під час вилучення прихованої інформації необхідно послідовно застосовувати функцію f до всіх блоків контейнера B_i. Припустимо, що відома статистика розподілу елементів немодифікованого блоку зображення $h(B_i)$. Тоді, використовуючи стандартні процедури, можна перевірити, чи перевищує статистика аналізованого блоку $h(B_i)$ деяке порогове значення. Якщо не перевищує, то передбачається, що в блоці зберігається біт 0, в іншому випадку - 1.

Найчастіше статистичні методи стеганографії складно застосовувати на практиці. По-перше, необхідно мати хорошу статистику $h(B_i)$, на основі якої приймається рішення про те, чи є аналізований блок зображення зміненим чи ні. По-друге, розподіл $h(B_i)$ для "нормального" контейнера має бути заздалегідь відомим, що в більшості випадків є досить складним завданням.

вання захищеності інформації в системі, яке є складовою експертизи або сертифікації на відповідність чинним нормативним документам і стандартам.

❒ **Автоматизована система (АС)** — організаційно-технічна система, що реалізує інформаційну технологію і об'єднує обчислювальну систему, фізичне середовище, персонал та інформацію, яка обробляється.



Поняття комп'ютерної безпеки

Безпека — це такі умови, в яких перебуває складна система, коли дія зовнішніх факторів і внутрішніх чинників не призводить до процесів, що вважаються негативними по відношенню до даної складної системи у відповідності до наявних на даному етапі потреб, знань та уявлень.

Інформаційна безпека – стан захищеності життєве важливих інтересів людини, суспільства і держави, при якому запобігається нанесення шкоди че-

рез: неповноту, невчасність та невірогідність інформації, що використовується; негативний інформаційний вплив; негативні наслідки застосування інформаційних технологій; несанкціоноване поширення, використання, порушення цілісності, конфіденційності та доступності інформації.

Інформаційна безпека – це стан захищеності систем обробки і зберігання даних, при якому забезпечено конфіденційність, доступність і цілісність інформації, або комплекс заходів, спрямованих на забезпечення захищеності інформації від несанкціонованого доступу, використання, оприлюднення, руйнування, внесення змін, ознайомлення, перевірки, запису чи знищення (у цьому значенні частіше використовують термін «захист інформації»).

Комп'ютерна безпека – це сукупність проблем у галузі телекомунікацій та інформатики, пов'язаних з оцінкою і контролюванням ризиків, що виникають при користуванні комп'ютерами та комп'ютерними мережами і розглядуваних з точки зору конфіденційності, цілісності і доступності.

Закон України «Про основні засади забезпечення кібербезпеки України» дає таке визначення: «**Кібербезпека** — захищеність життєве важливих інтересів людини і громадянина, суспільства та держави під час використання кіберпростору, за якої забезпечуються сталий розвиток інформаційного суспільства та цифрового комунікативного середовища, своєчасне виявлення, запобігання і нейтралізація реальних і потенційних загроз національній безпеці України у кіберпросторі».

Забезпечення безпеки зводиться найчастіше до управління ризиком: визначення потенційних загроз, оцінка ймовірності їхнього настання та оцінка потенційної шкоди, із наступним ужиттям запобіжних заходів в обсязі, що враховує технічні можливості й економічні обставини.

Безпека інформації – це стан інформації, інформаційних ресурсів та інформаційних систем, при якому з необхідною ймовірністю забезпечується захист інформації від витоку, розкрадання, втрати, несанкціонованого доступу, знищення, модифікації (підробки), несанкціонованого копіювання, блокування інформації і т.п.

Безпека (захищеність) інформації в комп'ютері – це такий стан усіх компонент ПК, при якому забезпечується захист інформації від можливих загроз на необхідному рівні.

Загроза безпеки інформації в КС– це потенційно можлива подія, процес або явище, які можуть привести до знищення, втрати цілісності, конфіденційності або доступності інформації, порушення умов спостережливості та керованості КС.

Властивостями, що захищаються, є:

❒ *конфіденційністьінформації* – властивість інформації, що суб'єктивно встановлюється її власником, коли йому може бути нанесений збиток від ознайомлення з інформацією неуповноважених на те осіб (за умови того, що власник вживає заходи по організації доступу до інформації тільки уповноважених осіб);

- за допомогою змінних частот, коли частота сигналу змінюється за певним псевдовипадковим законом.

Розглянемо один з варіантів реалізації широкосмугового методу. В якості контейнера використовується напівтонове зображення розміром $N \times M$. Всі користувачі прихованого зв'язку мають безліч $I(m)$ зображень φ_i розміром $N \times M$, які використовуються в якості стегоключа. Зображення φ_i ортогональні один одному, тобто

$\varphi_i \varphi_j = \sum_{x=1}^{N} \sum_{y=1}^{M} \varphi_i(x,y) \varphi_j(x,y) = G_i \delta_{ij}$, де $G_i = \sum_{x=1}^{N} \sum_{y=1}^{M} \varphi_i^2(x,y)$, δ_{ij} - дельта-функція.

Для приховування повідомлення необхідно згенерувати стегоповідомлення $E(x,y)$ у вигляді зображення, формуючи зважену суму

$E(x,y) = \sum_i m_i \varphi_i(x,y)$.

Потім, шляхом формування елементної суми обох зображень, вбудувати секретну інформацію E в контейнер $C : S(x,y) = C(x,y) - E(x,y)$.

В ідеалі, контейнерне зображення C має бути ортогональним до всіх φ_i (тобто $\prec C, \varphi_i \succ = 0$), і одержувач може отримати і-й біт повідомлення m_i, проектуючи стегозображення S на базисне зображення φ_i:

$$< C, \varphi_i > = < C, \varphi_i > + < \sum_j m_j \varphi_j, \varphi_i > = \sum_j m_j < \varphi_j \varphi_i > = G_i m_i . \qquad (5.1)$$

Секретна інформація може бути вилучена шляхом обчислення $m_i = < C, \varphi_i > / G_i$. Зауважимо, що на цьому етапі немає потреби знати вихідний контейнер C. Однак, на практиці контейнер C не буде повністю ортогональним до всіх зображень φ_i, тому в співвідношення (5.1) повинна бути введена величина похибки $(C, \varphi_i) = \Delta C_i$, тобто $(C, \varphi_i) = \Delta C_i + G_i m_i$.

Покажемо, що при деяких припущеннях, математичне сподівання ΔC_i дорівнює нулю. Нехай C і φ_i - дві незалежні випадкові величини розміром $N \times M$. Якщо припустити, що всі базиси зображення не залежать від переданих повідомлень, то:

$\bar{E}[\Delta C_i] = \sum_{i=1}^{N} \sum_{j=1}^{M} \bar{E}[C(x,y)] \bar{E}[\varphi_i(x,y)] = 0$.

Таким чином, математичне сподівання величини похибки $< C, \varphi_i > = 0$. Тому операція декодування полягає у відновленні секретного повідомлення шляхом проектування стегозображення S на всі функції φ_i : $S_i = < S, \varphi_i > = \Delta C_i + G_i m_i$. Якщо математичне сподівання ΔC_i дорівнює нулю, то $S_i \approx G_i m_i$. Якщо секретні повідомлення були закодовані як рядки -1 і 1 (замість простого використання двійкових рядків), значення m_i можуть бути відновлені за допомогою функції:

$$m_i = sign(S_i) = \begin{cases} -1, & \text{при } S_i < 0 \\ 0, & \text{при } S_i = 0 \\ 1, & \text{при } S_i > 0 \end{cases}$$, за умови, що $G_i >> 0$.

повідомлення. Процес приховування починається з випадкового вибору блоку b_i, призначеного для кодування і-го біта повідомлення. Для обраного блоку зображення b_i проводиться ДКП: $B_i = D\{b_i\}$. При організації секретного каналу абоненти повинні попередньо домовитися про два конкретні коефіцієнти ДКП, які будуть використовуватися для приховування секретних даних. Позначимо їх як (u_1, v_1) та (u_2, v_2). Ці два коефіцієнти повинні відповідати косинус-функціям із середніми частотами, що забезпечить збереження інформації, яка не буде знищуватися при JPEG-стиску, в істотних областях сигналу. Так як коефіцієнти ДКП-середніх є подібними, то процес приховування не внесе помітних змін в зображення.

Якщо для блоку виконується умова $B_i(u_1, v_1) > B_i(u_2, v_2)$, то вважається, що блок кодує значення 1, в іншому випадку - 0. На етапі вбудовування інформації вибрані коефіцієнти змінюють значення між собою, якщо їх відносний розмір не відповідає кодованому біту. На етапі квантування JPEG-стиснення може впливати на відносні розміри коефіцієнтів, тому, додаючи випадкові значення до обох величин, алгоритм гарантує що $|B_i(u_1, v_1) - B_i(u_2, v_2)| > x$, де $x > 0$. Чим більший x, тим алгоритм буде більш стійким до стиснення, але при цьому якість зображення погіршується. Після відповідного корегування коефіцієнтів виконується зворотне ДКП.

Витяг прихованої інформації проводиться шляхом порівняння двох коефіцієнтів, вибраних для кожного блоку.

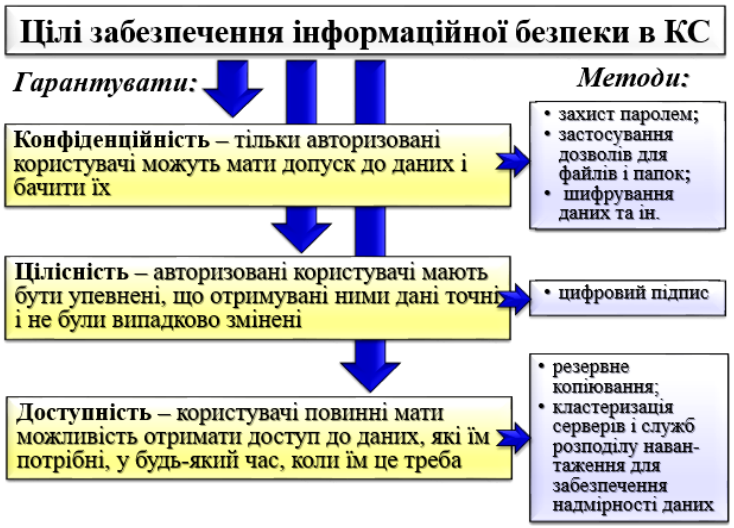
5.3.3 Широкосмугові методи

Широкосмугові методи передачі застосовуються в техніці зв'язку для забезпечення високої завадостійкості і ускладнення процесу перехоплення. Суть широкосмугових методів полягає в значному розширенні смуги частот сигналу, більш ніж це необхідно для передачі реальної інформації. Розширення діапазону виконується в основному за допомогою коду, який не залежить від переданих даних. Корисна інформація розподіляється по всьому діапазону, тому при втраті сигналу в деяких смугах частот в інших смугах присутньо достатньо інформації для її відновлення.

Таким чином, застосування широкосмугових методів в стеганографії ускладнює виявлення прихованих даних і їх видалення. Мета широкосмугових методів подібна завданням, які вирішує стегосистема: спробувати "розчинити" секретне повідомлення в контейнері і унеможливити його виявлення. Оскільки сигнали, розподілені по всій смузі спектра, важко видалити, стеганографічні методи, побудовані на основі широкосмугових методів, є стійкими до випадкових і навмисних спотворень.

Для приховування інформації застосовують два основних способи розширення спектра:

- за допомогою псевдовипадкової послідовності, коли секретний сигнал, що відрізняється на константу, модулюється псевдовипадковим сигналом;



❒ *цілісність*інформації – неспотвореність, достовірність, повнота, адекватність і т.д., тобто така властивість інформації, при якій її зміст і структура (даних) визначені уповноваженими особами і процесами;

❒ *доступність*інформації – така властивість інформації,при якій відсутні перешкоди доступу до інформації і закономірного її використання власником або уповноваженими особами;

❒ *спостереженість* – властивість комп'ютерної системи, що дозволяє фіксувати діяльність користувачів і процесів, використання пасивних об'єктів, а також однозначно установлювати ідентифікатори причетних до певних подій користувачів і процесів з метою запобігання порушення політики безпеки і/або забезпечення відповідальності за певні дії.

Загальні принципи забезпечення комп'ютерної безпеки:

❒ *Розумна достатність*– впровадженняв архітектуру, в алгоритми і технології функціонування КС захисних механізмів викликає додаткові витрати, витрати при створенні і експлуатації КС, обмежує, знижує функціональні можливості КС і параметри її ефективності (швидкодія, використані ресурси), викликає незручності в роботі користувачам КС, накладає на них додаткові навантаження і вимоги – тому захист має бути розумно достатнім (на мінімально необхідному рівні).

❒ *Цілеспрямованість*– усунення, нейтралізація (або забезпечення зниження потенційного збитку) конкретного переліку загроз (небезпек), характерних для конкретної КС в конкретних умовах її створення і експлуатації.

❒ *Системність* – вибір захисних механізмів з урахуванням системної суті КС, як організаційно-технологічної людино-машинної системи, що складається з взаємозв'язаних, таких, що становлять єдине ціле функціональних, програмних, технічних, організаційно-технологічних підсистем.

❒ *Комплексність* – вибір захисних механізмів різної і найбільш доцільної в конкретних умовах природи – програмно-алгоритмічних, процедурно-технологічних, нормативно-організаційних, криптографічних та ін., а також на всіх стадіях життєвого циклу (на етапах створення, експлуатації і виведення із ладу).

❒ *Безперервність* – захисні механізми повинні функціонувати у будь-яких ситуаціях, у тому числі, і позаштатних, забезпечуючи як конфіденційність, цілісність, так і збереження (правомірну доступність).

❒ *Керованість*–системазахисту КС будується як система управління – об'єкт управління (загрози безпеки і процедури функціонування КС), суб'єкт управління (засоби і механізми захисту), середовище функціонування, зворотний зв'язок в циклі управління, цільова функція управління (зниження ризику від загроз безпеки до необхідного (прийнятного) рівня), контроль ефективності (результативності) функціонування.

❒ *Поєднання уніфікації і оригінальності*– з одного боку з урахуванням досвіду створення і застосування КС, досвіду забезпечення безпеки КС повинні застосовуватися максимально перевірені, стандартизовані та уніфіковані архітектурні, програмно-алгоритмічні, організаційно-технологічні рішення, – зіншого боку, з урахуванням динаміки розвитку інформаційних технологій, діалектики засобів нападу і захисту повинні розроблятися і впроваджуватися нові оригінальні архітектурні, програмно-алгоритмічні, організаційно-технологічні рішення, що забезпечують безпеку КС в нових умовах загроз, з мінімізацією витрат, підвищенням ефективності і параметрів функціонування КС, зниженням вимог до користувачів.

відповідає двом елементам палітри, які вибираються відповідно до біту секретного повідомлення.

До методів заміни можна також віднести ***метод квантування зображень***. Даний метод заснований на міжпіксельній залежності, яку можна описати деякою функцією Q. У найпростішому випадку, можна розрахувати різницю між суміжними пікселями x_i та x_{i+1} і задати її в якості параметра для функції $Q: \Delta_i = Q(x_i - x_{i-1})$, де Δ_i — дискретна апроксимація різниці сигналів $x_i - x_{i-1}$. Так як Δ_i є цілим числом, а реальна різниця $x_i - x_{i-1}$— дійсним, то з'являється помилка квантування $\delta_i = \Delta_i - e_i$. Для сигналів, які сильно корелюються,ця помилка близька до нуля: $\delta_i \approx 0$. В даному методі приховування інформації проводиться шляхом коригування сигналу Δ_i. Стегоключ є таблицею, яка кожному можливому значенню Δ_i ставить у відповідність певний біт, наприклад:

Δ_i	–4	–3	–2	–1	0	1	2	3	4
	0	1	0	1	1	1	0	0	1

Для приховування і-го біта повідомлення обчислюється Δ_i. Якщо Δ_i не відповідає секретному біту, який необхідно приховати, то значення Δ_i замінюється найближчим Δ_i, для якого ця умова виконується. Витяг секретного повідомлення проводиться відповідно до різниці між Δ_i і стегоключем.

5.3.2 Методи приховування у частотній області зображення

Як уже зазначалося, стеганографічні методи заміни нестійкі до будь-яких спотворень, а застосування операції стиснення із втратами призводить до повного знищення всієї секретної інформації, прихованої методом НЗБ в зображенні. Більш стійкими до різних спотворень, у тому числі і стиснення, є методи, які використовують для приховування даних не тимчасову область, а частотну.

Існує кілька способів представлення зображення в частотній області. Наприклад, з використанням дискретного косинусного перетворення (ДКП), швидкого перетворення Фур'є або вейвлет-перетворення. Дані перетворення можуть застосовуватися як до всього зображення, так і до деяких його частин. При цифровій обробці зображення часто використовується двовимірна версія дискретного косинусного перетворення:

$$S(u,v) = \frac{2}{N} C(u)\, C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} S(x,y) \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi u(2y+1)}{2N}\right),$$

$$S(x,y) = \frac{2}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(u)\, C(v) S(u,v) \cos\left(\frac{\pi u(2x+1)}{2N}\right) \cos\left(\frac{\pi u(2y+1)}{2N}\right),$$

де $C(u) = 1/\sqrt{2}$, якщо $u = 0$ і $C(u) = 1$ в іншому випадку.

Один з найбільш популярних методів приховування секретної інформації в частотній області зображення заснований на відносній зміні величин коефіцієнтів ДКП. Для цього зображення розбивається на блоки розміром 8×8 пікселів. Кожен блок призначений для приховування одного біта секретного

При збільшенні $I(m)$ та $I(c) = const$ дана ймовірність прямує до одиниці. Для запобігання перетинів необхідно зберігати всі індекси використаних елементів j_i і перед приховуванням нового пікселя проводити перевірку його на повторюваність.

Ще один підхід в реалізації методу заміни (метод блочного приховування) полягає в наступному. Початкове зображення-контейнер розбивається на $I(m)$ непересічних блоків I_i довільної конфігурації і для кожного з них обчислюється біт парності $p(I_i)$:

$$p(I) = \sum_{j \in I} НЗБ(c_j) \bmod 2 .$$

У кожному блоці проводиться приховування одного секретного біта m_i. Якщо біт парності $p(I_i)$ блоку I_i не збігається з секретним бітом m_i, то відбувається інвертування одного з НЗБ блоку I_i, в результаті чого $p(I_i) = m_i$. Вибір блоку може проводитися випадково з використанням стегоключа. Хоча цей метод має таку ж стійкість до спотворень, як і всі попередні, він має ряд переваг. Перш за все, є можливість змінювати значення такого пікселя в блоці, для якого статистика контейнера зміниться мінімально. Крім того, вплив наслідків вбудовування секретних даних в контейнер можна зменшити за рахунок збільшення розміру блоку.

Методи заміни палітри. Для приховування даних можна також скористатися палітрою кольорів, яка присутня в форматі зображення.

Палітра з N кольорів визначається як список пар індексів (i, c_i), який визначає відповідність між індексом i і його вектором кольоровості c_i. У зображенні кожному пікселю присвоюється індекс в палітрі. Так як кольори в палітрі не завжди впорядковані, то приховану інформацію можна кодувати послідовністю зберігання кольорів у палітрі. Існує $N!$ різних способів перестановки N- колірної гами, що цілком достатньо для приховування невеликого повідомлення. Однак методи приховування, в основі яких лежить порядок формування палітри, також нестійкі: будь-яка атака, пов'язана зі змінами палітри, знищує секретне повідомлення.

Найчастіше сусідні кольори в палітрі не обов'язково схожі, тому деякі стеганометоди перед приховуванням даних проводять упорядкування палітри так, що суміжні кольори стають подібними.Наприклад, значення кольору може бути впорядковано за відстанню d в RGB-просторі, де $d = \sqrt{R^2 + G^2 + B^2}$. Так як органи зору людини більш чутливі до змін яскравості кольору, то набагато краще сортувати вміст палітри за значеннями яскравості сигналу. Після сортування палітри можна змінювати НЗБ індексів кольору без особливого спотворення зображення.

Деякі стеганометоди передбачають зменшення загальної кількості значень кольорів (до $N/2$) шляхом "розмивання" зображення. При цьому елементи палітри дублюються так, щоб значення кольорів для них розрізнялися несуттєво. В результаті кожне значення кольору розмитого зображення

1.1.2 Поняття і класифікація загроз

Що може спричинити порушення безпеки інформації та проти чого застосовують заходи захисту інформації

Загроза – сукупність умов і чинників, що визначають потенційну або реально існуючу небезпеку виникнення інциденту, який може привести до нанесення збитку виробу ІТ або його власникові.

Загроза безпеки інформації – сукупність умов і чинників, що створюють потенційну або реально існуючу небезпеку, пов'язану з витоком інформації, і/або несанкціонованими і/або неумисними діями на неї.

Загроза безпеки КС – сукупність умов і чинників, що визначають потенційну або реально існуючу небезпеку порушення конфіденційності, цілісності, (правомірної) доступності комп'ютерної інформації, спостереженості та керованості КС, і/або зниження надійності (безвідмовності і автентичності) реалізації функцій КС.

Загрози за природою походження

***Випадкові* (*об'єктивні*)**– виникають без умисного наміру:

- ❒ *відмови і збої апаратури*:
  - визначаються якістю і надійністю апаратури;
  - технічними рішеннями та іншими чинниками;
- ❒ *завади на лініях зв'язку від зовнішніх дій*:
  - правильність вибору місця (маршруту) прокладення;
  - технічних рішень по завадозахищеності;
  - електромагнітної обстановки;
- ❒ *помилки людини як ланки інформаційної системи*;

За місцем в системі:

- як джерела інформації;
- як оператора (введення/виведення даних);
- як обслуговуючого персоналу;
- як ланки прийняття рішень;

Інтенсивність - $2 \cdot 10^{-2} \ldots 4 \cdot 10^{-3}$

За типом:

- логічні (неправильні рішення);
- сенсорні (неправильне сприйняття);
- оперативні та моторні (неправильна реалізація або реакція);

- ❒ *схемні і системо-технічні помилки розробників*;
- ❒ *структурні, алгоритмічні і програмні помилки*:
  - спеціальні методи проектування і розробки;
  - спеціальні процедури тестування і відладки;
- ❒ *аварійні ситуації*:
  - по виходу з ладу електроживлення;
  - по стихійних лихах;
  - по виходу з ладу систем життєзабезпечення.

Умисні*(*суб'єктивні*)*загрози–викликанілюдиною або пов'язані з діями людини, визначаються так званим людським чинником (мотиви, категорії, можливості). До них відносяться:

❒ розвідка, використання з корисливою метою персоналу КС;

❒ несанкціонований доступ до інформації (порушення фізичної цілісності КС, режимів її функціонування, режимів функціонування систем життєзабезпечення);

❒ несанкціонована модифікація структур КС (впровадження заставних і підслуховуючих пристроїв, інших засобів технічної розвідки);

❒ підключення до каналів зв'язку, перехоплення передаванних даних, аналіз трафіку;

❒ розкриття атрибутів доступу в КС, порушення роботи систем захисту інформації;

❒ крадіжка носіїв інформації, несанкціоноване їх копіювання, читання «залишкової» інформації;

методика найбільш ефективна при використанні потокових контейнерів (відео).

Для контейнерів довільного доступу (зображень) може використовуватися метод псевдовипадкової перестановки. Загальний принцип даних методів полягає в заміні надлишкової, малозначущої частини зображення бітами секретного повідомлення. Для вилучення повідомлення необхідно знати місце, де була розміщена приховувана інформація.

Найбільш поширеним методом цього класу є метод заміни найменшого значущого біта (НЗБ).

Популярність методу НЗБ обумовлена його простотою і тим, що він дозволяє приховувати у відносно невеликих файлах досить великі обсяги інформації. Даний метод зазвичай працює з растровими зображеннями, які представлені в форматі без стиснення (наприклад, GIF і BMP). Основним його недоліком є сильна чутливість до найменших спотворень контейнера. Для ослаблення цієї чутливості часто застосовують завадостійке кодування.

Суть методу НЗБ полягає в заміні найменш значущих бітів пікселів зображення бітами секретного повідомлення. У найпростішому випадку проводиться заміна НЗБ всіх послідовно розташованих пікселів зображення. Однак, так як довжина секретного повідомлення зазвичай менша ніж кількість пікселів зображення, то після його вбудовування в контейнері будуть присутні дві області з різними статистичними властивостями (область, в якій найменш значущі біти були змінені, і область, в якій вони не змінювалися). Це може бути легко виявлено за допомогою статистичних тестів. Для створення еквівалентної зміни ймовірності всього контейнера секретне повідомлення зазвичай доповнюють випадковими бітами так, щоб його довжина в бітах дорівнювала кількості пікселів в оригінальному документі.

Інший підхід, ***метод випадкового інтервалу***, полягає у випадковому розподілі бітів секретного повідомлення по контейнеру, в результаті чого відстань між двома вбудованими бітами визначається псевдовипадково. Ця методика найбільш ефективна при використанні потокових контейнерів (відео).

Для контейнерів довільного доступу (зображень) може використовуватися ***метод псевдовипадкової перестановки***.

Його суть полягає в тому, що генератор псевдовипадкових чисел виробляє послідовність індексів $j_1, \ldots, j_{l(m)}$ та зберігає k - й біт повідомлення в пікселі з індексом j_k. Однак в цьому випадку один індекс може з'явитися в послідовності більш ніж один раз, тобто може статися "перетин" - спотворення вже вбудованого біта. Якщо число бітів повідомлення набагато менше розміру зображення, то ймовірність перетину незначна, і пошкоджені біти можуть бути відновлені за допомогою коригувальних кодів. Імовірність принаймні одного перетину оцінюється як

$$p \approx 1 - \exp\left(-\frac{I(m)[I(m)-1]}{2I(c)}\right),$$ за умови, що $I(m) << I(c)$.

Інформаційним відеопотокам, які складаються з послідовності окремих кадрів зображення, властива також надмірність, обумовлена інформаційною, технічною, тимчасовою і функціональною (смисловою) залежністю між кадрами.

Останнім часом створено достатню кількість методів приховування інформації в цифрових зображеннях і відео, що дозволяє провести їх систематизацію та виділити наступні групи:

- методи заміни в тимчасовій (просторовій) області;
- методи приховування в частотній області зображення;
- широкосмугові методи;
- статистичні методи;
- методи спотворення;
- структурні методи.

Розглянемо деякі особливості, які характерні для кожної з виділених груп стеганометодів.

5.3.1 Методи заміни

Загальний принцип даних методів полягає в заміні надлишкової, малозначущої частини зображення бітами секретного повідомлення. Для вилучення повідомлення необхідно знати місце, де була розміщена приховувана інформація.

Найбільш поширеним методом цього класу є ***метод заміни найменшого значущого біта*** (НЗБ).

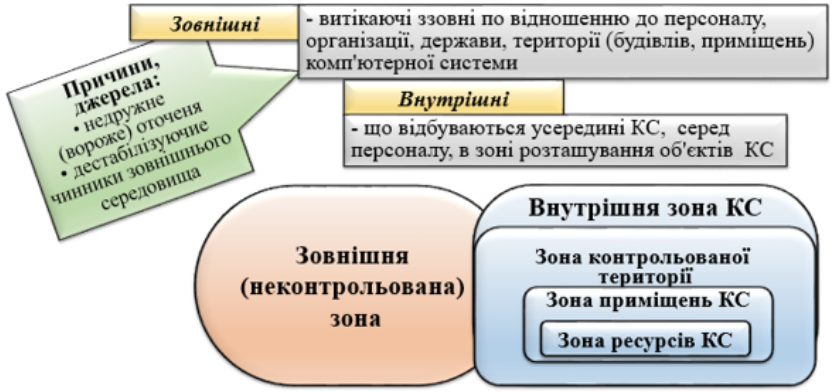
Популярність методу НЗБ обумовлена його простотою і тим, що він дозволяє приховувати у відносно невеликих файлах досить великі обсяги інформації. Даний метод зазвичай працює з растровими зображеннями, які представлені в форматі без стиснення (наприклад, GIF і BMP). Основним його недоліком є сильна чутливість до найменших спотворень контейнера. Для ослаблення цієї чутливості часто застосовують завадостійке кодування.

Суть методу НЗБ полягає в заміні найменш значущих бітів пікселів зображення бітами секретного повідомлення. У найпростішому випадку проводиться заміна НЗБ всіх послідовно розташованих пікселів зображення. Однак, так як довжина секретного повідомлення зазвичай менша ніж кількість пікселів зображення, то після його вбудовування в контейнері будуть присутні дві області з різними статистичними властивостями (область, в якій найменш значущі біти були змінені, і область, в якій вони не змінювалися). Це може бути легко виявлено за допомогою статистичних тестів. Для створення еквівалентної зміни ймовірності всього контейнера секретне повідомлення зазвичай доповнюють випадковими бітами так, щоб його довжина в бітах дорівнювала кількості пікселів в оригінальному документі.

Інший підхід, метод випадкового інтервалу, полягає у випадковому розподілі бітів секретного повідомлення по контейнеру, в результаті чого відстань між двома вбудованими бітами визначається псевдовипадково. Ця

- ❒ використання шкідливих програм (у тому числі, комп'ютерних вірусів, троянів, хробаків);
- ❒ впровадження і використання забороненого програмного забезпечення або несанкціоноване використання програм, які дозволяють отримати доступ до критичної інформації;
- ❒ використання засобів перехоплення побічних електромагнітних випромінювань і наведень, акустоелектричних перетворень небезпечних сигналів;
- ❒ використання радіочастотних засобів електромагнітної поразки напівпровідникової елементної бази КС.
- ❒

Загрози за напрямом здійснення



Несанкціонований доступ (НСД) – це доступ до інформації в КС з використанням засобів, включених до складу КС, що порушує встановлені правила розмежування доступу. НСД може здійснюватися:

- ❒ з використанням штатних засобів (сукупності програмно-апаратного забезпечення), що входять у затверджену конфігурацію КС;
- ❒ з використанням програмно-апаратних засобів, включених до складу КС порушником (наприклад, програмні або апаратні закладки).

Задачі НСД:

- ❒ безпосереднє звертання до об'єктів з метою одержання певного виду доступу;
- ❒ створення програмно-апаратних засобів, що виконують звертання до об'єктів в обхід засобів захисту;
- ❒ модифікація засобів захисту, що дозволяє здійснити НСД;
- ❒ впровадження в КС програмних або апаратних механізмів, що порушують структуру і функції КС і дозволяють здійснити НСД.

Класифікація загроз безпеки КС (можливостей впливу на КС, які прямо або непрямо завдають збитку її безпеки) за *аспектом інформаційної безпеки*, проти якої вони спрямовані:

❒ *загрози доступності* (відмова в обслуговуванні) – спрямовані на створення таких ситуацій, коли певні дії або блокують доступ до деяких ресурсів КС, або знижують її працездатність;

❒ *загрози цілісності інформації*, яка зберігається в КС або передається каналом зв'язку – спрямовані на зміну інформації або її спотворення, що призводить до порушення її якості або повного знищення;

❒ *загрози конфіденційності* – спрямовані на розголошування конфіденційної або таємної інформації. Інформація стає відомою особам, які не повинні мати до неї доступу. Загроза порушення конфіденційності має місце всякий раз, коли отриманий НСД до деякої закритої інформації, що зберігається в КС або передаваний від однієї системи до іншої;

❒ *загрози спостереженості та керованості КС* – спрямовані на порушення властивостей КС, які дозволяють фіксувати діяльність користувачів і процесів, використання пасивних об’єктів, а також однозначно установлювати ідентифікатори причетних до певних подій користувачів і процесів з метою запобігання порушення політики безпеки і/або забезпечення відповідальності за певні дії.

Технічні канали витоку інформації в комп’ютерних системах

Витік – це безконтрольний вихід конфіденційної (таємної) інформації за межі організації або кола осіб, яким вона довірена. Утворюється за рахунок неконтрольованих фізичних полів (акустичних, світлових, електро-магнітних, радіаційних, теплових та ін.)

Технічний канал витоку інформації – фізичний шлях від джерела інформації до порушника, за допомогою якого може бути здійснений НСД до відомостей, що охороняються. Технічні канали витоку підрозділяються на ві-

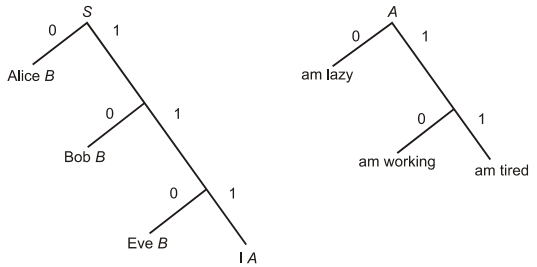


Рис. 5.5. Функція стиснення Хаффмана для $\ddot{I}_s$ та $\ddot{I}_A$

Розглянемо приклад. Нехай секретне повідомлення буде 11110. Тоді для зазначеної вище граматики Π на першому кроці перегляд дерева Π_s за допомогою трьох перших бітів повідомлення досягне листа I. Таким чином, початковий символ S буде замінений на $I\,A$. Потім, переглядаючи дерево ще раз, за допомогою наступних двох секретних бітів повідомлення станеться заміна наступних символів на am working. В результаті, кінцевий рядок буде складатися тільки з термінальних символів. В результаті стеганограмі 11110 відповідає повідомлення I am working.

Для вилучення прихованої інформації необхідно провести аналіз стеганограми з використанням дерева розбору КВГ. Так як граматика і продукції однозначні, то витяг прихованого повідомлення можливий.

Практичний досвід показав, що використання сучасних методів лінгвістичної стеганографії дозволяє створювати стеганограми, які важко виявити при автоматизованому моніторингу мереж телекомунікації, але обдурити з їх допомогою людину-цензора все ж дуже складно. У зв'язку з цим найбільший розвиток отримали стеганографічні методи захисту для інших інформаційних середовищ.

5.3 Приховування інформації в зображенні та відео

Розвиток мультимедійних засобів супроводжується великим потоком графічної інформації в обчислювальних мережах. При генерації зображення, як правило, використовується значна кількість елементарних графічних примітивів, що представляє особливий інтерес для стеганографічних методів захисту. Візуальне середовище (цифрові зображення і відео) має велику надмірність різної природи:

– кодову надмірність, що виникає при неоптимальному описі зображення;

– міжпіксельну надмірність, яка обумовлена наявністю сильної кореляційної залежності між пікселями реального зображення;

– псіховізуальну залежність, яка виникає через те, що органи зору людини не адаптовані для точного сприйняття зображення піксель за пікселем і сприймають кожну ділянку з різною чутливістю.

ними помилками. Для генерування більш осмислених текстів використовуються контекстно-вільні граматики (КВГ).

Контекстно-вільна граматика визначається впорядкованоючетвіркою $< V, \Sigma \subseteq V, \Pi, S \subset V \setminus \Sigma >$, де V та Σ – відповідно множини змінних і термінальних символів, Π – набір продукцій (правил виводу), а S – початковий символ. Продукції подібні до правил підстановки, вони перетворять змінну в рядок, що складається з термінальних або змінних символів. Якщо за допомогою правил виводу зі стартового символу можна отримати послідовність термінальних символів, то кажуть, що послідовність отримана граматикою. Такі граматики називаються контекстно-вільними, тому що будь-який символ можна замінити послідовністю символів, не звертаючи уваги на контекст, в якому він зустрівся. Якщо для кожного рядка s існує тільки один шлях, по якомуs може бути породженийіз початкового символу, то така граматика називається однозначною.

Однозначні граматики можуть використовуватися в якості апарату для стеганографічних перетворень. Розглянемо граматику

$< \{S, A, B, C\}, \{A, ..., Z, a, ..., z\}, \Pi, S >$,

де кожній можливій продукції приписана певна ймовірність:

$\Pi = \{ S \to_{0,5} Alice\ D, S \to_{0,3} Bob\ B, S \to_{0,1} Eve\ B, S \to_{0,1} I\ A;$

$A \to_{0,3} am\ working, A \to_{0,4} am lazi, A \to_{0,4} am\ tired; B \to_{0,5} is\ C, B \to_{0,5} can\ cook;$

$C \to_{0,5} reading, C \to_{0,1} sleeping, C \to_{0,4} working \}.$

Нехай $\Pi_{vi} = \{\pi_{i,1}, ..., \pi_{i,n}\}$ — набірвсіхпродукцій, якіпов'язанізізмінною V_i. Тоді для кожного набору Π_i можна створити функцію стиснення Хаффмана f_{Π_i}. На рис. 5.5 показані можливі дерева для Π_s та Π_A, з яких може бути легко отримана функція стиснення Хаффмана. Наприклад, продукція $Eve\ B$ кодуватиметься як 110, $I\ am\ tired$ — як 11 і т.д.

Для стеганографічних завдань використовується інверсійна функція Хаффмана. На етапі приховування даних відправник отримує за допомогою КВГ деякий рядок, який вважається стеганограмою. Стартуючи з початкового символу S, найлівіша змінна V_i замінюється по відповідній продукції. Ця продукція визначається відповідно до секретного повідомлення і функції стиснення Хаффмана для Π_{V_i} наступним чином. Відповідно до чергового біту секретного повідомлення відбувається перегляд дерева Хаффмана до тих пір, поки не буде досягнуто лист у дереві, після чого початковий символ замінюється на значення, яке приписане даному листу. Цей процес повторюється для всіх бітів повідомлення. Результуючий рядок складається тільки з термінальних символів.

зуально-оптичні, акустичні, електро-магнітні, матеріально-речові та ін.

Сигнали є *матеріальними носіями інформації*. По своїй фізичній природі сигнали можуть бути електричними, електромагнітними, акустичними і т.д., тобто сигналами, як правило, являються електромагнітні, механічні та інші види коливань (хвиль), причому інформація міститься в їх параметрах, що змінюються.

Електромагнітні канали витоку інформації. В електромагнітних каналах витоку інформації (ЕМКВІ) носієм небезпечної інформації є *електромагнітні випромінювання* (ЕМВ), що виникають при обробці інформації в КС.

У деяких ТЗОІ (наприклад, системах звукопідсилення) носієм інформації є електричний струм, параметри якого (сила струму, напруга, частота і фаза) змінюються за законом зміни інформаційного мовного сигналу. При протіканні електричного струму струмоведучими елементами ТЗОІ та їх сполучними лініями в просторі, що оточує їх, виникає змінне електричне і магнітне поле. Через це елементи ТЗОІ є випромінювачами електромагнітного поля, яке модулюється за законом зміни інформаційного сигналу.

Ініціаторами виникнення ПЕМВ можуть бути різного роду високочастотні генератори: задаючи генератори, генератори тактової частоти, генератори стирання і підмагнічування магнітофонів, гетеродини радіоприймальних і телевізійних пристроїв, генератори вимірювальних приладів і т.д.

Можливі режими роботи обчислювальної техніки, в яких виникають ПЕМВ:

- виведення інформації на екран монітора;
- введення даних з клавіатури;
- запис інформації на накопичувачі;
- читання інформації з накопичувачів;
- передача даних в каналах зв'язку;
- виведення даних на периферійні друкарські пристрої – принтері, плоттери;
- запис даних від сканерів на магнітний носій та ін.

Діапазон можливих частот ПЕМВ ЗОТ може складати 10 кГц – 2 ГГц.

Потенційні загрози і канали витоку в робочій станції

Людський чинник в загрозах безпеки і модель порушника

Порушник — фізична особа (необов'язково користувач системи), яка порушує політику безпеки системи.

Іноді використовують термін **зловмисник**, чим наголошують умисність здійсненого ним порушення, тоді як порушник може здійснювати порушення ненавмисно (наприклад, через необережність або недостатню обізнаність).

Роль людини в загрозах безпеки інформації:

- *носій/джерело загроз* (як внутрішніх, так і зовнішніх, як випадкових, так і умисних);
- *засіб, знаряддя здійснення загроз* (усіх умисних і певній частині випадкових загроз);

Нехай є файл A, який складається з символьних рядків. Позначимо через $p(t,a,A)$ ймовірність того, що символ a знаходиться в рядку t файлу A, а через $p(\cdot,a,A)$ та $p(t,\cdot,A)$ — незалежні ймовірності того, що символ a чи рядок t, відповідно, існують в A. Два файли A та B вважатимемо статистично еквівалентними в межах ε, якщо $[p(t,\cdot,A)-p(t,\cdot,B)]<\varepsilon$ для всіх рядків t, довжина котрих менше ніж n.

Функцією імітації n-го порядку будемо називати таку функцию f, яка в ε- околі виконує статистично еквівалентне перетворення файлу A у файл B.

Таким чином, якщо $p(t,A)$ — ймовірність появи деякого рядка t у файлі A, то фунція f перетворює файл A у файл B так, що для всіх рядків t довжиною менше n виконується співвідношення $[p(t,f(A))-p(t,B)]<\varepsilon$.

Можна запропонувати кілька типів функції імітації, які, в залежності від складності, моделюються регулярною, контекстно-вільною або рекурсивно-лічильною граматиками. Стеганографічні перетворення першого типу описуються в термінах процедур стиснення інформації; другого - контекстно-вільними граматиками, в яких приховувані біти керують несуперечливими одна одній продукціями; для опису функцій третього типу застосовується апарат машин Тьюринга.

Регулярні функції імітації можна змоделювати за допомогою схеми кодування по Хаффману. Відомо, що будь-яка мова має деякі статистичні властивості. Цей факт використовується багатьма методами стиснення даних. Якщо на алфавіті Σ задано розподіл ймовірностей A, то можна скористатися схемою кодування по Хаффману для створення функції стиснення з мінімальною надмірністю $f_A:\Sigma\to\{0,1\}^*$, де символ * використовується в сенсі $\Sigma^*=\cup_{i\geq 0}\{x_1\ldots x_i | x_1,\ldots,x_i\in\Sigma\}$. Таку функцію можна побудувати на основі функції стиснення Хаффмана: $G(x)=f_{B^{-1}}(f_A(x))$.

Таким чином, секретний файл можна стиснути за схемою Хаффмана з розподілом A, в результаті чого вийде файл двійкових рядків, які можуть інтерпретуватися як результат операції стиснення деякого файлу з розподілом B. Цей файл може бути відновлений з застосуванням інверсійної функції стиснення $f_{B^{-1}}$ до файлу двійкових рядків і використовуватися надалі як стеганограма. Якщо функції f_A та $f_{B^{-1}}$ є взаємно однозначними, то і створена функція імітації буде також взаємно однозначною. Доведено, що побудована таким чином функція подібності оптимальна в тому сенсі, що якщо функція стиснення Хаффмана f_A є теоретично оптимальною і файл x складається з випадкових біт, то взаємно однозначна функція $f_{A^{-1}}(X)$ має найкращу статистичну еквівалентність до A.

Регулярні функції імітації створюють стеганограми, які мають заданий статистичний розподіл символів, однак при цьому ігнорується семантика отриманого тексту. Для людини такі тексти видаються безглуздям з граматич-

Семантичні методи

Семантичні методи стеганографії аналогічні синтаксичним методам. Для цих методів елементарними лінгвістичними компонентами вважаються окремі слова, тому приховування даних реалізується шляхом безпосередньої заміни слів. Для такої заміни необхідні таблиці синонімів. Кодування секретного повідомлення проводиться вибором синоніма з необхідного місця таблиці. Наприклад, першому слову-синоніму відповідає 1, а другому - 0 (табл. 5.1). Якщо слову відповідає велика кількість синонімів, то можна кодувати більшу кількість біт одночасно.

Таблиця 5.1. **Фрагмент таблиці синонімів**

1	0
слід	відбиток
діра	отвір
оборона	захист
овації	оплески

На рис. 5.4 наведено приклад іншого підходу до приховування даних, в якому секретне повідомлення управляє перефразуванням тексту контейнера. В результаті виходить стеганограма, яка має той же зміст, що і текст контейнера.

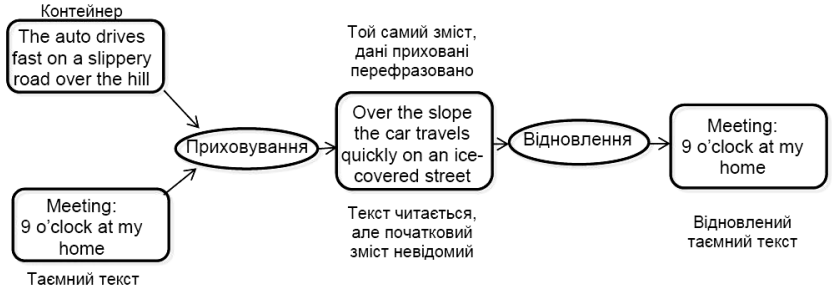


Рис. 5.4. Приклад роботи семантичної стегосистеми SubiText

5.2.3 Методи генерації стеганограм

На відміну від розглянутих вище стеганометодів, де прихована інформація впроваджується в текстовий контейнер, існують методи, які повністю породжують стеганограму на основі даних, що захищаються. В таких методах секретна інформація не впроваджується в текст, а представлена повністю всією стеганограмою. Теоретичну основу для методів генерації стеганограм розробив П. Вайнер в теорії функцій імітації. У стеганографії функції імітації застосовуються для того, щоб приховати ідентичність повідомлення шляхом зміни його статистичних властивостей.

❒ *предмет, об'єкт, середовище здійснення загроз* (як елементу людино-машинної КС).

Структура потенційних порушників (зловмисників)

Мотивидій, вчинків зі здійснення загроз:

❒ *усвідомлені*: ▪ користь, нажива; ▪ політика, влада, шпигунство; ▪ дослідницький інтерес;	❒ *неусвідомлені* (не цілком, не до кінця усвідомлювані): ▪ хуліганство; ▪ помста; ▪ заздрість; ▪ невдоволення; ▪ недбалість, несумлінність.

Модель порушника – сукупність представлень по людському чиннику здійснення загроз безпеки:

❒ категорії осіб, серед яких може виявитися порушник;

❒ його мотиваційні підстави і переслідувані цілі;

❒ його можливості по здійсненню тих або інших загроз (кваліфікація, технічна та інша інструментальна оснащеність);

❒ найбільш ймовірні способи його дій.

Модель порушника є основою для розробки і синтезу системи захисту інформації в КС.

Рівні можливостей порушників:

❒ *Перший* – запуск завдань (програм) з фіксованого набору, що реалізовують заздалегідь передбачені функції по обробці інформації.

❒ *Другий* – можливість створення і запуску власних програм з новими функціями по обробці інформації.

❒ *Третій* – можливість управління функціонуванням АС, тобто дією на базове програмне забезпечення системи і на склад і конфігурацію устаткування.

❒ *Четвертий* (найвищий) – увесь об'єм можливостей осіб, що здійснюють проектування, реалізацію і ремонт технічних засобів АС, аж до включення до складу засобів обчислювальної техніки власних технічних засобів з новими функціями по обробці інформації.

Основні способи НСД, здійснювані порушниками:

❒ безпосереднє звернення до об'єктів доступу;

❒ створення програмних і технічних засобів, що виконують звернення до об'єктів доступу в обхід засобів захисту;

❒ модифікація засобів захисту, що дозволяє здійснити НСД;

❒ впровадження в технічні засоби КС або АС програмних і технічних механізмів, що порушують передбачувану структуру і функції КС або АС і що дозволяють здійснити НСД.

1.1.3 Політика інформаційної безпеки

Мета захисту інформації – збереження цінності інформаційних ресурсів для їх власника.

Безпосередні заходи захисту спрямовують на збереження певних технологій створення інформаційних ресурсів, оброблення, зберігання, пошуку та надання користувачам. Ці технології мають ураховувати особливості інформації, які й роблять її цінною, а також давати змогу користувачам різних категорій працювати з інформаційними ресурсами (створювати, знаходити, копіювати, узагальнювати, порівнювати, модифікувати, перетворювати, знищувати тощо).

З цього випливає найважливіше для визначення мети захисту інформації поняття — **політика безпеки**.

Політика безпеки інформації(ПІБ) – сукупністьзаконів, правил, обмежень, нормативних документів, рекомендацій, інструкцій тощо, які регламентують порядок обробки інформації в КС і спрямованих на захист інформації від певних видів загроз.

Метою ПІБ має бути впровадження та *ефективне управління системою забезпечення інформаційної безпеки*, спрямованої на:

❒ захист інформаційних активів організації,

❒ забезпечення стабільної діяльності організації,

❒ мінімізації ризиків інформаційної безпеки,

❒ створення позитивних для організації інформаційних відносин з партнерами, клієнтами та всередині організації.

Основним завданням інформаційної безпеки є захист інформаційних активів від зовнішніх та внутрішніх навмисних та ненавмисних загроз.

ПІБ є фундаментальним документом по забезпеченню усього циклу без-

Наприклад, редактор TEX використовує складний алгоритм обчислення кінця рядка або сторінки. Фактично обчислюються якісь особливі налаштування, за якими визначається місце переходу з одного рядка (сторінки) на інший (у). Один з таких параметрів оцінює кількість пробілів, які необхідно вставити, щоб зберегти заданий стиль документа; інший – оцінює естетичний вигляд документа при виборі перенесення і т.д. В результаті, TEX намагається вибрати послідовність місць переносів таким чином, щоб сума всіх параметрів, які відносяться до редагованого параграфу, була мінімальною. Змінюючи деякі значення параметрів, можна управляти вибором місць переносів і використовувати їх для приховування даних.

До сих пір питання про створення безпечної лінгвістичної стегосистеми залишається відкритим. Будь-яка обробка тексту редактором, його друк або переведення в інший формат (HTML, PostScript, PDF або RTF) може змінити розташування пробілів і знищити прихований текст. Низька стійкість подібних методів до можливих модифікацій документа є однією з причин пошуку інших методів приховування даних в тексті.

Синтаксичні та семантичні методи абсолютно відрізняються від розглянутих вище, але можуть використовуватися одночасно з ними.

5.2.2 Синтаксичні та семантичні методи

Синтаксичні методи

До синтаксичних методів лінгвістичної стеганографії належать методи зміни пунктуації та методи зміни стилю і структури тексту.

У будь-якій мові існують випадки, коли правила пунктуації є неоднозначними і мають слабкий вплив на зміст тексту. Наприклад, обидві форми перерахування "хліб, масло і молоко" і "хліб, масло, молоко" є допустимими. Можна використовувати той факт, що вибір таких форм є довільним, і використовувати альтернативний вибір для кодування даних у двійковому вигляді. Наприклад, якщо з'являється форма перерахування з союзом "і", то кодується 1, інакше – 0. Для приховування можна також застосовувати скорочення і абревіатури.

У будь-якій мові є багато можливостей для синтаксичного приховування даних, але вони не часто зустрічаються в типових текстах. Середня швидкість передачі даних такими методами дорівнює кільком бітам на кілобайт тексту.

Хоча багато правил пунктуації є неоднозначними і надмірними, їх суперечливе використання може стати об'єктов уваги для цензора. Крім того, існують випадки, коли зміна пунктуації може сильно змінити зміст тексту. Тому такий підхід повинен використовуватися з обережністю.

До синтаксичних методів належать методи зміни стилю або структури тексту без істотної зміни його значення або тону. Наприклад, речення "До закінчення ночі я буду готовим" можна представити у вигляді "Я буду готовий швидше, ніж ніч закінчиться". Такий підхід більш прозорий, але можливість його обмежена.

бітове повідомлення 1000101101 скорочується до 001, а рядок 110011 - буде порожнім.

Розглянуті методи успішно працюють до тих пір, поки тексти представлені в коді ASCII. Існують також стеганографічні методи, які інтерпретують текст як бінарне зображення. У даних методах приховувана інформація кодується зміною відстані між послідовними рядками тексту або словами. Приховування даних відбувається шляхом вибору місця розташування рядків в документі, які зсуваються вгору або вниз відповідно до бітів приховуваних даних. При цьому деякі рядки залишають для синхронізації на місці (наприклад, кожен другий). У цьому випадку один секретний біт повідомлення кодується зрушенням одного рядка. Якщо рядок зрушено, то значення секретного біта дорівнює 1, інакше - 0.

Витяг прихованого повідомлення проводиться шляхом аналізу відстаней між центрами рядків, які розташовані поруч. Позначимо через Δ_{R+} - відстань між центрами зрушеного рядка і попередньої незміненого рядка (синхрорядка), Δ_{R-} — відстань між центрами зрушеної лінії і подальшого синхрорядка, а через Δ_{X+} та Δ_{X-} — відповідні відстані е вихідному документі. Тоді, якщо відстань між рядками було збільшено, то

$$\frac{\Delta_{R+}+\Delta_{R-}}{\Delta_{R+}-\Delta_{R-}} > \frac{\Delta_{X+}+\Delta_{X-}}{\Delta_{X+}-\Delta_{X-}}.$$

Аналогічно, якщо відстань було зменшено, то

$$\frac{\Delta_{R+}+\Delta_{R-}}{\Delta_{R+}-\Delta_{R-}} < \frac{\Delta_{X+}+\Delta_{X-}}{\Delta_{X+}-\Delta_{X-}}.$$

Відзначимо, що даний метод нечутливий до зміни масштабу документа, що забезпечує йому хорошу стійкість до більшості спотворень, які можуть мати місце при активних атаках.

Інша можлива схема приховування шляхом зсуву слів відформатованого тексту показана на рис. 5.3. Відповідно до цієї схеми змінюється горизонтальна позиція початку слів. Теоретично, можна використовувати зміну кожного проміжку між словами. Для того, щоб забезпечити збереження початкового вирівнювання тексту, необхідно дотримуватися єдиного обмеження: сума всіх зрушень в одному рядку повинна дорівнювати нулю.

Приклад приховування ‖ **даних** ‖ **в тексті**

Приклад приховування ‖ **даних** ‖ **в тексті**

Приклад приховування ‖ **даних** ‖ **в тексті**

Рис. 5.3. Приклад приховування даних в проміжках між словами (для наочності вказані вертикальні лінії)

Існують більш тонкі методи приховування інформації в текстовому середовищі. У деяких текстових редакторах реалізовані опції, які проводять автоматичне форматування тексту відповідно до визначених критеріїв.

пеки інформації в організації. Усі співробітники підрозділів, що відповідають за режим інформаційної безпеки організації, мають бути ознайомлені з політикою інформаційної безпеки під розпис. Адже на них ляже відповідальність за перевірку дотримання вимог ПІБ і знань основних її пунктів персоналом організації в частині, що їх торкається. Також має бути визначений процес проведення таких перевірок, обов'язки посадовців, що здійснюють такі перевірки, і розроблений графік перевірок.

ПІБ може бути розроблена як для окремого компонента ІС (КС, АС), так і для системи в цілому. Вона повинна враховувати наступні особливості інформаційної системи:

- ❒ технологію обробки інформації;
- ❒ обчислювальне середовище;
- ❒ фізичне середовище;
- ❒ середовище користувачів;
- ❒ правила розмежування доступу і т.д.

ПІБ повинна забезпечувати комплексне використання правових, морально-етичних норм, організаційних і технічних заходів, програмних, апаратних і програмно-апаратних засобів забезпечення інформаційної безпеки, а також визначати правила і порядок їх використання. ПІБ повинна базуватися на наступних *принципах*:

- ❒ безперервність захисту;
- ❒ достатність заходів і засобів захисту;
- ❒ їх відповідність ймовірності реалізації загроз;
- ❒ рентабельність, гнучкість структури;
- ❒ простота управління і використання і т.д.

Політика безпеки – це комплекс превентивних заходів по захисту конфіденційних даних та інформаційних процесів в організації. Політика безпеки включає вимоги на адресу персоналу, менеджерів і технічних служб. **Основні напрями розробки політики безпеки**:

- ❒ визначення які дані і наскільки серйозно необхідно захищати;
- ❒ визначення хто і який збиток може нанести організації в інформаційному аспекті;
- ❒ обчислення ризиків і визначення схеми зменшення їх до прийнятної величини.

ПІБ повинна містити пункти, в яких би була присутня інформація наступних розділів:

- ❒ концепція безпеки інформації;
- ❒ визначення компонентів і ресурсів інформаційної системи, які можуть стати джерелами порушення інформаційної безпеки і рівень їх критичності;
- ❒ зіставлення загроз з об'єктами захисту;
- ❒ оцінка ризиків;
- ❒ оцінка величини можливих збитків, пов'язаних з реалізацією загроз;

- ❒ оцінка витрат на побудову системи інформаційної безпеки;
- ❒ визначення вимог до методів і засобів забезпечення інформаційної безпеки;
- ❒ вибір основних рішень забезпечення інформаційної безпеки;
- ❒ організація проведення відновних робіт і забезпечення безперервного функціонування інформаційної системи;
- ❒ правила розмежування доступу.

Політика інформаційної безпеки дуже важлива для забезпечення комплексної безпеки організації. Програмно-апаратно її можна впроваджувати за допомогою DLP-рішень (*Data Leakage Prevention* – запобігання витокам даних).

1.2 Аналіз трафіку в мережі Internet

У мережі Internet основними базовими протоколами віддаленого доступу є TELNET і FTP (File Transfer Protocol). TELNET - це протокол віртуального терміналу (ВТ), що дозволяє підключатися до серверів Internet з віддалених хостів в режимі ВТ . FTP - протокол, призначений для передачі файлів між віддаленими хостами. Для отримання доступу до сервера за даними протоколів користувачеві необхідно виконати процедуру ідентифікації і аутентифікації. Інформацією, що ідентифікує користувача, виступає його ідентифікатор (ім'я), для аутентифікації використовується пароль. Особливістю протоколів FTP і TELNET є те, що паролі і ідентифікатори користувачів передаються по мережі у відкритому, незашифрованому вигляді.

Одним із способів отримання паролів і ідентифікаторів користувачів в мережі Internet є аналіз мережевого трафіку. Мережевий аналіз здійснюється за допомогою спеціальної програми-аналізатора пакетів, яка перехоплює всі пакети, що передаються по сегменту мережі, і виділяє серед них ті, у яких передаються ідентифікатор користувача і його пароль. Мережевий аналіз протоколів FTP і TELNET показує, що TELNET розбиває пароль на символи і пересилає їх по одному, поміщаючи кожен символ пароля у відповідний пакет, а FTP, навпаки, пересилає пароль повністю в одному пакеті.

1.2.1 Хибні сервери в мережі Internet

Хибний ARP-сервер в мережі Internet

Як уже неодноразово підкреслювалося в обчислювальних мережах, зв'язок між двома віддаленими хостами здійснюється шляхом передачі по мережі повідомлень, розміщених усередині пакетів обміну. У загальному випадку пакет, який передається по мережі, незалежно від використовуваного протоколу і типу мережі (Token Ring, Ethernet, X.25 та ін.), складається із заголовка пакета і поля даних.

Розглянемо схему адресації пакетів в мережі Internet. Відомо, що базовим мережевим протоколом обміну в мережі Internet є протокол IP (Internet Protocol). Протокол IP - це міжмережевий протокол, що дозволяє передавати

біту на 160 байт тексту). По-друге, можливість приховування залежить від структури тексту (деякі тексти, наприклад білі вірші, не мають чітких ознак кінця). По-третє, текстові редактори часто автоматично додають символи пробілу після точки.

Кодувати секретні дані можна додатковими пробілами в кінці кожного рядка тексту (рис. 5.1): два біта кодуються одним пропуском, чотири - двома, вісім - трьома і т.д. Перевага такого методу кодування полягає в тому, що воно може бути виконане з будь-яким текстом; зміни в форматі не надто помітні читачеві, забезпечується передача більшої кількості прихованих даних в порівнянні з попереднім методом (1 біт на 80 байт). Недолік методу полягає в тому, що деякі програми (наприклад, sendmail) можуть видаляти додаткові пробіли. Крім цього, приховані таким чином дані не завжди можуть бути відновлені з друкованої копії документа.

М	и		р	і	д	к	о		п	о	в	н	і	с	т	ю		р	о	-
з	у	м	і	є	м	о	,		ч	о	г	о		м	и		н	а	с	-
п	р	а	в	д	і		х	о	ч	е	м	о	.							

М	и		р	і	д	к	о		п	о	в	н	і	с	т	ю		р	о	-		
з	у	м	і	є	м	о	,		ч	о	г	о		м	и		н	а	с	-		
п	р	а	в	д	і		х	о	ч	е	м	о	.									

Рис. 5.1. Приклад приховування даних пробілами в кінці текстових рядків

Ще один метод приховування даних за допомогою пробілів маніпулює з текстами, вирівняними по обидва боки. У цьому методі дані кодуються шляхом керованого вибору місць для розміщення додаткових символів пробілу. Один символ між словами інтерпретується як 0, а два - як 1. Метод дозволяє вбудовувати кілька біт прихованої інформації в кожен рядок тексту (рис. 5.2).

Уникай_людей__,__які, _бачачи_твої_вади_і__недоліки, _виправдовують__їх_або _навіть__схвалюють. _Такі__люди__або_підлабузники, _або_боягузи, _або__ просто_дурні. __Від__них_не_чекай__допомоги__ні__в__якій__біді__чи_ нещасті __.__Г. _Сковорода

Рис. 5.2. Приклад приховування бітового повідомлення 0110≡100011010110

Оскільки текст часто вирівнюється по ширині листа, не кожен проміжок між словами може використовуватися для кодування прихованих даних. Для того щоб визначити, в якому з проміжків між словами захована інформація, а які проміжки є частиною оригінального тексту, використовується наступний метод декодування. Бітовий рядок, який витягується із стеганограми, розбивається на пари. Пара біт 01 інтерпретується як 1; пара 10 - як 0; а біти 00 і 11 є порожніми, тобто такими, які не несуть ніякої інформації. Наприклад,

		чих ознак вимагається корекція статистичних характеристик	браження товарної марки, реєстраційних номерів і т.п.

5.2 Текстові стеганограми

Сучасні стеганографічні засоби зазвичай працюють в інформаційних середовищах, які мають велику надмірність. На відміну від інформації, яка містить багато шумових даних (наприклад, звук і зображення), друкований текст містить малу кількість надлишкової інформації, яку можна використовувати для приховування даних.

Методи лінгвістичної стеганографії - приховування секретних повідомлень в тексті - відомі ще із середньовіччя. В основному такі методи використовують або природну надмірність мови, або формати представлення тексту. З розвитком комп'ютерних технологій середньовічні методи лінгвістичної стеганографії відродилися на якісно новому рівні і дозволяють в деяких випадках приховати факт таємного листування не тільки від "автоматичного цензора", який здійснює моніторинг мереж телекомунікацій, а й від людини.

Можна виділити наступні методи, які зустрічаються в сучасних лінгвістичних стеганографах:

— методи спотворення формату текстового документа;

— синтаксичні методи;

— семантичні методи;

— методи генерації стеганограм за допомогою приховуваного повідомлення.

5.2.1 Методи спотворення формату текстового документа

Приховування даних шляхом *зміни формату текстових файлів* зазвичай проводиться так, щоб стандартні текстові редактори не змогли виявити ознак присутності додаткової інформації. Розглянуті нижче методи маніпулюють інтервалами між словами і реченнями або ж пробілами в кінці текстових рядків. Використання пробілів для приховування даних обумовлено наступними причинами. По-перше, введення додаткових пробілів не вносить великих змін в значення фрази або пропозиції. По-друге, у випадкового читача навряд чи відразу виникне підозра щодо вставлених додаткових пробілів.

Приховування таємного повідомлення (в бітовому поданні) можна проводити шляхом додавання одного або двох символів пробілу в кінці речень після символу кінця (наприклад, точки - для натуральної мови або крапки з комою - для коду програми на мові С): один додатковий пробіл кодує значення біта "0", а два - "1". Цей простий метод має недоліки. По-перше, він не ефективний, тому що необхідний контейнер великого обсягу (швидкість передачі прихованих даних в даному випадку приблизно дорівнює одному

IP-пакети в будь-яку точку глобальної мережі. Для адресації на мережевому рівні (IP-рівні) в мережі Internet кожний хост має унікальну 32-розрядну IP-адресу. Для передачі IP-пакета на хост необхідно вказати в IP-заголовку пакета в полі Destination Address IP-адресу даного хоста.

Для адресації IP-пакетів в мережі Internet крім IP-адреси хоста необхідною є ще або Ethernet-адреса його мережевого адаптера (у випадку адресації всередині однієї підмережі), або Ethernet-адреса маршрутизатора (у випадку міжмережевої адресації). Спочатку хост може не містити інформації про Ethernet-адреси інших хостів, що знаходяться з ним в одному сегменті, у тому числі і про Ethernet-адресу маршрутизатора. Отже, перед хостом постає стандартна проблема, що вирішується за допомогою алгоритму віддаленого пошуку. У мережі Internet для розв'язання цієї проблеми використовується протокол ARP (Address Resolution Protocol). Протокол ARP дозволяє одержати взаємно однозначну відповідність IP- і Ethernet-адрес для хостів, що знаходяться всередині одного сегмента. Це здійснюється шляхом відправлення широкомовного ARP-запиту (на Ethernet-адресу FFFFFFFFFFFFh), який містить IP-адресу маршрутизатора і запит на сповіщення Ethernet-адреси.

Після перехоплення на атакуючому хості усередині даного сегмента мережі широкомовного ARP-запиту, можна надіслати хибну ARP-відповідь, у якій оголосити себе необхідним хостом (наприклад, маршрутизатором), і в подальшому активно контролювати та впливати па мережевий трафік "ошуканого" хоста за схемою "Хибний об'єкт РВС".

Для успішного проведення атак цього виду, необхідною умовою є знаходження сегмента мережі, що атакує всередині.

Хибний DNS-сервер в мережі Internet

Для вирішення проблеми перетворення імен в адреси створюється спеціальний файл (hosts file), в який вносяться імена і відповідні їм IP-адреси всіх хостів у мережі. Даний файл регулярно оновлюється і поширюється по всій мережі. Але із розвитком мережі Internet була створена нова система перетворення імен, що дозволяє користувачеві, у разі відсутності у нього інформації про відповідність імен і IP-адрес, отримати необхідні відомості від найближчого інформаційно-пошукового сервера (DNS-сервера). Ця система отримала назву доменної системи імен - DNS (Domain Name System).

Для реалізації системи DNS був створений спеціальний мережевий протокол DNS, для забезпечення ефективної роботи якого в мережі створюються окремі інформаційно-пошукові сервери - DNS-сервери.

DNS-сервер, отримавши запит, переглядає свою базу імен на наявність зазначеного в запиті імені. У випадку, якщо ім'я знайдено, а отже знайдена і відповідна йому IP-адреса, то на хост, із якого надійшов запит, DNS-сервер відправляє DNS-відповідь, у якій вказує шукану IP-адресу. У випадку, якщо зазначене в запиті ім'я DNS-сервер у своїй базі імен не виявив, то DNS-запит відсилається DNS-сервером на один з кореневих DNS-серверів, адреси яких містяться в файлі налаштувань.

З точки зору безпеки, вразливістю схеми віддаленого пошуку за допомогою протоколу DNS є можливість здійснення в мережі, що використовує протокол DNS, типової віддаленої атаки "Хибний об'єкт РВС".

1.2.2 Впровадження в мережу Internet хибних серверів, нав'язування та підтримка маршрутів

Впровадження в мережу Internet хибного DNS-сервера шляхом перехоплення DNS-запиту

Для реалізації атаки шляхом перехоплення DNS-запиту відбувається перехоплення DNS-запиту, з якого отримують номер UDP-порта відправника запиту, двухбайтовое значення ID ідентифікатора DNS-запиту і шукане ім'я, а потім надсилається хибна DNS-відповідь на витягнутий з DNS-запиту UDP-порт, у якому в якості шуканої IP-адреси вказується справжня IP-адреса хибного DNS-сервера. Це дозволить в подальшому повністю перехопити трафік між хостом, який було атаковано, і сервером, а також активно впливати на нього за схемою "Хибний об'єкт РВС".

Впровадження в мережу Internet хибного сервера шляхом створення спрямованого "шторму" помилкових DNS-відповідей на атакований хост

Для здійснення даної віддаленої атаки атакуючому необхідно вибрати хост, який його зацікавив, і змінити маршрут до нього так, щоб він проходив через хибний сервер - хост атакуючого. Це досягається постійною передачею (спрямованим "штормом") атакуючим хибних DNS-відповідей на хост від імені справжнього DNS-сервера на відповідні UDP-порти. У цих хибних DNS-відповідях у якості IP-адреси хоста вказується IP-адреса атакуючого. Далі атака розвивається за такою схемою. Як тільки мета атаки (хост, який було атаковано) звернеться по імені до хибного хоста, то від цього хоста в мережу буде переданий DNS-запит, який атакуючий ніколи не отримає, але цього йому і не потрібно, так як на хост відразу ж надійде постійна хибна DNS-відповідь, що і буде сприйнята ОС хоста, який було атаковано, як справжня відповідь від DNS-сервера.

Впровадження в мережу Internet хибного сервера шляхом перехоплення DNS-запиту або створення спрямованого "шторму" хибних DNS-відповідей на атакований сервер DNS

Спираючись на принципи віддаленого DNS-пошуку, слідує, що у випадку, коли зазначене в запиті ім'я DNS-сервер у своїй базі імен не виявив, запит відсилається сервером на один з кореневих DNS-серверів, адреси яких містяться в файлі налаштувань сервера root.cache.

І якщо у відповідь на запит від DNS-сервера атакуючий надішле хибну DNS-відповідь (або у разі "шторму" буде вести постійну передачу хибних відповідей), то в кеш-таблиці сервера з'явиться відповідний запис з неправдивими відомостями і в подальшому всі хости, які звернулися до даного DNS-сервера, будуть дезінформовані і при зверненні до хоста, маршрут до якого

2) Методи вибору певних позицій букв (нульовий шифр)	Акровірш - окремий випадок цього методу (наприклад, початкові букви кожного рядка утворюють повідомлення)	▪ низька ступінь скритності	
3) Методи використання спеціальних властивостей полів форматів, що не відображаються на екрані	Методи засновані на використанні спеціальних «невидимих», прихованих полів для організації виносок і посилань (наприклад, використання чорного шрифту на чорному фоні)		
1.3. Методи приховання в невживаних місцях дисків	Інформація записується в зазвичай невживаних місцях дисків	▪ слабка продуктивність методу ▪ передача невеликих об'ємів інформації ▪ низька ступінь скритності	▪ простота використання ▪ є опубліковане ПЗ реалізації методу
1.4. Методи використання імітуючих функцій (*mimic-function*)	Метод заснований на генерації текстів і є узагальненням акровірша. Для таємного повідомлення генерується осмислений текст, що приховує само повідомлення	▪ слабка продуктивність методу ▪ передача невеликих об'ємів інформації ▪ низька ступінь скритності	▪ результуючий текст не є підозрілим для систем моніторингу мережі
1.5. Методи видалення заголовка, що ідентифікує файл	Приховуване повідомлення шифрується і у результату віддаляється ідентифікуючий заголовок, залишаючи тільки шифровані дані. Одержувач заздалегідь знає про передачу повідомлення і має відсутній заголовок	▪ проблема приховання вирішується тільки частково ▪ необхідно заздалегідь передати частину інформації одержувачеві	▪ простота реалізації ▪ багато засобів (White Noise Storm, S-Tools та ін.) забезпечують реалізацію методу з PGP шифроалгоритмом
2. Використання надмірності аудіо і візуальній інформації			
2.1. Методи використання надмірності цифрових фотографій, цифрового звуку і цифрового відео	Молодші розряди цифрових відкліків містять дуже мало корисній інформації. Їх заповнення додатковою інформацією практично не впливає на якість сприйняття, що і дає можливість приховання конфіденційної інформації	▪ за рахунок введення додаткової інформації спотворюються статистичні характеристики цифрових потоків ▪ для зниження компрометую-	▪ можливість прихованої передачі великого об'-єму інформації ▪ можливість захисту авторського права, прихованого зо-

❒ *метод стрибаючих частот* – частота переносника повинна мінятися за певним псевдовипадковим законом.

Статистичний метод – метод приховання даних, при якому змінюються певні статистичні характеристики зображення, при цьому одержувач здатний розпізнати видозмінене зображення від вихідного.

Методи спотворення – методи приховання даних, при яких, залежно від секретного повідомлення, виконуються послідовні перетворення контейнера. У цьому методі важливо знать первинний вид контейнера. Знаючи відмінності між первинним контейнером і стеганограмой, можна відновити вихідну послідовність перетворень і витягнути приховані дані. При застосуванні цього методу важливо дотримувати правило: поширення набору первинних контейнерів здійснюється тільки через секретні канали доставки. У разі недотримання цього правила, порушник теж зможе оволодіти набором первинних контейнерів, що приведе до розтину таємного листування.

Структурний метод – метод приховання даних, при якому формується приховуваний текст, за допомогою здійснення послідовних модифікацій частин зображення. Цей метод дозволяє не лише модифікувати зображення, в якому буде приховано послання, але і створювати зображення по секретному повідомленню. Структурний метод дуже стійкий проти атак.

Таким чином, методи комп'ютерної стеганографії розвиваються по двох основних напрямах:

1) методи, засновані на використанні спеціальних властивостей комп'ютерних форматів;

2) методи, засновані на надмірності аудіо- і візуальній інформації.

Порівняльні характеристики комп'ютерних стеганографічних методів

Стеганографічні методи	Характеристика методів	Недоліки	Переваги
1. Використання спеціальних властивостей комп'ютерних форматів даних			
1.1. Методи використання зарезервованих для розширення полів комп'ютерних форматів даних	Поля розширення є в багатьох мультимедійних форматах, вони заповнюються нульовою інформацією і не враховуються програмою	▪ низька ступінь скритності ▪ передача невеликих обмежених об'ємів інформації	▪ простота використання
1.2. Методи спеціального форматування текстових файлів:			
1) Методи використання відомого зміщення слів, речень, абзаців	Методи засновані на зміні положення рядків і розставляння слів в реченні, що забезпечується вставкою додаткових пропусків між словами	▪ слабка продуктивність методу ▪ передача невеликих об'ємів інформації	▪ простота використання ▪ є опубліковане ПЗ реалізації методу

атакуючий вирішив змінити, зв'язок з ним буде здійснюватися через хост атакуючого за схемою "Хибний об'єкт РВС". І, що найгірше, з плином часу ця хибна інформація, що потрапила в кеш DNS-сервера, буде поширюватися на сусідні DNS-сервери вищих рівнів, а отже, все більше хостів в Internet будуть дезінформовані і атаковані.

Нав'язування хосту хибного маршруту з використанням протоколу ICMP з метою створення в мережі Internet хибного маршрутизатора

Для роботи в мережі Internet існує керуючий протокол ICMP, однією з функцій якого є віддалене управління маршрутизацією на хостах усередині сегмента мережі. Віддалене управління маршрутизацією необхідно для запобігання можливої передачі повідомлень по неоптимальному маршруту. У мережі Internet віддалене управління маршрутизацією реалізовано у вигляді передачі з маршрутизатора на хост керуючого ICMP-повідомлення: Redirect Message.

Для здійснення цієї віддаленої атаки необхідно підготувати хибне ICMP Redirect Host повідомлення, вказати в ньому кінцеву IP-адресу маршруту (адресу хоста, маршрут до якого буде змінений) і IP-адресу хибного маршрутизатора. Далі це повідомлення передається на хост, який було атаковано, від імені маршрутизатора. Для цього в IP-заголовку в полі адреси відправника вказується IP-адреса маршрутизатора. Можливе існування двох варіантів даної віддаленої атаки.

У першому випадку атакуючий знаходиться в тому ж сегменті мережі, що і мета атаки. Тоді, надіславши хибне ICMP-повідомлення, він замість IP-адреси нового маршрутизатора може вказати або свою IP-адресу, або будь-яку з адресу даної підмережі. Це дасть атакуючому можливість змінити маршрут передачі повідомлень, що направляються атакованим хостом на певну IP-адресу, і отримати контроль над трафіком між атакованим хостом і сервером, який цікавить атакуючого. Після цього атака перейде на другу стадію, пов'язану з отриманням, аналізом і передачею пакетів, одержуваних від "ошуканого" хоста.

У другому варіанті віддаленої атаки атакуючий знаходиться в іншому сегменті щодо мети атаки. Тоді, у разі передачі на атакований хост хибного ICMP Redirect повідомлення, сам атакуючий вже не зможе отримати контроль над трафіком, так як адреса нового маршрутизатора повинна знаходитися в межах підмережі атакованого хоста (див. реакцію мережевої ОС на ICMP Redirect повідомлення, описану вище в цьому пункті), тому використання даного варіанту цієї віддаленої атаки не дозволить атакуючому отримати доступ до інформації, що передається по каналу зв'язку інформації. Однак, в цьому випадку атака досягає іншої мети: порушується працездатність хоста. Атакуючий з будь-якого хоста в Internet може надіслати подібне повідомлення на атакований хост і в разі, якщо мережева ОС на даному хості не проігнорує дане повідомлення, то зв'язок між даним хостом і зазначеним у хибному

ICMP-повідомленні сервером буде порушено. Це станеться через те, що всі пакети, що направляються хостом на цей сервер, будуть відправлені на IP-адресу неіснуючого маршрутизатора.

Підміна одного із суб'єктів TCP-з'єднання в мережі Internet (hijacking)

Протокол TCP (Transmission Control Protocol) є одним із базових протоколів транспортного рівня мережі Internet. Цей протокол дозволяє виправляти помилки, які можуть виникнути в процесі передачі пакетів, і є протоколом зі встановленням логічного з'єднання - віртуального каналу. По цьому каналу передаються і приймаються пакети з реєстрацією їх послідовності, здійснюється управління потоком пакетів, організовується повторна передача спотворених пакетів, а в кінці сеансу канал розривається. При цьому протокол TCP є єдиним базовим протоколом із сімейства TCP/IP, що має додаткову систему ідентифікації повідомлень та з'єднань. Саме тому протоколи прикладного рівня FTP та TELNET, які надають користувачам віддалений доступ на хости Internet, реалізовані на базі протоколу TCP. Для ідентифікації TCP-пакета в TCP-заголовку існують два 32-розрядні ідентифікатори, які також відіграють роль лічильника пакетів. Їх назви - Sequence Number і Acknowledgment Number, а також поле, назване Control Bits.

При створенні TCP-з'єднання єдиними ідентифікаторами TCP-абонентів і TCP-з'єднання є два 32-бітні параметри Sequence Number і Acknowledgment Number. Отже, для формування хибного TCP-пакету атакуючому необхідно знати поточні ідентифікатори для даного з'єднання - ISSa і ISSb. Проблема можливої підміни TCP-повідомлення стає ще більш важливою, так як аналіз протоколів FTP і TELNET, реалізованих на базі протоколу TCP, показав, що проблема ідентифікації FTP- і TELNET-пакетів повністю покладається даними протоколами на транспортний рівень, тобто на TCP. Це означає, що атакуючому досить, підібравши відповідні поточні значення ідентифікаторів TCP-пакета для даного TCP-з'єднання (наприклад, дане з'єднання може являти собою FTP- або TELNET-підключення), надіслати пакет з будь-якого хоста в мережі Internet від імені одного з учасників даного з'єднання (наприклад, від імені клієнта), і даний пакет буде сприйнятий як вірний. До того ж, так як FTP і TELNET не перевіряють IP-адреси відправників, від яких їм надходять повідомлення, то у відповідь на отриманий помилковий пакет, FTP- або TELNET-сервер відправлять відповідь на вказану в хибному пакеті справжню IP-адресу атакуючого, тобто атакуючий почне роботу з FTP- або TELNET-сервером зі своєї IP-адреси, але з правами легально підключеного користувача, який, в свою чергу, втратить зв'язок з сервером через неузгодженості лічильників.

Таким чином, для здійснення описаної вище атаки необхідною і достатньою умовою є знання двох поточних 32-бітних параметрів ISSa і ISSb, що ідентифікують TCP-з'єднання.

Інші методи приховання інформації в графічних файлах орієнтовані на формати файлів з втратою, наприклад, JPEG. На відміну від LSB вони стійкіші до геометричних перетворень. Це виходить за рахунок варіювання в широкому діапазоні якості зображення, що призводить до неможливості визначення джерела зображення.

Ехо-методи застосовуються в цифровій аудіостеганографії і використовують нерівномірні проміжки між ехо-сигналами для кодування послідовності значень. При накладенні ряду обмежень дотримується умова непомітності для людського сприйняття. Ехо характеризується трьома параметрами:

- початковою амплітудою;
- ступенем загасання;
- затримкою.

Досягши деякого порогу між сигналом і ехо вони змішуються. У цій точці людське вухо не може вже відрізнити ці два сигнали. Наявність цієї точки складно визначити, і вона залежить від якості початкового запису та слухача. Найчастіше використовується затримка близько 1/1000. Для позначення логічного 0 і 1 використовується дві різні затримки. Вони мають бути менше, ніж поріг чутливості вуха слухача до отримуваної ехо.

Ехо-методи стійкі до амплітудних і частотних атак, але нестійкі до атак за часом.

Фазове кодування застосовується в цифровій аудіостеганографії. Відбувається *заміна початкового звукового елементу на відносну фазу, яка і є секретним повідомленням*. Фаза елементів, що йдуть підряд, має бути додана так, щоб зберегти відносну фазу між початковими елементами. Фазове кодування є одним з найефективніших методів приховання інформації.

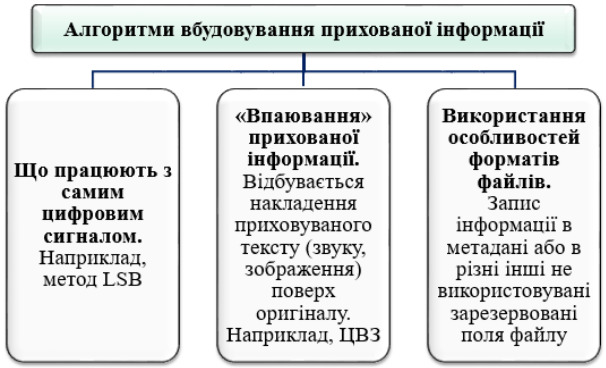
Метод розширеного спектру. Метод вбудовування повідомлення полягає в тому, що спеціальна випадкова послідовність вбудовується в контейнер, потім, з використанням узгодженого фільтру, ця послідовність детектується. Цей метод дозволяє вбудовувати велику кількість повідомлень в контейнер, і вони не створюватимуть перешкоди один одному за умови ортогональності вживаних послідовностей. Перевагою цього методу є протидія геометричним перетворенням, видаленню частини файлу і т.д.

Широкосмугові методи використовуються при передачі даних, забезпечуючи високу завадостійкість і перешкоджаючи процесам їх перехоплення. Відмітною особливістю є розширення діапазону частот сигналу за рахунок коду, на який не впливають передаванні дані. Необхідна інформація розосереджена по усій смузі частот і, у разі втрати сигналу, дані можуть бути відновлені з інших смуг частот. Подібний підхід до приховання сигналів значно ускладнює процес виявлення зашифрованої інформації, а також її видалення. Тому широкосмугові методи стійкі до будь-яких атак. Існує два основні методи розширення спектру:

- *метод псевдовипадкової послідовності* – використовується секретний сигнал, що модулюється псевдовипадковим сигналом;

ваної передачі або прихованого зберігання даних і методи для приховання даних в цифрових об'єктах з метою захисту самих цифрових об'єктів (наприклад, захист авторських прав).

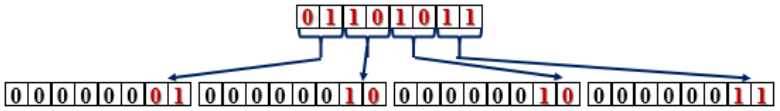
За ***типом інформаційного середовища*** виділяються стеганографічні методи для текстового або аудіо середовища, а також для зображень (стопкадрів) і відео середовища.



Метод LSB (*Least Significant Bit*, найменший значущий біт) – заміна останніх значущих бітів в контейнері (зображення, аудіо або відеозаписи) на біти приховуваного повідомлення. Різниця між порожнім і заповненим контейнерами має бути не відчуваєма для органів сприйняття людини.

Наприклад, є 8-бітове зображення в градаціях сірого: 00000000 означає ***чорний колір***, 11111111 – *білий*. Всього є 256 градацій (2^8). Повідомлення складається з 1 байта – 01101011.

При використанні 2 молодших біт в описах пікселів буде потрібно 4 пікселі (тобто 4 байта). Допустимо, вони чорного кольору. Тоді пікселі, що містять приховане повідомлення, виглядатимуть таким чином:



Колір пікселів зміниться: першого - на 1/255, другого і третього - на 2/255, четвертого - на 3/255. Такі градації, мало того, що непомітні для людини, можуть взагалі не відобразитися при використанні низькоякісних пристроїв виводу.

Методи LSB є *нестійкими до усіх видів атак* і можуть бути використані тільки за відсутності шуму в каналі передачі даних.

Виявлення LSB-кодованого стегоповідомлення здійснюється за аномальними характеристиками розподілу значень діапазону молодших бітів відліків цифрового сигналу.

1.3 Атаки на інформацію в комп'ютерних системах

Мережеві атаки

Мережеві атаки настільки ж різноманітні, як і системи, проти яких вони спрямовані. Деякі атаки відрізняються великою складністю. Інші може здійснити звичайний оператор, який навіть не припускає, які наслідки може мати його діяльність. Для оцінки типів атак необхідно знати деякі обмеження, безпосередньо властиві протоколу TPC/IP. Мережа Інтернет створювалася для зв'язку між державними установами та університетами для допомоги навчальному процесу та науковим дослідженням. Творці цієї мережі не підозрювали, наскільки широко вона поширіться. В результаті у специфікаціях ранніх версій Інтернет-протоколу (IP) були відсутні вимоги безпеки. Саме тому багато реалізації IP є уразливими від самого початку. Через багато років, отримавши безліч рекламацій (RFC - Request for Comments), нарешті стали впроваджувати засоби безпеки для IP. Однак з огляду на те, що спочатку засоби захисту для протоколу IP не розроблялися, всі його реалізації стали доповнюватися різноманітними мережевими процедурами, послугами і продуктами, що знижують ризики, властиві цим протоколом. Далі ми коротко обговоримо типи атак, які зазвичай застосовуються проти мереж.

Сніффери пакетів

Сніффер пакетів є прикладною програмою, яка використовує мережеву карту, що працює в режимі promiscuous mode (в цьому режимі всі пакети, отримані по фізичних каналах, мережевий адаптер відправляє додаткам для обробки). При цьому сніффер перехоплює всі мережеві пакети, які передаються через певний домен. В даний час сніфери працюють в мережах на цілком законній підставі. Вони використовуються для діагностики несправностей і аналізу трафіку. Однак, з огляду на те, що деякі мережеві додатки передають дані в текстовому форматі (Telnet, FTP, SMTP, POP3 і т.д.), за допомогою сніфферу можна дізнатися корисну, а іноді і конфіденційну інформацію (наприклад, імена користувачів і паролі). Перехоплення імен і паролів створює велику небезпеку, так як користувачі часто застосовують один і той же логін і пароль для безлічі додатків і систем. Багато користувачів взагалі мають один пароль для доступу до всіх ресурсів і додатків. Якщо додаток працює в режимі клієнт/сервер, а аутентифікаційні дані передаються по мережі в доступному для читання текстовому форматі, цю інформацію з великою ймовірністю можна використовувати для доступу до інших корпоративних або зовнішніх ресурсів. Хакери занадто добре знають і використовують наші людські слабкості (методи атак часто базуються на методах соціальної інженерії). Вони прекрасно знають, що ми користуємося одним і тим же паролем для доступу до безлічі ресурсів, і тому їм часто вдається, дізнавшись наш пароль, отримати доступ до важливої інформації. У найгіршому випадку хакер отримує доступ до призначеного для користувача ресурсу на системному рівні і з його допомогою створює нового користувача, якого можна в будь-який момент використову-

вати для доступу в мережу і до її ресурсів. Пом'якшити загрозу сніффінга пакетів можна за допомогою таких засобів:

- Аутентифікація.

Сильні сторони аутентифікації є першим способом захисту від сніффінга пакетів. Під «сильним» ми розуміємо такий метод аутентифікації, який важко обійти. Прикладом такої аутентифікації є одноразові паролі (OTP - One-Time Passwords). OTP - це технологія двофакторної аутентифікації, при якій відбувається поєднання того, що у вас є, з тим, що ви знаєте. Типовим прикладом двофакторної аутентифікації є робота звичайного банкомату, який розпізнає вас, по-перше, по вашій пластиковій картці і, по-друге, по ПІН-коду, що вами вводиться. Для аутентифікації в системі OTP також потрібен ПІН-код і ваша особиста картка. Під «карткою» (token) розуміється апаратний або програмний засіб, що генерує (за випадковим принципом) унікальний одномоментний одноразовий пароль. Якщо хакер дізнається цей пароль за допомогою сніффера, ця інформація буде марною, тому що в цей момент пароль вже буде використаний і виведений з ужитку. Зауважимо, що цей спосіб боротьби зі сніффінгом ефективний тільки для боротьби з перехопленням паролів. Сніффери, які перехоплюють іншу інформацію (наприклад, повідомлення електронної пошти), не втрачають своєї ефективності.

- Комутована інфраструктура.

Ще одним способом боротьби зі сніффінгом пакетів у вашому мережевому середовищі є створення комутованої інфраструктури. Якщо, наприклад, у всій організації використовується комутований Ethernet, хакери можуть отримати доступ тільки до трафіку, що надходить на той порт, до якого вони підключені. Комутована інфраструктура не ліквідує загрозу сніффінга, але помітно знижує її гостроту.

- Анти-сніфери.

Третій спосіб боротьби зі сніффінгом полягає в установці апаратних або програмних засобів, які розпізнають сніфери, що працюють у вашій мережі. Ці засоби не можуть повністю ліквідувати загрозу, але, як і багато інших засобів мережевої безпеки, вони включаються в загальну систему захисту. Так звані «анти-сніфери» вимірюють час реагування хостів і визначають, чи не доводиться хостам обробляти «зайвий» трафік. Один із таких засобів, що поставляються компанією LOpht Heavy Industries, називається AntiSniff. Більш детальну інформацію можна отримати на сайті www.10pht.com/antisniff/

- Криптографія.

Найефективніший спосіб боротьби зі сніффінгом пакетів не запобігає перехопленню і не розпізнає роботу сніфферів, але робить цю роботу марною. Якщо канал зв'язку є криптографічно захищенним, це означає, що хакер перехоплює не повідомлення, а зашифрований текст (тобто незрозумілу послідовність бітів). Криптографія Cisco на мережевому рівні базується на протоколі IPSec. IPSec є стандартним методом захищеного зв'язку між пристроями за допомогою протоколу IP. До інших криптографічних протоколів мережевого

мація, програми, графічні або звукові файли). В цьому випадку заздалегідь відомі розміри файлу і його вміст. Приховувані біти можуть бути рівномірно вибрані за допомогою відповідної псевдовипадкової функції. Недолік таких контейнерів полягає в тому, що вони мають набагато менші розміри, чим потокові, а також те, що відстані між приховуваними бітами рівномірно розподілені між найбільш коротким і найбільш довгим заданими відстанями, тоді як істинний шум матиме експоненціальний розподіл довжин інтервалу. Перевага подібних контейнерів полягає в тому, що вони можуть бути заздалегідь оцінені з точки зору ефективності вибраного стеганографічного перетворення.

За ***типом організації*** контейнери можуть бути систематичними і несистематичними.

❒ В**систематично** організованих контейнерах можна вказати конкретні місця стеганограми, де знаходяться інформаційні біти самого контейнера, а де – шумові біти, призначені для приховуваної інформації (як, наприклад, в широко поширеному методі найменшого значущого біта).

❒ При **несистематичній** організації контейнера такого розподілу зробити не можна. В цьому випадку для виділення прихованої інформації необхідно обробляти вміст усієї стеганограми.

За ***використовуваними принципами*** стеганометоди можна розбити на два класи: цифрові методи і структурні методи.

❒ **Цифрові методи** стеганографії, використовуючи надмірність інформаційного середовища, в основному, маніпулюють з цифровим представленням елементів середовища, куди впроваджуються приховувані дані (наприклад, в пікселі, в різні коефіцієнти косинус-косинусних перетворень, перетворень Фур'є, Уолша-Радемахера або Лапласа).

❒ **Структурні методи** стеганографії для приховання даних використовують семантично значущі структурні елементи інформаційного середовища.

Основним напрямом комп'ютерної стеганографії є використання властивостей *надмірності інформаційного середовища*. Слід врахувати, що при прихованні інформації відбувається спотворення деяких статистичних властивостей середовища або порушення його структури, які необхідно враховувати для зменшення демаскуючих ознак.

В особливу групу можна виділити методи, які використовують спеціальні властивості форматів представлення файлів:

❒ зарезервовані для розширення поля комп'ютерних форматів файлів, які зазвичай заповнюються нулями і не враховуються програмою;

❒ спеціальне форматування даних (зміщення слів, речень, абзаців або вибір певних позицій букв);

❒ використання незадіяних місць на магнітних носіях;

❒ видалення ідентифікуючих заголовків для файлу та ін.

В основному, для таких методів характерні низька ступінь скритності, низька пропускна спроможність і слабка продуктивність.

За ***призначенням*** розрізняють стеганографічні методи власне для прихо-

Кадр інформаційного середовища – це деяка його частина, виділена за певними ознаками, наприклад, семантичними характеристиками. Як кадр може бути вибраний деякий окремий малюнок, звуковий файл, веб-сторінка та ін.

За ***способом відбору контейнера*** розрізняють методи сурогатної, селективної і конструктивної стеганографії.

❒ В методах **сурогатної (безальтернативної) стеганографії** відсутня можливість вибору контейнера і для приховання повідомлення вибирається перший контейнер, що попався, часто не зовсім відповідний до вбудованого повідомлення. В цьому випадку, біти контейнера замінюються бітами приховуваного повідомлення так, щоб ця зміна не була помітною. Основний недолік цих методів – дозволяють приховувати лише незначну кількість даних.

❒ В методах **селективної стеганографії** передбачається, що заховане повідомлення повинне відтворювати спеціальні статистичні характеристики шуму контейнера. Для цього генерують велике число альтернативних контейнерів, щоб потім вибрати найбільш відповідний з них для конкретного повідомлення. Окремим випадком такого підходу є обчислення деякій хеш-функції для кожного контейнера. При цьому для приховання повідомлення вибирається той контейнер, хеш-функція якого співпадає зі значенням хеш-функції повідомлення (тобто стеганограмою є вибраний контейнер).

❒ В методах **стеганографії, що конструюють**, контейнер генерується самій стегосистемой. Тут може бути декілька варіантів реалізації. Так, наприклад, шум контейнера може моделюватися приховуваним повідомленням. Це реалізується за допомогою процедур, які не лише кодують приховуване повідомлення під шум, але і зберігають модель первинного шуму. У граничному випадку за моделлю шуму може будуватися ціле повідомлення. Прикладами можуть служити метод, реалізований в програмі **MandelSteg**, де як контейнер для вбудовування повідомлення генерується фрактал Мандельброта, або ж апарат функцій імітації (**mumic function**).

За ***способом доступу до приховуваної інформації*** розрізняють методи для потокових (безперервних) контейнерів і методи для контейнерів з довільним доступом (обмеженої довжини).

❒ Методи, що використовують **потокові контейнери**, працюють з потоками безперервних даних (наприклад, інтернет-телефонія). В цьому випадку приховувані біти необхідно в режимі реального часу включати в інформаційний потік. Про потоковий контейнер не можна заздалегідь сказати, коли він почнеться, коли закінчиться і наскільки тривалим він буде. Більше того, об'єктивно немає можливості дізнатися заздалегідь, якими будуть наступні шумові біти. Існує цілий ряд труднощів, які необхідно здолати кореспондентам при використанні потокових контейнерів. Найбільшу проблему при цьому складає синхронізація початку прихованого повідомлення.

❒ Методи, які використовуються для **контейнерів з довільним доступом**, призначені для роботи з файлами фіксованої довжини (текстова інфор-

управління відносяться протоколи SSH (Secure Shell) і SSL (Secure Socket Layer).

IP-спуфінг

IP-спуфінг відбувається, коли хакер, що знаходиться всередині корпорації або поза нею, видає себе за санкціонованого користувача. Це можна зробити двома способами. По-перше, хакер може скористатися IP-адресою, що знаходиться в межах діапазону санкціонованих IP-адрес, або авторизованою зовнішньою адресою, якій дозволяється доступ до певних мережевих ресурсів. Атаки IP-спуфінга часто є відправною точкою для інших атак. Класичний приклад - атака DoS, яка починається з чужої адреси, що приховує справжню особистість хакера. Зазвичай IP-спуфінг обмежується вставкою хибної інформації або шкідливих команд у звичайний потік даних, переданих між клієнтським і серверним додатком або по каналу зв'язку між однорангових пристроями. Для двостороннього зв'язку хакер повинен змінити всі таблиці маршрутизації, щоб направити трафік на помилкову IP-адресу. Деякі хакери, проте, навіть не намагаються отримати відповідь від додатків. Якщо головне завдання полягає в отриманні від системи важливого файлу, відповіді додатків не мають значення. Якщо ж хакеру вдається змінити таблиці маршрутизації і направити трафік на помилкову IP-адресу, хакер отримає всі пакети і зможе відповідати на них так, ніби він є санкціонованим користувачем. Загрозу спуфінгу можна послабити (але не усунути) за допомогою таких заходів:

- Контроль доступу

Найпростіший спосіб запобігання IP-спуфінгу полягає в правильному налаштуванні управління доступом. Щоб знизити ефективність IP-спуфінгу, налаштуйте контроль доступу на відсікання будь-якого трафіку, що надходить із зовнішньої мережі з вихідною адресою, який повинен розташовуватися всередині вашої мережі. Зауважимо, що це допомагає боротися з IP-спуфінгом, коли санкціонованими є тільки внутрішні адреси. Якщо санкціонованими є і деякі адреси зовнішньої мережі, даний метод стає неефективним.

- Фільтрація RFC 2827.

Ви можете припинити спроби спуфінгу чужих мереж користувачами вашої мережі (і стати добропорядним «мережевим громадянином»). Для цього необхідно відбраковувати будь-який вихідний трафік, початкова адреса якого не є однією з IP-адрес вашої організації. Цей тип фільтрації, відомий під назвою «RFC 2827», може виконувати і ваш провайдер (ISP). В результаті відбраковують весь трафік, який не має вихідної адреси, очікуваної на певному інтерфейсі. Наприклад, якщо ISP надає з'єднання з IP-адресою 15.1.1.0/24, він може налаштувати фільтр таким чином, щоб з даного інтерфейсу на маршрутизатор ISP допускався тільки трафік, що надходить з адреси 15.1.1.0/24. Зауважимо, що до тих пір, поки всі провайдери не впровадять цей тип фільтрації, її ефективність буде набагато нижче можливої. Крім того, чим далі від фільтрованих пристроїв, тим важче проводити точну фільтрацію. Так, наприклад, фільтрація RFC 2827 на рівні маршрутизатора доступу вимагає пропуску

всього трафіку з головної мережевої адреси (10.0.0.0/8), тоді як на рівні розподілу (у цій архітектурі) можна обмежити трафік більш точно (адреса - 10.1.5.0/24).

Найбільш ефективний метод боротьби з IP-спуфінгом той же, що і у випадку зі сніффінгом пакетів: необхідно зробити атаку абсолютно неефективною. IP-спуфінг може функціонувати тільки за умови, що аутентифікація відбувається на базі IP-адрес. Тому впровадження додаткових методів аутентифікації робить цей вид атак марними. Кращим видом додаткової аутентифікації є криптографічний. Якщо вона неможлива, хороші результати може дати двофакторная аутентифікація з використанням одноразових паролів.

Відмова в обслуговуванні (Denial of Service — DoS)

DoS, поза всяким сумнівом, є найбільш відомою формою хакерських атак. Крім того, проти атак такого типу найважче створити стовідсотковий захист. Навіть серед хакерів атаки DoS вважаються тривіальними, а їх застосування викликає зневажливі усмішки, тому що для організації DoS потрібно мінімум знань і умінь. Проте, саме простота реалізації і величезна шкода, якої завдають атаки, потребують пильної уваги адміністраторів, які відповідають за мережеву безпеку. Якщо ви хочете більше дізнатися про атаки DoS, вам слід розглянути їх найбільш відомі різновиди, а саме:

— TCP SYN Flood,

— Ping of на сайті http://www.cert.Org/techJ:ips/deniai of_service.html Death,

— Tribe Flood Network (TFN) и Tribe Flood Network 2000 (TFN2K) Trinco,

— Stacheldracht,

— Trinity.

Гарним джерелом інформації з питань безпеки є група екстреного реагування на комп'ютерні проблеми (CERT - Computer Emergency Response Team), що опублікувала чудову роботу по боротьбі з атаками DoS. Цю роботу можна знайти на сайті http://cert.org/tech_tips/denial_of_servase.html

Атаки DoS відрізняються від атак інших типів. Вони не націлені на отримання доступу до вашої мережі або на отримання з цієї мережі будь-якої інформації. Атака DoS робить вашу мережу недоступною для звичайного використання за рахунок превищення допустимих меж функціонування мережі, операційної системи або програми.

У разі використання деяких серверних додатків (таких як web-сервер або FTP-сервер), атаки DoS можуть полягати в тому, щоб перевантажити всі з'єднання, доступні для цих додатків, і тримати їх в завантаженому стані, не допускаючи обслуговування звичайних користувачів. В ході атак DoS можуть використовуватися звичайні Інтернет-протоколи, такі як TCP і ICMP (Internet Control Message Protocol). Більшість атак DoS спирається не на програмні помилки або проломи в системі безпеки, а на загальні слабкості системної

Семаграми – це таємні повідомлення, в яких шифрпозначеннями є будь-які символи, окрім букв і цифр. Ці повідомлення можуть бути передані, наприклад, в малюнку, що містить точки і тире для читання кодом Морзе.

У рамках **комп'ютерної стеганографії** розглядаються питання, пов'язані з прихованням інформації, яка зберігається на носіях або передається мережами телекомунікацій, з організацією прихованих каналів в комп'ютерних системах і мережах, а також з технологіями цифрових водяних знаків і відбитків пальців.

Класифікація методів приховання інформації у комп'ютерній стеганографії

Більшість методів комп'ютерної стеганографії базуються на **принципах**:

1. Файли, які не вимагають абсолютної точності (наприклад, файли із зображенням, звуковою інформацією і т.п.), можуть бути до певної міри видозмінені без втрати функціональності.

2. Відсутність спеціального інструментарію або нездатності органів чуття людини надійно розрізняти незначні зміни в таких вихідних файлах.

В основі базових підходів до реалізації методів комп'ютерної стеганографії лежить *виділення малозначимих фрагментів середовища* і заміна існуючої в них інформації на інформацію, яку передбачається захистити. Оскільки в комп'ютерній стеганографії розглядаються середовища, підтримувані засобами обчислювальної техніки, то усе інформаційне середовище, зрештою, може представлятися в цифровому вигляді. Таким чином, незначущі для кадру інформаційного середовища фрагменти відповідно до того або іншого алгоритму або методики замінюються (змішуються) на фрагменти приховуваної інформації.

6) *Атака на підставі відомого порожнього контейнера*. В даному випадку аналітик має можливість порівняти порожній і заповнений контейнери.

7) *Атака на підставі вибраного порожнього контейнера* – якщо порушникові відомий порожній контейнер, то порівнюючи його з передбачуваним стего можна встановити наявність стегоканала. Незважаючи на уявну простоту методу, існує теоретичне обгрунтування ефективності цього методу. Особливий інтерес представляє випадок, коли контейнер відомий з деякою погрішністю (таке можливе при додаванні до нього шуму).

8) *Атака на підставі відомої математичної моделі контейнера* або його частини – порушник визначає відмінність підозрілого послання від відомої йому моделі. Наприклад, нехай біти усередині відліку зображення корельовані. Тоді відсутність кореляції може служити сигналом про наявність прихованого повідомлення. При цьому завдання впроваджувального повідомлення полягає в тому, щоб не порушити статистичних закономірностей в контейнері.

5.1.3 Сучасні методи стеганографії

У сучасній стеганографії можна виділити два напрями розвитку: *технологічну* та *інформаційну*.

До методів **технологічної стеганографії** відносяться методи, засновані на використанні хімічних або фізичних властивостей різних матеріальних носіїв інформації.

Хімічні методи стеганографії зводяться майже виключно до застосування невидимого чорнила, до якого відносяться органічні рідини і симпатийовані хімікалії.

До **фізичних методів** можна віднести мікроточки, різного виду тайники і методи камуфляжу. Нині фізичні методи представляють інтерес в області дослідження різних носіїв інформації з метою запису на них даних, які б не виявлялися звичайними методами прочитування. З'явився цілий ряд нових технологій, які, базуючись на традиційній стеганографії, використовують останні досягнення мікроелектроніки (голограми, кінеграми).

До **інформаційної стеганографії** можна віднести методи *лінгвістичної* і *комп'ютерної* стеганографії.

Лінгвістичні методи стеганографії підрозділяються на дві основні категорії: умовний лист і семаграми.

Існують три види **умовного листа**: *жаргонний код*, *пустушковий шифр* і *геометрична система*.

У **жаргонному коді** зовні нешкідливе слово має абсолютно інше реальне значення, а текст складається так, щоб виглядати як можна безневинне і правдоподібно. При застосуванні **пустушкового шифру** в тексті мають значення лише деякі певні букви або слова. Третім видом умовного листа є **геометрична форма**. При її застосуванні слова, що мають значення, розташовуються на сторінці в певних місцях або в точках перетину геометричної фігури заданого розміру.

архітектури. Деякі атаки зводять до нуля продуктивність мережі, перевантажуючи її небажаними і непотрібними пакетами або повідомляючи хибну інформацію про поточний стан мережевих ресурсів. Запобігти цьому типу атак важко, так як для цього потрібна координація дій з провайдером. Якщо трафік, призначений для перевантаження вашої мережі, не зупинити у провайдера, то при вході в мережу ви це зробити вже не зможете, тому що вся смуга пропускання буде зайнята. Коли атака цього типу проводиться одночасно через безліч пристроїв, ми говоримо про розподілену атаку DoS (DDoS - distributed DoS).

Загрозу атак типу DoS можна послабити трьома способами:

- Функції анти-спуфінга.

Правильна конфігурація функцій анти-спуфінга на ваших маршрутизаторах і міжмережевих екранах допоможе знизити ризик DoS. Ці функції, як мінімум, повинні включати фільтрацію RFC 2827. Якщо хакер не зможе замаскувати свою справжню особистість, він навряд чи зважиться провести атаку.

- Функції анти-DoS.

Правильна конфігурація функцій анти-DoS на маршрутизаторах і міжмережевих екранах може знизити ефективність атак. Ці функції часто обмежують число напіввідкритих каналів в будь-який момент часу.

- Обмеження обсягу трафіку (traffic rate limiting). Організація може попросити провайдера (ISP) обмежити об'єм трафіку. Цей тип фільтрації дозволяє обмежити обсяг некритичного трафіку, що проходить по вашій мережі. Типовим прикладом є обмеження обсягів трафіку ICMP, який використовується тільки для діагностичних цілей. Атаки (D)DoS часто використовують ICMP.

Парольні атаки

Хакери можуть проводити парольні атаки за допомогою цілого ряду методів, таких як простий перебір (brute force attack), «троянський кінь», IP-спуфінг і сніффінг пакетів. Хоча логін і пароль часто можна отримати за допомогою IP-спуфінга і сніффінга пакетів, хакери часто намагаються підібрати пароль і логін, використовуючи для цього багаторазові спроби доступу. Такий підхід носить назву простого перебору (brute force attack). Часто для такої атаки використовується спеціальна програма, яка намагається отримати доступ до ресурсу загального користування (наприклад, до сервера). Якщо в результаті хакер отримує доступ до ресурсів, він отримує його на правах звичайного користувача, пароль якого був підібраний. Якщо цей користувач має значні привілеї доступу, хакер може створити для себе «прохід» для майбутнього доступу, який буде діяти, навіть якщо користувач змінить свій пароль і логін.

Ще одна проблема виникає, коли користувачі застосовують один і той же (нехай навіть дуже хороший) пароль для доступу до багатьох систем: корпоративної, персональної і системи Інтернет. Оскільки стійкість пароля дорівнює стійкості найбільш слабкого хоста, хакер, що довідався пароль через цей

хост, отримує доступ до всіх інших систем, де використовується той же пароль. Перш за все, парольних атак можна уникнути, якщо не користуватися паролями в текстовій формі. Одноразові паролі і/або криптографічна аутентифікація можуть практично звести нанівець загрозу таких атак. На жаль, не всі програми, хости і пристрої підтримують зазначені вище методи аутентифікації. При використанні звичайних паролів намагайтеся придумати такий пароль, який було б важко підібрати. Мінімальна довжина пароля повинна бути не менше восьми символів. Пароль повинен включати символи верхнього регістру, цифри та спеціальні символи (#,%. $ і т.д.). Найкращі паролі важко підібрати і важко запам'ятати, що змушує користувачів записувати паролі на папері. Щоб уникнути цього, користувачі і адміністратори можуть скористатися одним з останніх технологічних досягнень. Так, наприклад, існують прикладні програми, що шифрують список паролів, який можна зберігати в кишеньковому комп'ютері. В результаті користувачеві потрібно пам'ятати тільки один складний пароль, тоді як всі інші паролі будуть надійно захищені додатком. З точки зору адміністратора, існує кілька методів боротьби з підбором паролів. Один з них полягає у використанні середовища LOphtCrack, яке часто застосовують хакери для підбору паролів у Windows NT. Це середовище швидко покаже вам, чи легко підібрати пароль, вибраний користувачем.

Додаткову інформацію можна отримати за посиланням www.10phtcrack.com.

Атаки типу Man-in-the-Middle

Для атаки типу Man-in-the-Middle хакеру потрібен доступ до пакетів, що передаються по мережі. Такий доступ до всіх пакетів, що передаються від провайдера в будь-яку іншу мережу, може, наприклад, отримати співробітник цього провайдера. Для атак цього типу часто використовуються сніфери пакетів, транспортні протоколи і протоколи маршрутизації. Атаки проводяться з метою крадіжки інформації, перехоплення поточної сесії і отримання доступу до приватних мережевих ресурсів, для аналізу трафіку і отримання інформації про мережу та її користувачів, для проведення атак типу DoS, спотворення переданих даних і введення несанкціонованої інформації в мережеві сесії.

Ефективно боротися з атаками типу Man-in-the-Middle можна тільки за допомогою криптографії. Якщо хакер перехопить дані зашифрованої сесії, у нього на екрані з'явиться не перехоплене повідомлення, а безглуздий набір символів. Зауважимо, що якщо хакер отримає інформацію про криптографічну сесію (наприклад, ключ сесії), це може зробити можливою атаку Man-in-the-Middle навіть у зашифрованому середовищі.

Атаки на рівні додатків

Атаки на рівні додатків можуть виконуватися кількома способами. Найпоширеніший з них полягає у використанні добре відомих слабкостей серверного програмного забезпечення (sendmail, HTTP, FTP). Використовуючи ці

Класифікація атак на стегосистеми

Методи стегоаналізу переважно базуються на різного роду статистичних критеріях, які дозволяють виявити деякі неоднорідності, залежності (кореляції) та нерівномірності у послідовностях символів можливо модифікованих контейнерів, що обумовлені принципами побудови системи та статистичними властивостями відкритих повідомлень.

1) *Атака на підставі відомого заповненого контейнера* (*суб'єктивна атака*) – аналітик досліджує контейнер без допомоги спеціальних засобів, намагаючись «на око» визначити, чи містить той стего. Тобто, якщо контейнер є зображенням, то дивиться на нього, якщо аудіозапис, то слухає. Не дивлячись на те, що подібна атака ефективна тільки проти майже не захищених стеганографічних систем, атака широко поширена на початковому етапі розкриття системи.

2) *Атака на підставі відомого вбудованого повідомлення* – у порушника є одне або декілька стего. У разі декількох стего вважається, що запис прихованої інформації проводився відправником однаковим способом. Завдання порушника полягає у виявленні факту наявності стегоканалу, а також доступу до нього або визначення ключа. Маючи ключ, можна розкрити інші стегоповідомлення.

3) *Атака на підставі вибраного вбудованого повідомлення*. Використовується у разі, коли аналітик може вибрати повідомлення і аналізувати відправлені заповнені контейнери.

4) *Адаптивна атака на підставі вибраного вбудованого повідомлення*. Окремий випадок атаки на основі вибраного прихованого повідомлення, коли аналітик має можливість вибирати повідомлення, виходячи з результатів аналізу попередніх контейнерів.

5) *Атака на підставі вибраного заповненого контейнера*.

5.1.2 Атаки на стегосистеми

Атака на стегосистему (чи **стегоаналіз**) – це спроба виявити, витягнути, змінити приховане стегоповідомлення.

Здатність стегосистеми протистояти атакам називається *стеганографічною стійкістю*.

Порушник (аналітик) прагне зламати стеганографічну систему, тобто виявити факт передачі повідомлення, витягнути повідомлення або модифікувати повідомлення, або заборонити пересилку повідомлення. Зазвичай аналітики проводять декілька етапів злому системи:

1) виявлення факту наявності прихованого повідомлення;
2) витягання повідомлення;
3) модифікація повідомлення;
4) заборона на виконання пересилки повідомлення.

Система вважається *зламаною*, якщо аналітикові вдалося довести хоч би наявність прихованого повідомлення.

В ході перших двох етапів аналітики зазвичай можуть проводити такі заходи:

- ❒ суб'єктивна атака;
- ❒ сортування стего за зовнішніми ознаками;
- ❒ визначення використаних алгоритмів вбудовування повідомлень;
- ❒ виділення повідомлень з відомим алгоритмом вбудовування;
- ❒ перевірка достатності об'єму матеріалу для аналізу;
- ❒ перевірка можливості аналізу за окремими випадками;
- ❒ аналіз матеріалів і розробка методів розкриття системи.

Виділяють декілька **видів порушників (стегоаналітиків)**:

1. *Пасивний порушник*, здатний тільки виявити факт пересилки повідомлення і, можливо, витягнути повідомлення.

2. *Активний порушник*, здатний окрім виявлення і витягання також руйнувати і видаляти повідомлення.

3. *Зловмисний порушник*, здатний, додатково до виявлення, витягання, руйнування і видалення, створювати помилкові стего.

Для стеганографічних систем прийнято визначати *невиявлюваність* – *ймовірність пропуску* (тобто відсутність виявлення стегосистеми, коли вона була представлена для аналізу), і *ймовірність помилкового виявлення* (коли стегосистема помилково виявляється при її дійсній відсутності). Практичні підходи оцінки стійкості стегосистем засновані на їх стійкості до виявлення за допомогою розроблених до теперішнього часу алгоритмів стегоаналізу. Усі вони побудовані на тому, що усі алгоритми вбудовування так чи інакше вносять до стегограми спотворення відносно використаних контейнерів.

слабкості, хакери можуть отримати доступ до комп'ютера від імені користувача, що працює з додатком (зазвичай це не простий користувач, а привілейований адміністратор з правами системного доступу). Відомості про атаки на рівні додатків широко публікуються, щоб дати можливість адміністраторам виправити проблему за допомогою корекційних модулів (патчів). На жаль, багато хакерів також мають доступ до цих відомостей, що дозволяє їм вчитися.

Головна проблема з атаками на рівні додатків полягає в тому, що вони часто використовують порти, яким дозволений прохід через міжмережевий екран. Наприклад, хакер, який експлуатує відому слабкість web-сервера, часто використовує в ході атаки TCP порт 80.

Оскільки web-сервер надає користувачам web-сторінки, міжмережевий екран повинен надавати доступ до цього порту. З точки зору брандмауера, атака розглядається як стандартний трафік для порту 80. Повністю виключити атаки на рівні додатків неможливо. Хакери постійно відкривають і публікують в Інтернет нові вразливі місця прикладних програм. Найголовніше тут - гарне системне адміністрування. Ось деякі заходи для зниження вразливості атак цього типу:

— Читайте лог-файли операційних систем і мережеві лог-файли і/або аналізуйте їх за допомогою спеціальних аналітичних програм.

— Підпишіться на послуги з розсилки даних про слабкі місця прикладних програм: Bugtrad (http://www.securityfocus.com) і CERT (http://www.cert.com).

— Користуйтеся найсвіжішими версіями операційних систем і додатків і найостаннішими коррекційними модулями (патчами).

Крім системного адміністрування, користуйтеся системами розпізнавання атак (IDS). Існують дві взаємодоповнюючі один одного технології IDS:

— мережева система IDS (NIDS) відстежує всі пакети, що проходять через певний домен. Коли система NIDS бачить пакет або серію пакетів, які збігаються з сигнатурою відомої або ймовірної атаки, вона генерує сигнал тривоги і/або припиняє сесію;

— хост-система IDS (HIDS) захищає хост за допомогою програмних агентів. Ця система бореться тільки з атаками проти одного хоста.

У своїй роботі системи IDS користуються сигнатурами атак, які представляють собою профілі конкретних атак або типів атак. Сигнатури визначають умови, за яких трафік вважається хакерським. Аналогами IDS в фізичному світі можна вважати систему попередження або камеру спостереження. Найбільшим недоліком IDS є її здатність генерувати сигнали тривоги. Щоб мінімізувати кількість хибних сигналів тривоги і домогтися коректного функціонування системи IDS в мережі, необхідне ретельне налаштування цієї системи.

Мережева розвідка

Мережевою розвідкою називається збір інформації про мережу за допомогою загальнодоступних даних і додатків. При підготовці атаки проти будь-

якої мережі хакер, як правило, намагається отримати про неї якомога більше інформації. Мережева розвідка проводиться у формі запитів DNS, тестування з відлунням (ping sweep) і сканування портів. Запити DNS допомагають зрозуміти, хто володіє тим чи іншим доменом і які адреси цього домену привласнені. Тестування з відлунням (ping sweep) адрес, розкритих за допомогою DNS, дозволяє побачити, які хости реально працюють в даному середовищі. Отримавши список хостів, хакер використовує засоби сканування портів, щоб скласти повний список послуг, що надаються цими хостами. І нарешті, хакер аналізує характеристики додатків, що працюють на хостах. В результаті видобувається інформація, яку можна використовувати для злому.

Повністю позбавитися від мережевої розвідки неможливо. Якщо, наприклад, відключити відлуння ICMP і відлуння-відповідь на периферійних маршрутизаторах, ви позбудетеся загрози тестування з відлунням, але втратите дані, необхідні для діагностики мережевих збоїв. Крім того, сканувати порти можна і без попереднього тестування з відлунням. Просто на це потрібно більше часу, так як сканувати доведеться і неіснуючі IP-адреси. Системи IDS на рівні мережі і хостів зазвичай добре справляються із завданням повідомлення адміністратора про те, що ведеться мережева розвідка, що дозволяє краще підготуватися до майбутньої атаки і сповістити провайдера (ISP), в мережі якого встановлена система, яка проявляє надмірну цікавість.

Зловживання довірою

Власне кажучи, цей тип дій не є «атакою» або «штурмом». Він являє собою зловмисне використання довіри, існуючої в мережі. Класичним прикладом такого зловживання є ситуація в периферійній частині корпоративної мережі. У цьому сегменті часто розташовуються сервери DNS, SMTP і HTTP. Оскільки всі вони належать до одного і того ж сегменту, злом одного з них призводить до злому і всіх інших, так як ці сервери довіряють іншим системам своєї мережі. Іншим прикладом є система, встановлена із зовнішнього боку брандмауера, що має довіру до системи, встановленої з його внутрішньої сторони. У разі злому зовнішньої системи, хакер може використовувати відносини довіри для проникнення в систему, захищену брандмауером. /Ризик зловживання довірою можна знизити за рахунок більш жорсткого контролю рівнів довіри в межах своєї мережі. Системи, розташовані з зовнішньої сторони брандмауера, ніколи не повинні користуватися абсолютною довірою з боку захищених екраном систем. Відносини довіри повинні обмежуватися певними протоколами і, по можливості, аутентифікуватися не тільки по IP-адресам, а й за іншими параметрами.

Переадресація портів

Переадресація портів являє собою різновид зловживання довірою, коли зламаний хост використовується для передачі через міжмережевий екран трафіку, який в іншому випадку був би обов'язково відбракований. Уявімо собі

хования впроваджуваного повідомлення: для забезпечення безперешкодного проходження стегоповідомлення каналом зв'язку воно жодним чином не повинне притягнути увагу порушника.

2. Стегоповідомлення має бути стійке до спотворень, у тому числі і зловмисним. В процесі передачі зображення (звук або інший контейнер) може зазнавати різні трансформації: зменшуватися або збільшуватися, перетворюватися в інший формат і т.д. Крім того, воно може бути стисле, у тому числі і з використанням алгоритмів стискування з втратою даних.

3. Для збереження цілісності вбудованого повідомлення потрібне використання коду з виправленням помилки.

4. Для підвищення надійності вбудоване повідомлення має бути продубльовано.

Істотний вплив на надійність стегосистеми і можливість виявлення факту передачі прихованого повідомлення робить вибір контейнера.

За принципами побудови виділяють наступні **типи стегосистем**:

1. **Безключові стегосистеми** – не вимагають ніяких додаткових даних у вигляді стегоключа окрім алгоритму стеганографічного перетворення. Їх безпека заснована на секретності використовуваних стеганографічних перетворень, що суперечить основному принципу Керкхоффа для систем захисту інформації.

2. **Стегосистеми з секретним ключем** – безпека системи ґрунтується на секретному стегоключі, без знання якого не можна витягнути з контейнера секретну інформацію. Відправник, вбудовувавши секретне повідомлення у вибраний контейнер, використовує секретний стегоключ ***k*** в стеганографічном перетворенні. Якщо використовуваний стегоключ ***k*** відомий одержувачеві, то він зможе витягнути приховане повідомлення з контейнера. Без знання такого ключа будь-який інший користувач цього зробити не зможе.

Цей тип стегосистем припускає наявність безпечного каналу для обміну стегоключами.

Іноді стегоключ ***k*** обчислюють за допомогою секретної хеш-функції **HASH**, використовуючи деякі характерні особливості контейнера.

3. **Стегосистеми з відкритим ключем** не потребують додаткового каналу ключового обміну. Для їх функціонування необхідно мати два стегоключа: один секретний, який користувач повинен зберігати в таємниці, а другий – відкритий, який зберігається в доступному для усіх місці. При цьому відкритий ключ використовується в процесі приховання інформації, а секретний – для її витягання.

4. **Змішані стегосистеми.** У більшості застосуваньприйнятними є безключові стегосистеми; хоча такі системи можуть бути відразу скомпрометовані у разі, якщо порушник дізнається про стеганографічне перетворення, що використовується. У зв'язку з цим в безключових стегосистемах часто використовують особливості криптографічних систем з відкритим і (або) секретним ключем.

❒ **стеганографічна система (стегосистема)** – об'єднання методів і засобів, використовуваних для створення прихованого каналу для передачі інформації. При побудові такої системи умовилися про те, що:

- порушник *представляє роботу стегосистеми*, однак невідомим для нього є ключ, за допомогою якого можна дізнатися про факт існування і зміст таємного повідомлення;
- при виявленні порушником наявності прихованого повідомлення він *не повинен змогти витягнути повідомлення* до тих пір, поки він не володітиме ключем;
- порушник *не має технічних та інших переваг*;

❒ **стегоповідомлення** – термін, використовуваний для загальної назви передаваної прихованої інформації, будь то лист з написами молоком, голова раба або цифровий файл;

❒ **контейнер** – будь-який фізичний або віртуальний об'єкт, використовуваний для приховання таємного повідомлення:

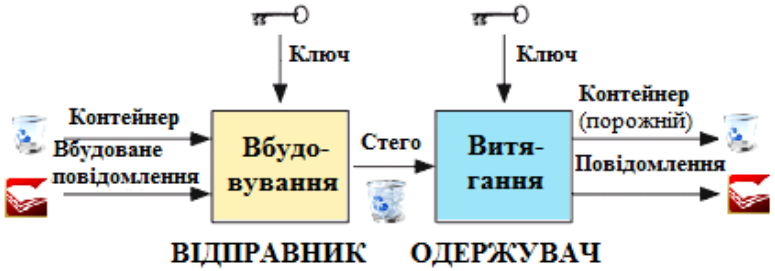
- *порожній контейнер* – контейнер, що не містить секретного послання;
- *стегоконтейнер* – заповнений контейнер, тобто контейнер, що містить секретне послання;

❒ **стеганографічний канал (стегоканал)** – канал передачі стегоконтейнера;

❒ **стегоключ** – секретний ключ, потрібний для приховання стегоконтейнера. Ключі в стегосистемах бувають двох типів:

- *закриті* (*секретні*) *ключі* (якщо стегосистема використовує закритий ключ, то він має бути створений або до початку обміну повідомленнями, або переданий по захищеному каналу);
- *відкриті ключі* (стегосистема, що використовує відкритий ключ, має бути влаштована так, щоб було неможливо отримати з нього закритий ключ. В цьому випадку відкритий ключ можна передавати незахищеним каналом).

Узагальнена модель стегосистеми



Будь-яка стегосистема повинна відповідати наступним **вимогам**:

1. Властивості контейнера мають бути модифіковані, щоб зміну неможливо було виявити при візуальному контролі. Ця вимога визначає якість при-

міжмережевий екран з трьома інтерфейсами, до кожного з яких підключений певний хост. Зовнішній хост може підключатися до хосту загального доступу (DMZ), але не до хосту, встановленому з внутрішньої сторони брандмауера. Хост загального доступу може підключатися і до внутрішнього, і до зовнішньому хосту. Якщо хакер захопить хост загального доступу, він зможе встановити на ньому програмний засіб, перенаправляє трафік з зовнішнього хоста прямо на внутрішній хост. Хоча при цьому не порушується жодне правило, чинне на екрані, зовнішній хост в результаті переадресації отримує прямий доступ до захищеного хосту. Прикладом програми, яка може надати такий доступ, є netcat.j. Більш детальну інформацію можна отримати на сайті http://www.avian.org.

Основним способом боротьби з переадресацією портів є використання надійних моделей довіри (див. вище). Крім того, перешкодити хакеру встановити на хості свої програмні засоби може хост-система IDS (H1DS).

Несанкціонований доступ

Несанкціонований доступ не може вважатися окремим типом атаки. Більшість мережевих атак проводяться заради отримання несанкціонованого доступу. Щоб підібрати логін Telnet, хакер повинен спочатку отримати підказку Telnet на своїй системі. Після підключення до порту Telnet на екрані з'являється повідомлення «authorization required to use this resource» (для користування цим ресурсом потрібна авторизація). Якщо після цього хакер продовжить спроби доступу, вони будуть вважатися несанкціонованими. Джерело таких атак може перебувати як усередині мережі, так і зовні. Способи боротьби з несанкціонованим доступом досить прості. Головним тут є скорочення або повна ліквідація можливостей хакера з отримання доступу до системи за допомогою несанкціонованого протоколу. Як приклад можна розглянути недопущення хакерського доступу до порту Telnet на сервері, який надає web-послуги зовнішнім користувачам. Не маючи доступу до цього порту, хакер не зможе його атакувати. Що ж стосується брандмауера, то його основним завданням є запобігання найпростішим спробам несанкціонованого доступу.

Висновки

1. **Захист інформації** – сукупність методів і засобів, що забезпечують цілісність, конфіденційність і доступність інформації за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації.
2. **Комп'ютерна безпека** – це сукупність проблем у галузі телекомунікацій та інформатики, пов'язаних з оцінкою і контролюванням ризиків, що виникають при користуванні комп'ютерами та комп'ютерними мережами і розглядуваних з точки зору конфіденційності, цілісності і доступності.

3. **Загроза безпеки КС** – сукупність умов і чинників, що визначають потенційну або реально існуючу небезпеку порушення конфіденційності, цілісності, (правомірної) доступності комп'ютерної інформації, спостереженості та керованості КС, і/або зниження надійності (безвідмовності і автентичності) реалізації функцій КС.

4. **Політика безпеки інформації** – сукупність законів, правил, обмежень, нормативних документів, рекомендацій, інструкцій тощо, які регламентують порядок обробки інформації в КС і спрямованих на захист інформації від певних видів загроз.

5. Основними механізмами, що забезпечують безпеку інформації в КС, які реалізовані в ОС сімейства MS Windows є засоби контролю і управління доступом в комп'ютерну систему, до файлів і папок, що зберігаються і обробляються; засоби, що забезпечують безпеку інформації в місцях її зберігання, в процесі обробки і в ході передачі каналами зв'язку (криптографічний захист); засоби захисту від шкідливого програмного забезпечення (антивірусний захист); засоби захисту периметра комп'ютерної системи (міжмережеві екрани – брандмауери).

6. **Дозвіл** – правило, пов'язане з об'єктом і використовуване для управління доступом користувачів до цього об'єкту. Передбачений як стандартний набір дозволів (для загальних випадків), так і спеціалізований набір – для «тонкого» налаштування.

Питання для самоконтролю

1. Що таке загроза інформації в ПК?
2. У чому полягає безпека інформації в ПК?
3. Як класифікуються загрози безпеки в ПК?
4. Які цілі забезпечення інформаційної безпеки в ПК?
5. Які функції ОС Windows забезпечують захист інформації в ПК?
6. Що таке неправдиві сервери?
7. Особливості нав'язування в мережі Internet.
8. Як здійснюється підтримка маршрутів?
9. Як здійснюється впровадження в мережі Internet неправдивих серверів?
10. Які атаки здійснюються в комп'ютерних системах?
11. Особливості атак на комп'ютерні системи?

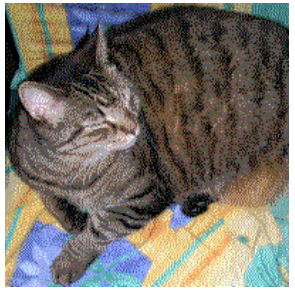
Література

1. Ленков С.В. Методы и средства защиты информации. В 2-х томах / С.В. Ленков, Д.А. Перегудов, В.А. Хорошко. Том 1. Несанкционированное получение информации – К: Арий, 2008.- 464 с.

дять до помітних змін цих об'єктів. Крім того, в оцифрованих об'єктах, що спочатку мають аналогову природу, завжди присутній шум квантування; далі, при відтворенні цих об'єктів з'являється додатковий аналоговий шум і нелінійні спотворення апаратури, усе це сприяє більшій непомітності прихованої інформації.

Стегоконтейнер – зображення дерева з прихованим за допомогою цифрової стеганографії в нім іншого зображення

Зображення кота, витягнуте із зображення дерев



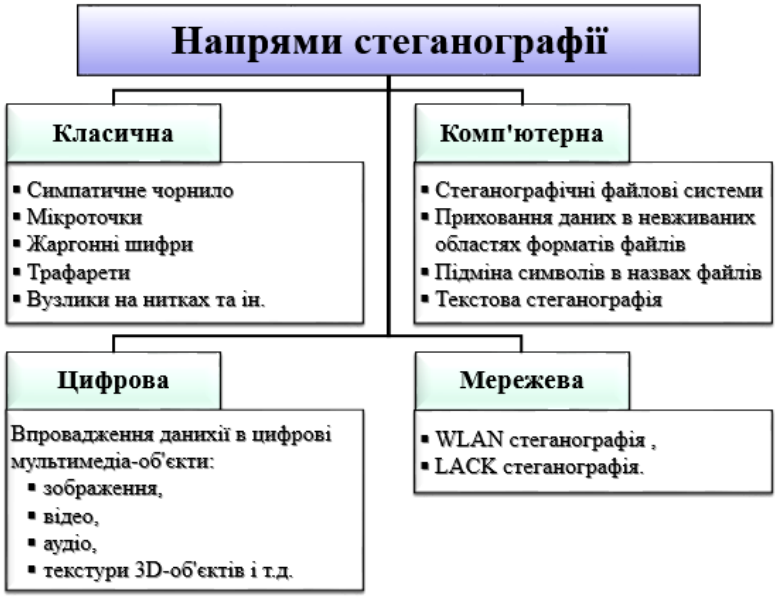
Мережева стеганографія – напрям, заснований на прихованні секретної інформації, що передається через комп'ютерні мережі, з *використанням особливостей роботи протоколів передачі даних*. Методи мережевої стеганографії включають зміну властивостей одного з мережевих протоколів. Основні методи мережевої стеганографії:

- **WLAN-стеганографія**грунтується на методах, які використовуються для передачі стеганограм в безпровідних локальних обчислювальних мережах (*Wireless Local Area Networks*). Наприклад, прихована система зв'язку HICCUPS (*Hidden Communication System for Corrupted Networks*);
- **LACK-стеганографія** – приховання повідомлень під час розмов з використанням IP-телефонії. Наприклад, використання пакетів, які затримуються, або навмисно ушкоджуються та ігноруються приймачем (*Lost Audio Packets Steganography* – стеганографія, заснована на втрачених аудіопакетах) або приховання інформації в полях заголовка, які не використовуються.

Принцип функціонування LACK виглядає так: передавач ***A*** вибирає один з пакетів голосового потоку, і його корисне навантаження замінюється бітами секретного повідомлення – стеганограмой, яка вбудовується в один з пакетів. Потім вибраний пакет навмисно затримується. Кожного разу, коли надмірно затриманий пакет досягає одержувача, незнайомого із стеганографічною процедурою, він відкидається. Проте, якщо одержувач ***B*** знає про прихований зв'язок, то замість видалення отриманих RTP-пакетів він витягає приховану інформацію.

Понятійний апарат стеганографії

У 1996 році на конференції **Information Hiding: First Information Workshop** була прийнята єдина термінологія у галузі стеганографії:



Комп'ютерна стеганографія – це використовування *особливостей комп'ютерної платформи*, наприклад,

- ❒ стеганографічна файлова система **StegFS** для ОС Linux;
- ❒ приховання даних в невживаних областях форматів файлів;
- ❒ підміна символів в назвах файлів;
- ❒ текстова стеганографія;
- ❒ використання зарезервованих полів комп'ютерних форматів файлів: частина поля розширень, не заповнена інформацією про розширення, за замовчуванням заповнюється нулями, тоді можна використовувати цю частину для запису приховуваних даних;
- ❒ використання особливих властивостей полів форматів, які не відображуються на екрані, наприклад, написання чорним шрифтом на чорному фоні;
- ❒ використання особливостей файлових систем – при зберіганні на жорсткому диску файл завжди займає ціле число кластерів, наприклад, у файловій системі FAT32 стандартний розмір кластера – 4 Кбайт. Відповідно для зберігання 1 Кбайт інформації на диску виділяється 4 Кбайт пам'яті, з яких 1 Кбайт потрібний для зберігання файлу, а інші 3 ні на що не використовуються – їх можна використовувати для таємного зберігання інформації.

Цифрова стеганографія – напрям, заснований на прихованні або впровадженні секретної інформації в цифрові мультимедіа-об'єкти (зображення, відео, аудіо, текстури 3D-об'єктів і т.д.), викликаючи при цьому деякі спотворення цих об'єктів.

Спотворення знаходяться нижче за поріг чутливості людини і не призво-

2. Андреєв В.І. Стратегія управління інформаційною безпекою / В.І. Андреєв, С.Д. Козюра, Л.М. Скачек, В.О. Хорошко – К: ДУІКТ, 2007. – 277 с.
3. Бурячок В.Л. Політика інформаційної безпеки / В.Л. Бурячок, Р.В. Грищук, В.О. Хорошко – К: ВПП «Задруга», 2014. – 222 с.
4. Коженевський С.Р. Термінологічний довідник з технічного захисту інформації на об’єктах інформаційної діяльності / С.Р. Коженевський, Г.В. Кузнєцов, В.О. Хорошко, Д.В. Чирков / К: ДУІКТ, 2007. – 365 с.
5. Головань С.М. Нормативно – правове забезпечення інформаційної безпеки / С.М. Головань, О.С. Петров, В.О. Хорошко – Луганськ: ВИД Наука, 2012. – 480 с.
6. Андреєв В.І. Основи інформаційної безпеки. Вид. 2-е / В.І. Андреєв, С.Д., В.О. Хорошко, В.С. Чередніченко, М.Є. Шелест – К: ДУІКТ, 2009. – 292 с.
7. Дудикевіч В.Б. Основи інформаційної безпеки / В.Б. Дудикевіч, В.О. Хорошко, Ю.Є. Яремчук – Вінниця: ВНТУ, 2018. – 316 с.
8. Єжова Л.Ф. Управління інформаційною безпекою : підручник : у 2 т., Т. 1 / Л.Ф.Єжова, А.О. Корченко, І.О. Мачалін, Л.М. Скачек, В.О. Хорошко. - К., 2012. - 369 с.

2. СЕРВІСИ ЗАХИСТУ ІНФОРМАЦІЇ В MS WINDOWS

2.1 Механізми безпеки, які реалізовані в ОС MS Windows

У будь-якому середовищі, що надає доступ до одних і тих же фізичних або мережевих ресурсів відразу декільком користувачам, гостро стоїть питання запобігання неавторизованому доступу до конфіденційних даних. Операційна система, разом з окремими користувачами, повинна мати можливість захистити файли, пам'ять і налаштування конфігурації від небажаного перегляду і зміни. Засоби безпеки ОС включають такі цілком очевидні механізми, як облікові записи, паролі і захист файлів. Вони також включають такі менш помітні механізми, як захист ОС від ушкодження, недопущення здійснення ряду дій (наприклад, перезавантаження комп'ютера) з боку менш привілейованих користувачів і заборона несприятливої дії призначених для користувача програм на програми інших користувачів або на операційну систему.

Розробники операційної системи Windows приділили серйозну увагу забезпеченню безпеки роботи користувачів. Це підтверджується категоріями, присвоєними різним версіям цієї операційної системи по тих або інших міжнародних і національних критеріях оцінки безпеки. Так, по класифікації **«Помаранчевої книги»** ОС Windows NT 4 ще в 1999 році отримала клас безпеки **C2**, за стандартом **ISO/IEC 15408 Common Criteria for Information Technology Security Evaluation** (*Загальні критерії оцінки безпеки інформаційних технологій*) клієнтські і серверні версії від Windows 2000 до Windows 10, від Windows Server 2008 до Windows Server 2013 отримали високий рівень гарантованої безпеки**EAL 4+**.

Використання в сучасних версіях операційної системи MS Windows файлової системи **NTFS** відкриває широкі можливості з організації захисту інформації в процесі її зберігання, обробки і передачі.

Основними механізмами, що забезпечують безпеку інформації в КС, які реалізовані в ОС сімейства MS Windows є:

- ❒ засоби контролю і управління доступом в комп'ютерну систему, до файлів і папок, що зберігаються і оброблюються;
- ❒ засоби, що забезпечують безпеку інформації в місцях її зберігання, в процесі обробки і в ході передачі каналами зв'язку (криптографічний захист);
- ❒ засоби захисту від шкідливого програмного забезпечення (антивірусний захист);
- ❒ засоби захисту периметра комп'ютерної системи (міжмережеві екрани – брандмауери).

контенту. Наприклад, це може бути фотографія. У випадку якщо фотографію опублікують без дозволу фотографа, сказавши, що нібито не він автор цієї роботи, фотограф може спробувати довести своє авторство за допомогою стеганографії. В даному випадку в кожну фотографію необхідно вбудувати інформацію про серійний номер фотоапарата або інші дані, що дозволяють «прив'язати» фотографію до одного єдиного фотоапарата;

2) *захист достовірності документів* – стеганографія використовується не для підтвердження авторства, а для підтвердження достовірності документу (документ, що не містить СВЗ, вважається підробним);

3) *водяний знак в системах запобігання витоку даних* (*Data Leak Prevention*, *DLP*) – при створенні документу, що має конфіденційний характер, украплюється певна мітка, яка не змінюється незалежно від кількості копій і/або ревізій документу.Для витягання мітки потрібний стегоключ, який тримається в таємниці. DLP-система перед схваленням або відмовою видати документ зовні, перевіряє наявність або відсутність водяного знаку: якщо знак присутній, то система не дозволяє відправляти документ зовні системи.

4) *невідчужуваність інформації* – існує ряд документів, для яких важлива цілісність. Її можна здійснити резервуванням даних. Але що робити, якщо є необхідність мати документи у такому вигляді, щоб неможливо було одну інформацію відокремити від іншої інформації? Як приклад можна привести медичні знімки. Пропонується усередину знімків украплювати інформацію про ім'я, прізвище та інші дані пацієнта.

Напрями стеганографії

Класична стеганографія. Перший запис про використання стеганографії зустрічається в трактаті Геродота «Історія» (440 рок до Р.Х.): на поголену голову раба записувалося повідомлення, а коли його волосся відростало, він вирушав до адресата, який знову голив його голову і прочитував доставлене повідомлення.

Використання *симпатичного чорнила* (з'явилися в I столітті н.е.). Текст, записаний таким чорнилом, проявляється тільки за певних умов (нагрів, освітлення, хімічний проявник і т.д.). Існує також чорнило з хімічно нестабільним пігментом.

Під час Другої світової війни активно використовувалися *мікроточки* – мікроскопічні фотознімки (розміром до 1мм), що вклеюються в текст листів.

Альтернативні методи класичного приховання інформації:

- ❒ запис усередині вареного яйця;
- ❒ «жаргонні шифри», де слова мають інше обумовлене значення;
- ❒ трафарети, які, будучи покладеними на текст, залишають видимими тільки значущі букви;
- ❒ вузлики на нитках і т.д.

MPEG4, AVI, WMV, MPEG-PS, FLV, 3GPP, WebM). Проте можна використовувати стеганографію для зберігання даних в інших форматах. Наприклад, сайт hid.im дозволяє користувачам приховувати файли .torrent усередині зображень PNG;

4) *прихована передача управляючого сигналу* – стеганографія застосовується для доставки якого-небудь управляючого сигналу системі в таємниці від супротивника. Використання тільки криптографії, без стеганографії, може дати супротивникові інформацію про те, що щось змінилося і спровокувати його на небажані дії;

5) *стеганографічні botnet-мережи* – це застосування є часткою випадком прихованої передачі управляючого сигналу від органу управління бот-мережею на заражені комп'ютери з метою організації кібератаки;

6) *Funkspiel* («*Радіогра*») – стегоповідомлення містить дані (наприклад, яку-небудь хеш-функцію або наперед встановлену послідовність біт), повідомляючі про те, чи варто сприймати інформацію контейнера серйозно;

7) *стеганографічне відвернення* – задача – відвернути увагу супротивника. Для цього необхідно, щоб генерація стегоконтейнерів була істотно «дешевша» (з точки зору машинних і тимчасових ресурсів), ніж виявлення стеганографії супротивником. Стеганографічне відвернення чимось нагадує DoS і DDoS атаки. Відволікається увага супротивника від контейнерів, які дійсно містять щось цінне;

8) *стеганографічне відстежування* – тут мета стеганографії – піймати порушника, який «зливає» інформацію (аналог «мічених грошей», використовуваних правоохоронними органами, для того, щоб злочинець, що отримав гроші за яку-небудь незаконну діяльність, не міг би потім заявити, що ці гроші були у нього до угоди).

II. Цифровий відбиток:

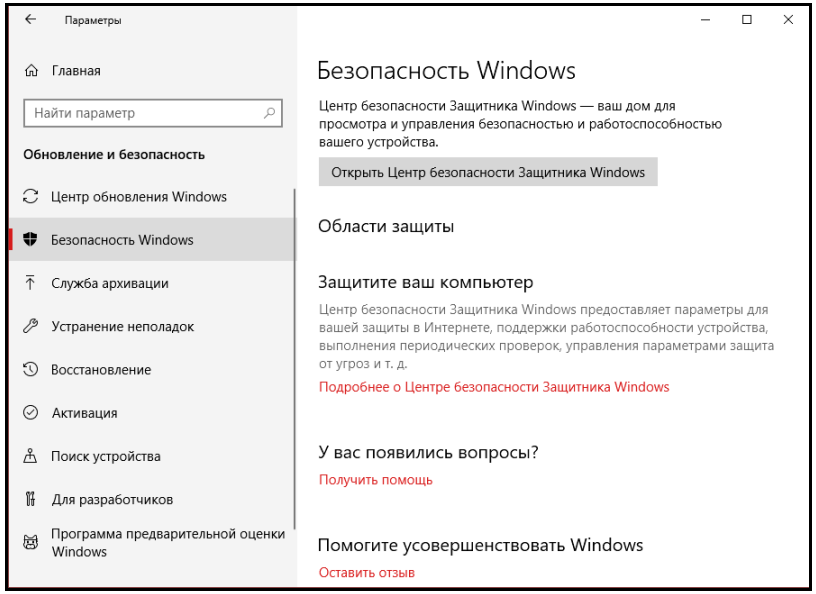
1) *захист виняткового права* – наприклад, в голографічних багатоцільових дисках (Holographic Versatile Disc, HVD), що містять до 200 Гб даних і використовуваних компаніями теле- і радіомовлення для зберігання відео- і аудіоінформації, наявність ЦВ усередині кодів цих дисків може використовуватися як засіб для захисту ліцензійного права; в Інтернет-продажи інформаційних ресурсів (книг, фільмів, музики і т.д.) кожна копія повинна містити спеціальну мітку для перевірки ліцензійна ця копія або не ліцензійна;

2) *індивідуальний відбиток в системі електронного документообігу* – використання індивідуального відбитку усередині *.docx та інших документів при роботі з ними користувачів дозволяє пізнати, хто працював з документом, а хто ні;

3) *підтвердження достовірності переданої інформації* – стегоповідомлення містить дані, такі, що підтверджують коректність передаваних даних в стегоконтейнері. Це може бути контрольна сума або хеш-функція (дайджест).

III. Стеганографічний водяний знак:

1) *захист авторського права* – одним знаком захищається кожна копія



2.1.1 Контроль посвідчень і доступу

Функції контролю посвідчень і доступу забезпечуються сервісами **Windows Hello** і **Microsoft Passport**, які краще захищають посвідчення користувачів завдяки простій в розгортанні і використанні багатофакторній перевірці достовірності (MFA). Ще однією новою функцією є **Credential Guard**, що використовує систему безпеки на основі віртуалізації (**VBS**) для захисту підсистем перевірки достовірності Windows і облікових даних користувачів.

Багатофакторна автентифікація (*MFA*, *multi-factor authentication*) – розширена автентифікація, метод контролю доступу до комп'ютера, в якому користувачеві для діставання доступу до інформації необхідно пред'явити більш за один «доказ механізму автентифікації». До категорій таких доказів відносять:

- *знання* – інформація, яку знає суб'єкт (пароль, пін-код);
- *володіння* – річ, яку має суб'єкт (електронна або магнітна карта, токен, флеш-пам'ять);
- *властивість*, яку має суб'єкт (біометрія, природні унікальні відмінності: особа, відбитки пальців, веселкова оболонка очей, капілярні візерунки, послідовність ДНК).

Контроль доступу до КС, її програмних і інформаційних ресурсів є процесом, який складається з трьох компонентів.

- **Ідентифікація** – користувач вказує своє унікальне посвідчення в комп'ютерній системі з метою доступу до ресурсу, наприклад файлу або принтеру.
- **Автентифікація(Перевірка достовірності)** – процедура підтвердження вказаного посвідчення і перевірка того, що суб'єкт дійсно є тим, за кого себе видає.
- **Авторизація** – виконується системою з метою порівняти права доступу суб'єкта, що пройшло перевірку достовірності, і дозволу об'єкту і або надати запитаний доступ, або відмовити в ньому.

Реалізація цих компонентів істотно зміцнює захист таємних даних від порушників. Тільки користувач, що підтвердив свою особу і що отримав право на доступ до даних, зможе здійснити доступ. У системі безпеки існують різні міри перевірки достовірності посвідчень і безліч різних вимог до лімітів авторизації. Забезпечення гнучкості контролю доступу, необхідної в більшості середовищ підприємств, є складним завданням для будь-якої ОС.

Рішення для завдань в області контролю доступу в Windows 10

Завдання контролю доступу	Рішення Windows 10
Організації часто використовують паролі. Організації, що вибирають альтернативи паролю, наприклад смарт-карти, повинні придбавати прочитуючи пристрої для смарт-карт, смарт-карти і ПЗ для управління ними, а також управляти усіма цими ресурсами	**Windows Hello** на пристроях з підтримкою біометрії і **Microsoft Passport** значно спрощують **MFA**
Користувачі планшетів повинні вводити пароль на сенсорному екрані, тому можливі помилки. В цілому цей метод менш ефективний, чим введення з клавіатури	**Windows Hello** дозволяє безпечно проводити перевірку достовірності на основі розпізнавання особи
Відділ IT повинен придбати засоби сторонніх постачальників і управ-	У поєднанні з ОС **Windows Server 2012** динамічний контроль доступу забезпечує мо-

5. СТЕГАНОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

5.1 Стеганографічні методи захисту інформації

5.1.1 Поняття стеганографії

Стеганографія (з грецької Στεγανός — *прихований* + γράφω — *пишу*; буквально «*тайнопис*») – міждисциплінарна наука і мистецтво про приховану передачу або зберігання інформації з урахуванням збереження в таємниці самого факту такої передачі (зберігання).

На відміну від криптографії, яка приховує вміст таємного повідомлення, *стеганографія приховує сам факт його існування*.

Включає сукупність методів, що ґрунтуються на різних принципах, які забезпечують приховання самого факту існування секретної інформації в тому або іншому середовищі, а також засобів реалізації цих методів. Так тільки у США опублікована близько сотні патентів із стеганографії, включаючи такі специфічні, як патент Kursh K., Lav R. «Food steganography» («*Продовольча стеганографія*»).

Метою стеганографії є створення:

- **прихованої передачі даних (ППД)** – це «класична» мета стеганографії, відома з IV століття до Р.Х. Завдання – передати дані так, щоб супротивник не здогадався про сам факт появи повідомлення;
- **цифрового відбитку (ЦВ)** – різних стеганографічних міток-повідомлень для кожної копії контейнера;
- **стеганографічного водяного знаку (СВЗ)** – стеганографічної мітки, однакової для кожної копії контейнера.

Практичне застосування стеганографії:

І. Прихована передача даних:

1) *непомітна передача інформації* - на відміну від криптографічних методів (які таємниці, але не потайні), стеганографія застосовується як метод непомітної передачі інформації (це складає класичне практичне її застосування);

2) *приховане зберігання інформації* – стеганографія використовується для зберігання якої-небудь інформації, виявлення самого факту наявності якої (нехай хоч навіть в зашифрованому виді) користувачеві небажано. Надмірність на багатьох носіях може бути неймовірна великою (наприклад, загальний об'єм даних, які можна записати на CD диск складають 1828 Мб даних – це величезна надмірність, яку можна використовувати для приховання даних);

3) *зберігання інформації, яка не декларується* – багато інформаційних ресурсів дозволяють зберігати дані тільки певного виду (наприклад портал YouTube дозволяє зберігати тільки відеоінформацію у форматах MOV,

6. Кузнецов О.О. Захист інформації в інформаційних системах – О.О. Кузнецов, С.П. Євсеєв, О.Г. Король – Харків: ХНЕУ, 2011. – 512 с.
7. Гулак Г.М. Основи криптографічного захисту інформації / Г.М. Гулак, В.А. Мухачов, В.О. Хорошко, Ю.Є. Яремчук – Вінниця: ВНТУ, 2011. – 199 с.
8. Ємець В. Сучасна криптографія. Основні поняття / В. Ємець, А. Мельник, Р. Попович – Львів: БаК, 2003. – 144 с.
9. Горбенко І.Д. Прикладна криптографія. Теорія, практика, застосування / І.Д. Горбунко, Ю.І. Горбенко – Харків: ВИД. «Форт», 2012. – 880 с.

ляти ними, щоб забезпечити дотримання нормативних вимог до контролю доступу і аудиту.	жливість гнучкого контролю доступу і аудиту з дотриманням численних вимог регулюючих органів в області безпеки
Користувачам не подобається вводити паролі	Єдиний вхідзабезпечує можливість одноразового входу по паспорту **Microsoft Passport** і дістання доступу до усіх ресурсів організації без повторної перевірки достовірності. **Windows Hello** дозволяє виконувати безпечну перевірку достовірності на основі розпізнавання відбитків пальців і обличчя. Цей спосіб можна використовувати для повторного підтвердження достовірності користувача при здійсненні доступу до конфіденційних ресурсів.
Windows додає затримки, що збільшуються за тривалістю, між спробами входу і може заблокувати обліковий запис користувача у разі атаки методом підбору	Якщо на системному диску включений **BitLocker** і активований захист від атак методом підбору, Windows може перезавантажити ПК після певної кількості невдалих спроб ввести пароль, заблокувати доступ до жорсткого диска і зажадати від користувача введення 48-значного ключа відновлення **BitLocker**, щоб запустити пристрій і дістати доступ до диска

Microsoft Passport дозволяє виконувати строгу двохфакторну перевірку достовірності (**2FA**). Ця технологія замінює паролі комбінацією із зареєстрованого пристрою і пін-кода або **Windows Hello**. Технологія Microsoft Passport по суті схожа з технологією смарт-карт, проте є гнучкішою. Перевірка достовірності виконується з використанням пари асиметричних ключів, а не порівняння рядків (наприклад, пароля), і матеріал ключа користувача можна забезпечити апаратним захистом.

На відміну від смарт-карт **Microsoft Passport** не вимагає додаткових інфраструктурних компонентів для розгортання. Зокрема, не потрібно інфраструктуру відкритих ключів (PKI).

Microsoft Passport відрізняється безпрецедентною гнучкістю. Незважаючи на те що формат і застосування паролів і смарт-карт є фіксованими, Microsoft Passport дає як адміністраторам, так і користувачам можливості управляти перевіркою достовірності. По-перше, Microsoft Passport працює з біометричними датчиками і пін-кодами. По-друге, можна використовувати ПК або навіть телефон як одного з чинників перевірки достовірності на ПК. Нарешті, облікові дані користувача можуть поступати з інфраструктури PKI. Крім того, Windows сама може створити облікові дані.

Microsoft Passport ефективно знижує дві головні загрози безпеки. По-перше, виключається використання паролів для входу в систему. Отже, знижується ризик розкрадання і використання облікових даних користувача порушником. По-друге, оскільки технологія Microsoft Passport має на увазі використання пар асиметричних ключів, облікові дані користувачів не будуть викрадені у разі порушення безпеки постачальника посвідчень або веб-сайтів, до яких користувач здійснює доступ.

Windows Hello– цебіометрична технологія входу для **Microsoft Passport**, що дозволяє розблоковувати пристрої за допомогою обличчя або відбитків пальців. Отже, перевірка достовірності на пристроях і для ресурсів забезпечується комбінацією унікального біометричного ідентифікатора користувача і самим пристроєм.

Біометричні дані користувача, які використовуються для Windows Hello, вважаються локальним елементом, вони не переміщаються між пристроями користувача і не зберігаються централізовано. Біометричне зображення користувача, яке робить датчик, перетвориться в алгоритм, який неможливо перетворити назад в початкове зображення (зняте датчиком). Пристрої з TPM 2.0 шифрують біометричні дані в недоступній для читання формі, що захищає конфіденційну інформацію на випадок переміщення з пристрою. Якщо декілька користувачів спільно використовують пристрій, то кожен користувач зможе реєструватися і використовувати **Windows Hello** для свого профілю Windows.

Windows Hello підтримує два типи біометричних датчиків:

- ❒ *розпізнавання обличчя* використовує спеціальні інфрачервоні камери, щоб відрізнити фотографію або скан від живої людини;
- ❒ *розпізнавання відбитків пальців* використовує спеціальний датчик для сканування відбитків пальців.

Windows Hello забезпечує ряд великих переваг:

- ❒ вирішені проблеми розкрадання і поширення облікових даних, тому що порушникові треба не лише дістати доступ до пристрою, але і підробити біометричні дані користувача, що набагато складніше, ніж викрасти пароль або пін-код;
- ❒ використання біометрії дає користувачам механізм перевірки достовірності, який завжди при них: його неможливо забути, втратити або залишити будинки;
- ❒ не треба розгортати додаткові драйвери, оскільки підтримка **Windows Hello** інтегрована безпосередньо в ОС.

2.1.2 Криптографічний захист інформації в MS Windows 10

Для криптографічного захисту інформації в Windows 10 використовуються сервіси **BitLocker** і **BitLocker To Go**. Крім того реалізовано шифрування на рівні файлів, використовується також і система захисту корпоративних даних, що виконує розподіл і ізолювання даних. У поєднанні із службою **Rights Management** ця технологія дозволяє зберегти дані зашифрованими, коли вони покидають мережу підприємства. Windows 10 також забезпечує безпеку даних за допомогою віртуальних приватних мереж (**VPN**) і **IPSec**.

Де б не зберігалися конфіденційні дані, їх необхідно захистити від несанкціонованого доступу. В ОС Windows послідовно удосконалюються механізми захисту даних: покращуються існуючі схеми і з'являються нові стратегії. Так, за допомогою **BitLocker** можна шифрувати не лише повні і переносні

4. Найбільш вдалим варіантом можна вважати використання комбінованих криптосистем, коли асиметричні системи використовуються для передачі секретного ключа між абонентами, а само повідомлення шифрується і передається за допомогою симетричних криптосистем. Перспективним напрямом використання криптосистем з відкритим ключем є їх застосування в протоколах розподілу ключів і цифрового підпису.

Питання для самоконтролю

1. Що є криптологія? Які основні напрями вона включає?
2. Дайте визначення шифру, ключу, криптосистемі.
3. Які загрози існують з точки зору криптоаналітика? Як класифікуються атаки криптоаналітика?
4. Що є основним об'єктом криптографії? Які вимоги пред'являються до криптографічного закриття інформації?
5. Що є моделлю Шеннона криптосистеми з секретним ключем?
6. Що є шифрами підстановки (заміни)? Наведіть приклади таких шифрів.
7. Що є шифрами переставлення? Наведіть приклади таких шифрів.
8. Що є криптосистемами з відкритим ключем? На яких принципах вони будуються?
9. Що таке одностороння функція і функція з «лазівкою»? Наведіть приклади таких функцій.
10. Як реалізується обмін ключами в асиметричної криптосистемі?
11. Що таке цифровий підпис і як він реалізується?
12. Як здійснюється тестування криптографічних систем?
13. Особливості методології тестування криптографічних програмних систем.
14. Якість методів тестування криптографічних систем.

Література

1. Указ Президента України від 22.05.1998 р. № 505/98 «Про Положення про порядок здійснення криптографічного захисту інформації в Україні».
2. Указ Президента України від 11.02.1998 р. № 110/98 «Про заходи щодо вдосконалення криптографічного захисту інформації в телекомунікаційних та інформаційних системах».
3. Диффи У., Хеллмэн М.Э. Новые направления в криптографии. ТИИЭР, № 22, 1976, с. 644-654.
4. Шеннон К. Теория связи в секретных системах. В «Работы по теории информации и кибернетике», с. 333-402, – М.: Изд. ИЛ, 1963.
5. Мухачев В.А., Хорошко В.А. Методы практической криптографии. – К.: ООО «Полиграф-Консалтинг», 2005. – 215 с.

❒ засоби – вказуються необхідні засоби тестування, включаючи план, в якому визначається їх розробник або майбутній користувач, а також спосіб і місце їх використання;

❒ машинний час – розробляється план надання машинного часу для кожної фази тестування;

❒ конфігурація апаратури – якщо потрібна спеціальна конфігурація апаратури, план тестування описує етапи, на яких в ній виникає необхідність, і вимоги, які вона повинна задовольняти;

❒ людські та фінансові ресурси;

❒ комплексування – тут визначається, як буде зібрана дана програма;

❒ процедури відстежування – вказуються засоби відстеження різних аспектів прогресу в тестуванні і його оцінки по відношенню до графіка, ресурсів і критерію завершення;

❒ процедури налагодження – визначаються механізми видачі повідомлень про виявлені помилки, відстеження складу і порядку внесення доповнень в систему;

❒ ризики якості програмного продукту.

Цей план є свого роду моделлю всього процесу тестування, але в той же час він не є незмінним. Навпаки, безпосередньо в процесі самого тестування в план можуть вноситися зміни.

Висновки

1. У сучасних ІТС криптоалгоритми знаходять широке застосування не лише для вирішення завдань захисту даних, але і для автентифікації і перевірки цілісності даних, розподілу секретних ключів, захисту авторських прав і т.д. На сьогодні існують добре відомі і апробовані алгоритми (як з симетричними, так і несиметричними ключами), стійкість яких або доказана математично, або заснована на необхідності рішення математично складного завдання.

2. В той же час, не слід вважати, що шифрування забезпечить надійний захист інформації, якщо нехтувати певними моментами. Дійсно, часто з'являються повідомлення про помилки або «діри» в тій або іншій криптосистемі, або про те, що вона була зламана хакерами. Це створює недовіру як до конкретних програм, так і до можливості взагалі захистити що-небудь криптографічними і стеганографічними методами.

3. Основна перевага криптосистем з відкритим ключем – відсутність необхідності в закритому каналі зв'язку для обміну ключовою інформацією. Ідея побудови криптосистем з відкритим ключем полягає у використанні односторонніх функцій, для яких досі не розроблені методи отримання зворотного перетворення окрім повного перебору варіантів ключа. Для забезпечення заданої стійкості секретний ключ, за допомогою якого одержувач криптограми може розшифрувати її, вибирається випадковим чином за певними правилами цілочисельної арифметики.

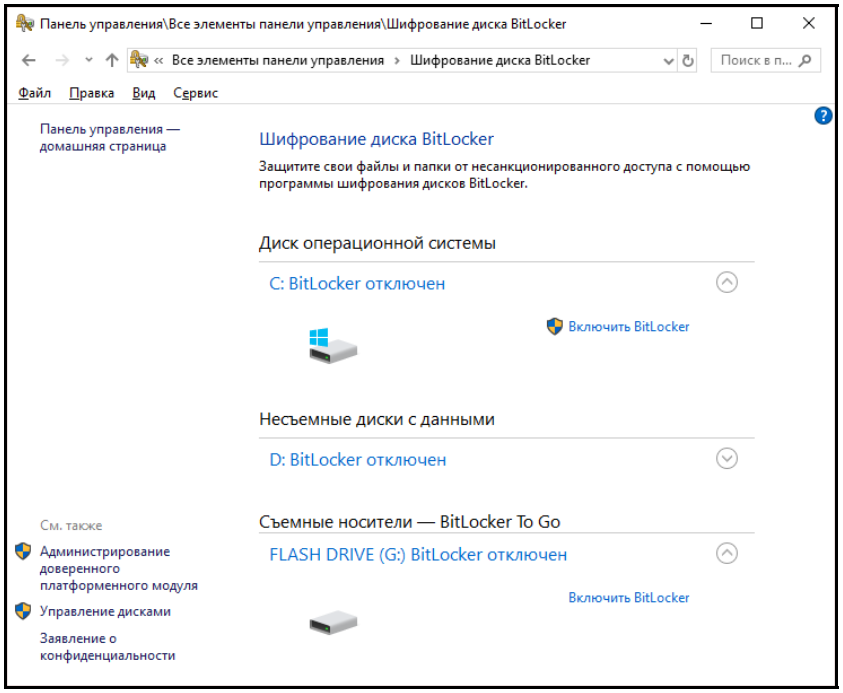
диски, але і захищати навіть окремі файли, реалізована можливість запобігання втраті даних.

Захист даних в Windows 10 і Windows 7

Windows 7	Windows 10
Якщо **BitLocker** використовується разом з пін-кодом для забезпечення безпеки в ході завантаження, перезавантажити ПК видалено неможливо.	Захист сучасних пристроїв Windows забезпечується готовою системою шифрування пристрою і підтримкою **SSO**, що забезпечує зручний захист ключів шифрування **BitLocker** від атак методом холодного перезавантаження. Мережеве розблокування дозволяє виконувати автоматичний запуск ПК за наявності підключення до внутрішньої мережі
Для зміни пін-кода або пароля **BitLocker** користувачі повинні звернутися у відділ ІТ.	Сучасні пристрої Windows більш не вимагають пін-код в передзавантажувальному середовищі, щоб захистити ключі шифрування **BitLocker** від атак за допомогою холодного перезавантаження. Користувачі із стандартними привілеями можуть змінити свій пін-код або пароль **BitLocker** на застарілих пристроях, де вимагається пін-код
Якщо **BitLocker** включений, процедура підготовки може зайняти декілька годин	Попередня підготовка **BitLocker**, шифрування жорстких дисків і шифрування тільки використовуваного простору дозволяє адміністраторам швидко включати **BitLocker** на нових комп'ютерах
Відсутня **підтримка** використання BitLocker з самошифрующимися дисками	**BitLocker** підтримує розвантаження шифрування на зашифровані жорсткі диски.
Адміністратори повинні використовувати для управління зашифрованими жорсткими дисками спеціальні засоби	**BitLocker** підтримує зашифровані жорсткі диски завдяки вбудованому устаткуванню для шифрування, що дозволяє адміністраторам використовувати знайомі інструменти адміністрування BitLocker для управління ними
Шифрування нового флеш-диска може зайняти більше 20 хвилин.	Шифрування тільки використовуваного простору в **BitLocker To Go** дозволяє шифрувати диски за секунди
BitLocker може зажадати від користувачів введення ключа відновлення при внесенні змін до конфігурації системи	**BitLocker** вимагає введення ключа відновлення тільки у разі ушкодження диска або втрати пін-кода або пароля користувачем
Користувачі повинні ввести пін-код для запуску ПК, а потім пароль для входу в Windows.	Захист сучасних пристроїв Windows все частіше забезпечується готовою системою шифрування пристрою і підтримкою **SSO**, що допомагає захистити ключі шифрування **BitLocker** від атак методом холодного перезавантаження

SSO (*Single Sign-On*) –технологія єдиного входу.

Windows 10 пропонує надійні і зручні рішення для шифрування цілих томів, знімних пристроїв або окремих файлів. Можна зробити певні заходи заздалегідь, щоб підготуватися до шифрування даних і зробити розгортання максимально простим і швидким.



BitLocker може шифрувати жорсткі диски повністю, включаючи системні диски і диски з даними. Попередня підготовка **BitLocker** дозволяє істотно скоротити час, необхідний для включення **BitLocker** на нових ПК.

Windows 10 автоматично включає шифрування пристроїв за допомогою **BitLocker** на пристроях, підтримувальних **InstantGo**. Шифрування пристроїв забезпечує додатковий захист системи завдяки прозорій реалізації шифрування даних в масштабах пристрою.

На відміну від стандартної реалізації **BitLocker** шифрування пристрою включається автоматично, тому пристрій захищений завжди.

Зашифровані жорсткі диски мають вбудовані криптографічні можливості, що дозволяє шифрувати дані на дисках, підвищує продуктивність дисків і системи завдяки перенесенню криптографічних обчислень з процесора ПК на сам диск і швидкому шифруванню диска з використанням спеціального виділеного устаткування.

Якщо **BitLocker** включений на системному диску, можна зажадати від користувачів введення пін-кода, перш ніж **BitLocker** розблоковує диск. Ця вимога захищає від порушників, що дістали фізичний доступ до ПК, і не дозволяє їм навіть дійти до входу в систему Windows. В результаті дістати доступ до даних користувача або системних файлів або змінити їх практично неможливо.

Для подальшої обробки існує безліч методик, що дозволяють виконати аналіз ризиків якості. Далі наведені найбільш поширені з них.

Методика неформального аналізу. Визначити потенційні проблеми системи, а потім спільно з ключовими особами розставити їх за пріоритетами. Як джерело списку потенційних проблем використовують стандартні списки проблем якості ПЗ, списки проблем предметної області, загальні знання про види дефектів, властивих типу системи, яка знаходиться в розробці.

Стандарт Software engineering – Software product quality ISO / IEC 9126. Для шести характеристик якості – функціональності, надійності, зручності використання, продуктивності, зручності супроводу і мобільності (скорочено FRUEMP, за першими літерами англійських слів) – визначаються ознаки якості, метрики для вимірювання якості системи на основі кожної ознаки якості і спосіб переведення цих метрик в оцінки якості.

Методика вартості виявлення проблеми. Визначаються потенційні проблеми та їх вплив на систему, а потім оцінюється вартість проблем для бізнесу і ймовірність їх виникнення.

Аналіз видів помилок і їхнього впливу. Визначаються потенційні проблеми, оцінюється їхній вплив (на замовників і користувачів), а потім виконується класифікація (з урахуванням думки зацікавлених осіб) за серйозністю (впливу на систему), пріорітетністю (впливу на бізнес) і ймовірністю виникнення дефекту.

Всі наведені методики дозволяють здійснити оцінку ризиків якості з різним ступенем деталізації. Найбільш легкою в застосуванні, тією, яка не вимагає великих знань і досвіду оцінки ризиків є методика неформального аналізу. Методики аналізу ризиків якості незалежно від їх складності та деталізації аналізу використовуються тестерами для відокремлення найбільш значущих ризиків якості від менш значущих. Внаслідок цієї обставини методики оцінки ризиків якості повинні застосовуватися за зростаючою деталізацією відповідно до вимог етапу планування.

Для завершення етапу планування повинно бути розроблено і узгоджено план тестування. Він є головною частиною процесу тестування, структура якого виглядає наступним чином:

- ❒ цілі – тут визначаються цілі кожної фази тестування;
- ❒ критерії завершення – вказується критерій, за яким фіксується завершення кожної фази тестування;
- ❒ графіки робіт – календарні графіки робіт складаються для кожної фази тестування і відображають фоки проектування, кодування і виконання тестів;
- ❒ відповідальні – тут вказуються прізвища осіб, які будуть під час кожної фази проектувати, кодувати, виконувати і перевіряти результати прогону тестів, а також тих, хто буде виправляти знайдені помилки;
- ❒ бібліотеки тестів і стандарти – у великих проектах необхідні систематичні методи ідентифікації, написання і зберігання тестів;

- неправильного розташування діаграм, ілюстрацій;
- неправильності чи незрозумілості відомостей, навчальних посібників (tutorial), інструкцій;
- некоректності опису системних вимог;
- некоректності документації для завантаження (PDF, HTML або текстових файлів).

Будь-які приклади, наведені в документації, оформлюються як тест і подаються для перевірки на вхід програми.

Ризик використання недокументованих обсягів пам'яті - використання системою обсягів пам'яті, що не задовольняють вимоги тих систем, в яких дана система буде застосовуватися.

Ризик виникнення помилок установки - помилки, що виникають при установці системи в наступних умовах:

- нова установка, нова машина, на якій жодного разу не встановлювався цей програмний продукт;
- перевстановлення на машину, на якій була встановлена та ж версія даного програмного продукту;
- перевстановлення на машину, на якій була встановлена більш рання версія даного програмного продукту.

Таке категорування (рис. 4.1) дозволяє ранжувати помилки в залежності від їх впливу на якість продукту, що випускається, що дозволяє акцентувати увагу на найбільш значущих ризиках.

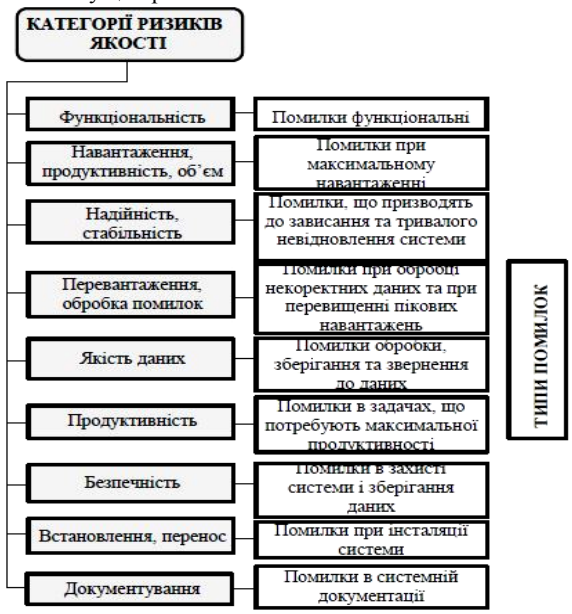


Рис. 4.1. Відповідність категорії ризику якості видам помилок

Вимога введення пін-кода при запуску – корисниймеханізм забезпечення безпеки, тому що він виступає другим чинником перевірки достовірності («щось, що ви знаєте»). Проте, є і інша сторона медалі. По-перше, необхідно регулярно міняти пін-код. В організаціях, де **BitLocker** використовувався з ОС Windows 7 і Windows Vista, користувачам доводилося звертатися до системних адміністраторів, щоб відновити пін-код або пароль **BitLocker**. В результаті не лише росли витрати на управління, але і користувачі не бажали міняти пін-код або пароль **BitLocker** досить часто.

Користувачі Windows 10 можуть оновлювати свої пін-коди і паролі **BitLocker** самостійно, облікові дані адміністратора для цього не потрібні. Це не лише дозволяє понизити витрати на підтримку, але і підвищити безпеку, тому що заохочується регулярна зміна пін-кодів і паролів користувачами. Крім того, для запуску пристроїв з **InstantGo** не вимагається пін-код: вони не призначені для регулярного запуску, тому реалізовані інші заходи для зменшення поверхні для атак в системі.

2.1.3 Стійкість до шкідливого програмного забезпечення

Опір шкідливому ПЗ включає архітектурні зміни, які можуть ізолювати ключові системні компоненти і компоненти безпеки і захистити їх від загроз. Декілька нових функцій Windows 10 допомагають понизити риски, пов'язані з шкідливим ПЗ, включаючи **VBS**, **Device Guard**, **Microsoft Edge** і абсолютно нову версію **Захисника Windows**. Крім того, багато функцій захисту від шкідливого ПЗ з ОС Windows 8.1, включаючи контейнери **AppContaine**r для ізоляції застосувань і численні функції захисту ОС при запуску (наприклад, надійне завантаження), перенесені в Windows 10 і вдосконалені в новій системі.

Windows 10 забезпечує надійний захист від шкідливого ПЗ, тому що використовує надійне апаратне забезпечення, що гарантує безпечний запуск і що захищає базову архітектуру ОС і робочий стіл.

Загрози і рішення, реалізовані в Windows 10

Загроза	Рішення Windows 10
Комплекти завантаження вбудованого ПЗ (буткити) замінюють його шкідливим ПЗ	Усі сертифіковані ПК оснащені **UEFI** з технологією безпечного завантаження, яка вимагає використання підписаного вбудованого ПЗ для оновлення **UEFI** і **Option ROM** (додатковий ПЗП).
Буткити запускають шкідливе ПЗ до запуску Windows	Безпечне завантаження **UEFI** перевіряє цілісність завантажувача ОС Windows, щоб переконатися, що ніяка шкідлива ОС не запущена до запуску Windows
Системні руткити або драйвери руткитів запускають шкідливе ПЗ на рівні ядра під час запуску Windows, до запуску **Захисника Windows** і рішень, що захищають від шкідливого ПЗ	Система надійного завантаження Windows перевіряє завантажувальні компоненти Windows; драйвери Майкрософту і драйвер захисту від шкідливого ПЗ **ELAM**, який перевіряє сторонні драйвери. Паралельно надійному завантаженню виконується вимірюване завантаження, яке надає видаленому серверу інформацію про стан завантаження при-

	строю і гарантує успішну перевірку системи модулем надійного завантаження і іншими завантажувальними компонентами.
Шкідливе ПЗ рівня користувача використовує уразливість в системі або застосуванні і вступає у володіння пристроєм	Удосконалення в області технологій **ASLR**, **DEP**, архітектура куп і алгоритмів управління пам'яттю знижує вірогідність успішного захоплення системи з використанням уразливостей. Технологія захищених процесів ізолює недовірені процеси один від одного і від важливих компонентів ОС. **VBS** на основі **Microsoft Hyper-V** захищає секретні процеси Windows від ОС Windows, ізолюючи їх від процесів, що виконуються в призначеному для користувача режимі, і ядра Windows. Цілісність коду, що налаштовується, забезпечує реалізацію політик адміністрування, що вказують, які саме застосування можуть виконуватися в призначеному для користувача режимі. Запускати інші застосування заборонено.
Користувачі викачують небезпечне ПЗ (наприклад, застосування, яке здається нормальним, але містить вбудований троянський вірус) і запускають його, не усвідомлюючи риски	Функція репутації застосувань за даними **SmartScreen** є частиною базової ОС; **Microsoft Edge** і **Internet Explorer** можуть використовувати цю функцію, щоб попередити користувачів або заблокувати скачування і запуск потенційно шкідливого ПЗ
Шкідливе ПЗ використовує уразливість в надбудові браузеру	**Microsoft Edge** – цеуніверсальне застосування, яке не запускає старі двійкові розширення, у тому числі **Microsoft Active X** і допоміжні об'єкти браузеру, які часто використовуються на панелях інструментів. Тим самим вказані риски виключаються
Веб-сайт з шкідливим кодом використовує уразливість в **Microsoft Edge** і **IE** для запуску шкідливого ПЗ на клієнтському ПК	У **Microsoft Edge** і **IE** реалізований розширений захищений режим, що використовує пісочниці **AppContainer** для захисту системи від уразливостей, які можуть виявлятися в розширеннях (наприклад, **Adobe Flash**, **Java**), що запускаються в браузері, або самому браузері

UEFI (*Unified Extensible Firmware Interface*) –інтерфейс розширюваної прошивки–цестандартизоване рішення, що є сучасною заміною для BIOS і забезпечує функцію безпечного завантаження ОС.

UEFI виконує внутрішні перевірки цілісності, що дозволяє перевірити цифровий підпис вбудованого ПЗ до запуску. Оскільки тільки виготівник устаткування комп'ютера має доступ до цифрового сертифікату, необхідного для створення дійсного підпису вбудованого ПЗ, UEFI може запобігти роботі буткитов, заснованих на вбудованому ПЗ.

Ризик порушення працездатності – помилки, що виникають при функціонуванні системи в наступних високооб'ємних умовах*:*

- максимальну (дійсну чи фізично доступну) кількість клієнтів з'єднано (або змітовано), причому всі виконують одні й ті ж дії;
- максимальний розмір бази даних досягнуто і одночасно виконується безліч запитів / повідомлень.

Стресовий ризик – помилки, що виникають при номінальних або максимальних навантаженнях, коли із системою працює велика кількість користувачів, що в принципі може призводити до різних конфліктів.

Ризик порушення зручності експлуатації – прояв психологічних (користувацьких) проблем або проблем зручності експлуатації.

Ризик порушення призначеного для користувача інтерфейсу – помилки, що виникають при взаємодії користувача із системою: помилки доступу і навігації через функції додатку, помилки функціонування об'єктів всередині призначеного для користувача інтерфейсу.

Ризик порушення захисту – помилки, що виникли в області функціональної/інформаційної безпеки. Існування/виникнення можливості доступу користувача до тих функцій інформації, на які даний тип користувача не має права.

Ризик конфігурацій обладнання – помилки, що виникають в роботі системи при різних програмних і апаратних конфігураціях. У більшості середовищ певні апаратні специфікації для клієнтських робочих станцій, мережевих з'єднань і серверів баз даних змінюються. Клієнтські робочі станції можуть бути навантажені різними програмами (додатками, драйверами і т.д.), і в будь-який конкретний час безліч різних комбінацій можуть бути активними і використовувати різні ресурси.

Ризик несумісності (конверсії) – помилки, які проявляються при сумісності з існуючою системою і процедурах конверсії процесу переходу від одного методу обробки даних до іншого.

Ризик виходу з ладу/відновлення – помилки в процесах (ручних або автоматичних), які відновлюють базу даних, додатки і систему при виникненні наступних умов:

- відключення живлення на стороні клієнта;
- відключення живлення на сервері;
- порушення зв'язку через мережевий(і) сервер(и);
- незакінчені цикли (переривання процесів фільтрації даних, переривання процесів синхронізації даних);
- невірний покажчик/ключ в базі даних;
- невірні/пошкоджені елементи даних в базі даних.

Ризик існування помилок в документації - виявлення існуючих в призначеній для користувача документації наступних помилок:

- неточності будь-яких тверджень;
- некоректності діаграм, ілюстрацій;

- графік розробки тестів для кожного з компонентів тестованого продукту;
- потреби в тимчасових і людських ресурсах, а також у фінансових коштах, на проведення тестування в напередодні визначених рамках;
- терміни проведення тестування, як в цілому, так і окремо по кожному етапу тестування;
- ризики якості програмного продукту;
- обсяг і охоплення тестування;
- план заходів і склад учасників.

Оцінити витрати на тестування можна тільки попередньо оцінивши ризики якості, які незмінно супроводжують тестування програмного продукту. Самі ж ризики якості попередньо слід розділити відповідно до методів тестування, як це буде наведено нижче.

Під ризиком якості тестованого програмного продукту розуміють потенційний вид помилки, спосіб поведінки системи, при якому вона, ймовірно, не відповідає обгрунтованим очікуванням якості системи, що є у користувача або замовника.

Ми ж пропонуємо розглянути інший підхід до питання про визначення поняття ризику якості. На нашу думку, ризик якості можна визначити як добуток виникнення виду помилки на вартість усунення виявленої помилки.

Ризик якості можна категоризувати в залежності від видів помилок, які є невід'ємною частиною тестованої системи і які виявляються в процесі її тестування. Нижче наведено список категорій ризику якості з відповідними їм видами помилок.

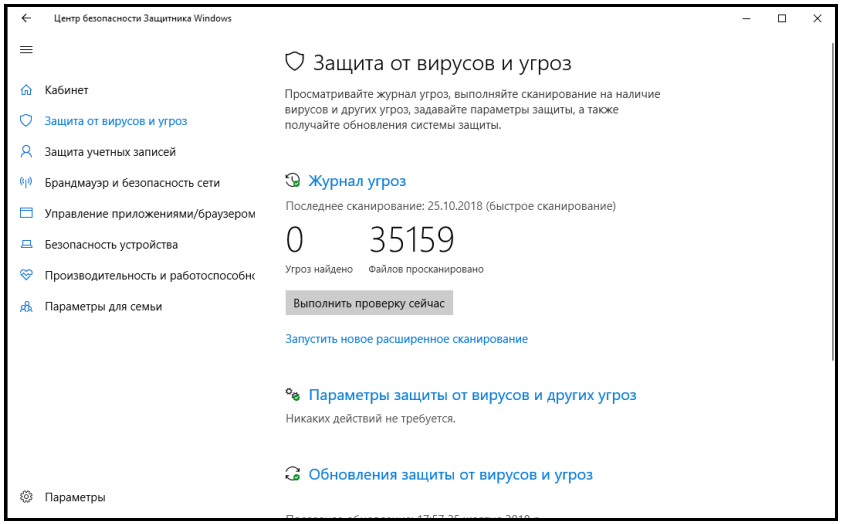
Інтеграційний, або інтегральний, ризик - помилки, що виникають в процесі інтеграції програмних компонентів або модулів, в тому числі дефекти, що порушують їх взаємодію.

Ризик порушення функціональності - помилки, що виникають при прийнятті даних, їх обробці і виведенні і в реалізації вимог.

Ризик порушення зручності використання - помилки у виконанні пунктів технічного завдання та виконанні системою поставленого завдання. Такі помилки часто можуть бути допущені без використання комп'ютера.

Ризик порушення циклу активності - помилки, що виникають в процесах системи протягом певного проміжку часу і призводять до такого функціонування системи, що не задовольняє необхідну модель. Такий часовий проміжок слід визначити (наприклад, один рік), після чого транзакції і дії, які будуть відбуватися протягом року (всі щоденні, щотижневі, щомісячні цикли і події, чутливі до даних), повинні бути виконані.

Ризик зменшення продуктивності - помилки, що виникають при різних заданих відомих навантаженнях, які визначаються в термінах числа користувачів, обсягу даних і швидкості їх обробки (часу відгуку). Проводиться окремо від функціонального тестування, на спеціальному обладнанні (стенді).



Забезпечення безпеки на основі віртуалізації

Одна з найістотніших змін, реалізованих в Windows 10, – цезабезпечення безпеки на основі віртуалізації(**VBS**, *Virtualization-based Security*) має на увазі використання можливостей системи віртуалізації ПК і є передовим способом захисту системних компонентів від загроз. VBS може ізолювати найуразливіші і важливіші компоненти системи безпеки Windows 10. Ці компоненти системи безпеки не просто ізолюються за допомогою обмежень **API** або проміжного рівня: вони виконуються в іншому віртуальному середовищі і ізольовані від самої ОС Windows 10.

Завдяки VBS реалізовано два важливі удосконалення в системі безпеки Windows 10: нова межа довіри між ключовими системними компонентами Windows і безпечне середовище виконання, в якому вони виконуються. Межа довіри між ключовими системними компонентами Windows реалізується завдяки віртуалізації платформи в середовищі VBS для ізоляції середовища VBS від ОС Windows.

Захист робочого столу Windows

У Windows 10 реалізовані важливі удосконалення в ядрі Windows і настільного середовища, де найчастіше трапляються атаки і куди найчастіше потрапляє шкідливе ПЗ. Настільне середовище стало стійкішим до шкідливого ПЗ завдяки значним удосконаленням в **Захиснику Windows** і фільтрах **SmartScreen**. Переглядати сторінки в Інтернеті також стало безпечніше завдяки абсолютно новому браузеру **Microsoft Edge**. **Магазин Windows** також сприяє зниженню ймовірності проникнення шкідливого ПЗ на пристрої: усі застосування, що потрапляють в екосистему **Магазину Windows**, ретельно

тестуються і тільки потім стають доступні користувачам. Універсальні застосування Windows спочатку безпечніші, ніж типові застосування, оскільки вони ізольовані. Ізоляція знижує ризик злому застосування або виконання з ним яких-небудь дій, що ставлять під загрозу систему, дані та інші застосування.

Microsoft Edge.Безпека браузеру є найважливішим компонентом будь-якої стратегії безпеки і не даремно: браузер – це інтерфейс користувача в Інтернеті, а Інтернет буквально кишить шкідливими сайтами і вмістом, який чекає своєї години, щоб заразити комп'ютер. Внаслідок цього браузер став каналом № 1 для атак хакерів і порушників.

У Windows 10 представлений абсолютно новий браузер – MicrosoftEdge. Microsoft Edge безпечніший, ніж попередники. Це обумовлено наступним:

❒ Microsoft Edge не підтримує сторонні двійкові розширення. Він підтримує вміст Flash і перегляд PDF за замовчуванням завдяки вбудованим розширенням, проте ніякі інші двійкові розширення (включаючи елементи управління **ActiveX** і **Java**) не підтримуються.

❒ Microsoft Edge виконує тільки 64-розрядні процеси, що забезпечує підвищену безпеку при виявленні уразливостей і спробі використовувати їх.

❒ Microsoft Edge розроблений як універсальне застосування Windows. Він розділений на частини і виконується в контейнері **AppContainer**, який ізолює браузер від системи, даних та інших застосувань.

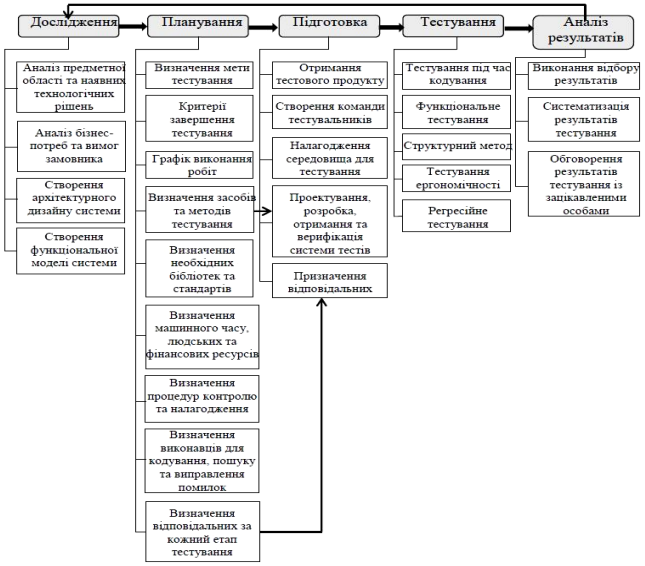
❒ Microsoft Edge спрощує завдання по налаштуванню системи безпеки. Оскільки в Microsoft Edge використовується спрощена структура застосування і єдина конфігурація пісочниці, число необхідних налаштувань безпеки значно менше. Крім того, налаштування Microsoft Edge за замовчуванням відбивають передові практики в області безпеки. Це застосування безпечне за замовчуванням.

Фільтр SmartScreenдопомагає захищати користувачів від шкідливих застосувань і веб-сайтів за допомогою застосування **SmartScreen** і служб репутації URL-адрес. Якщо застосування або URL-адрес не є гарантовано безпечним, фільтр SmartScreen попереджає користувача або навіть блокує скачування застосування.

Захисник Windows забезпечує надійніший захист від шкідливого ПЗ завдяки чотирьом чинникам: об'ємному локальному контексту, усеосяжним глобальним датчикам, захисту від злому і наданню фахівцям з IT-безпеки ширших можливостей.

Об'ємний локальний контекст підвищує ефективність виявлення шкідливого ПЗ. Windows 10 інформує **Захисник Windows** не лише про вміст (файлах і процесах), але і джерела цього вмісту, місця зберігання і т.д. Наприклад, застосування, викачане з Інтернету, перевірятиметься ретельніше, ніж застосування, встановлене з довіреного сервера. Windows 10 зберігає історію застосування з Інтернету на рівні ОС, щоб застосування не могло стерти свої сліди. Історія відстежується і зберігається в сховищі Persisted Store, який надійно

ня, аналізу та контролю виявлених дефектів, виконання аналізу ризиків, постійного поліпшення тестування і власне процесу розробки.



Алгоритм процесу тестування
криптографічних програмних продуктів

Цей етап тестування слідує безпосередньо після (або в деяких випадках може починатися під час) розробки і обігравання еталонної моделі створюваної системи. У процесі планування беруть участь всі особи, зацікавлені в розробці системи. На цьому етапі в плані тестування визначаються і документально фіксуються такі елементи процесу тестування:

- цілі;
- завдання;
- системи і підсистеми тестування;
- бажана якість системи;
- вимоги, що пред'являються до тестування;
- стандарти і основні напрямки створення процедур тестування;
- апаратні, програмні та мережеві засоби, необхідні для підтримки тестового середовища;
- вимоги до даних для тестування;
- попередній план-графік тестування;
- вимоги щодо оцінки продуктивності;
- процедури управління конфігурацією і середовищем тестування;
- процедури відстеження дефектів і засоби для їхнього проведення;

Далі ми докладніше зупинимося на розгляді основних етапів тестування пропонованої комплексної методики. Окремо розглянуті етапи тестування програмної криптографічної системи в сукупності дають повне уявлення про процес тестування і про проведені заходи на кожному з етапів.

Основні принципи процесу тестування:

- процес тестування повинен починатися в момент виникнення задуму і тривати протягом усього циклу розробки програми;
- в процесі тестування необхідне застосування комбінованого підходу (використання декількох методів тестування).

Це дозволяє:

- зменшити загальні витрати на тестування;
- підвищити надійність системи;
- дотримуватися виконання планів і графіків роботи.

Дослідження

На етапі дослідження проводиться концептуалізація розроблюваної системи, що включає в себе наступні заходи:

- аналіз предметної області, а також наявних технологічних рішень бізнес-аналітиками;
- аналіз потреб і вимог замовника системними архітекторами, а також пропозиція варіанта реалізації завдання засобами інформаційних технологій, бізнес-технологій;
- створення архітектурного дизайну системи;
- створення функціональної моделі системи.

Цей етап дослідження є найбільш важливим, так як під час його виконання визначається і формується модель еталонної системи. Ця модель створюється з урахуванням зробленого аналізу інформації, отриманої в результаті перерахованих вище заходів. Якщо дана модель не буде спроектована, подальше тестування стане неможливим, так як не буде з чим порівнювати розроблювану систему.

Спроектована модель еталонної системи обігрується для виявлення помилок на найбільш ранній стадії. Це дозволяє вже на самому початку розробки системи виявити її слабкі сторони і найбільш чітко сформулювати вимоги до неї. Висунуті вимоги повинні пред'являтися не до всієї системи в загальному, а так, щоб їх можна було перевіряти в процесі всього циклу розробки. Ці вимоги поділяються на функціональні (які функції і наскільки якісно повинно реалізовувати ПЗ) і нефункціональні (обмеження на час вирішення завдання, швидкість доступу до даних, вимоги до займаних ресурсів і т. п.). В ході розробки сформульовані вимоги повинні бути доступні всім учасникам процесу розробки для перевірки їх виконання.

Планування тестування

Планування необхідне для визначення і рівномірного розподілу виділених ресурсів (як фінансових, так і людських), управління процесом тестуван-

управляє об'ємним локальним контекстом і захищає від несанкціонованої зміни і видалення даних.

Усеосяжні глобальні датчики допомагають підтримувати Захисник Windows в актуальному стані і інформують його навіть про найновіші шкідливі програми. Це досягається двома способами: збором даних про об'ємний локальний контекст з кінцевих точок і централізованим аналізом цих даних. Мета - виявити нове, передове шкідливе ПЗ і заблокувати його в перші (найважливіші) години життя, щоб запобігти поширенню по ширшій екосистемі ПК.

Наявність Захисника Windows дозволила Майкрософт представити систему хмарного захисту **Windows Defender Cloud Protection**, яка допомагає краще реагувати на зміни в динамічному ландшафті шкідливого ПЗ. Мета – блокувати шкідливе ПЗ при першій появі, протягом перших декількох, найважливіших годин атаки.

Звичайно, системний адміністратор має централізований контроль над усіма параметрами Захисника Windows через групову політику.

У Захиснику Windows реалізований захист від злому; він використовує декілька технологій безпеки, доступних в Windows 10, основний з яких є захищені процеси, що запобіжуть зміні компонентів Захисника Windows, розділів реєстру і т.д. недовіреними процесами.

Розширення можливостей для IT-фахівцівз безпеці має на увазі, що Захисник Windows надає IT-фахівцяміструменти і параметри конфігурації, необхідні для створення висококласного антишкідливого рішення. Він володіє безліччю висококласних функцій, що дозволяє поставити його в один ряд з провідними продуктами цієї категорії :

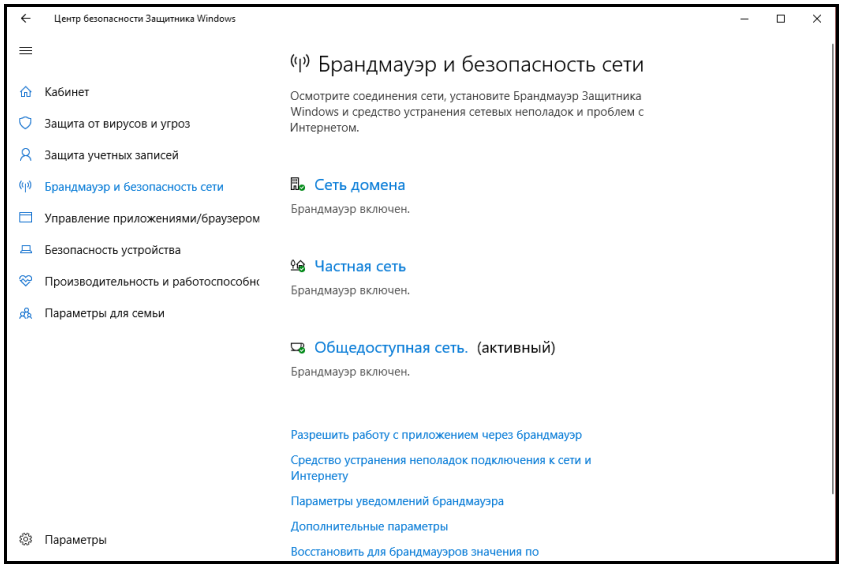
Захисник Windows підтримує стандарт управління пристроями **Open Mobile Alliance**, що дозволяє здійснювати централізоване управління багатьма сторонніми рішеннями для управління пристроями.

Брандмауер Windows є фільтром мережевих пакетів, здатним відбити стандартні мережеві атаки і не допустити низькорівневого мережевого підключення до захищеного комп'ютера. Він допомагає при боротьбі з вірусами і хробаками, що намагаються потрапити в комп'ютер через Інтернет, зберігаючи можливість роботи з електронною поштою і веб-сервісами.

Брандмауер має заздалегідь визначений набір правив для обох типів трафіку. Самі правила можуть бути відредаговані і змінені користувачем і програмним забезпеченням, яке користувач встановлює на ПК.

За замовчуванням замовчуванням він дозволяє користувачам робити декілька речей: серфінг в Інтернеті, використовувати служби миттєвих повідомлень, підключення до домашньої групи, спільне використання файлів, папок і пристроїв і т.д.

Правила застосовуються по-різному, залежно від налаштування мережі профілю для поточного мережевого з'єднання.



2.2 Дозволи, облікові записи і профілі користувачів

2.2.1 Дозволи NTFS

Розвинена система безпеки – це один з «козирів» ОС MS Windows 10. Основою цієї безпеки служить файлова система **NTFS**, яка припускає використання дозволів.

Дозвіл NTFS – це правило, пов'язане з об'єктом (файлом, папкою) і використовуване для управління доступом користувачів до цього об'єкту.

При цьому під **користувачем** розуміється не лише *користувач-людина*, але і *програми*, запущені від його імені (під його обліковим записом).

Передбачений як **стандартний набір дозволів**(для загальних випадків), так і **спеціалізований набір**– для «тонкого» налаштування.

Стандартні дозволи NTFS для папок

Дозвіл	Дії, що допускаються
Читання (*Read*)	Дозволяється переглядати вкладені папки і файли, а також їх властивості
Запис (*Write*)	Дозволяється створювати і розміщувати усередині папки нові файли і підпапки, а також змінювати атрибути папки і переглядати її властивості
Список вмісту папки (*List folder contents*)	Дозволяється переглядати імена файлів, що містяться в папці, і вкладених підпапок
Читання і виконання (*Read&Execute*)	Дозволяється дістати доступ до файлів в підпапках, навіть якщо немає доступу до самої папки. Крім того дозволя-

виконання замовником заздалегідь визначеного набору тестових ситуацій, що імітують типові умови, в яких програма буде працювати після введення в експлуатацію. Заключним етапом приймальних випробувань є установча перевірка, за умовами якої завершена версія програмного продукту встановлюється на об'єктах замовника з метою отримати від нього підтвердження, що програмний продукт відповідає всім вимогам.

Тестування ергономічності

Цей вид тестування є частиною процесу створення необхідних умов для «зручності користувача». Він включає в себе набір методів, які дозволяють тестерам перевірити програмний продукт на предмет зручності його використання кінцевим користувачем. Типовий тест на ергономічність полягає в тому, що користувачі виконують з прототипом (або іншою програмою) ряд операцій, в той час як спостерігачі документують все, що вони роблять і говорять. Таке тестування проводиться одночасно з одним або декількома користувачами, що працюють разом.

Тестування може полягати в зборі інформації:

- про послідовність дій, що здійснюються користувачем в процесі виконання завдання;
- про помилки, які вони здійснюють;
- про те, коли і в чому вони проявляють невдоволення;
- наскільки швидко вони виконують операції;
- наскільки вони успішні у виконанні цих операцій;
- наскільки вони задоволені роботою з даним продуктом.

Мета більшості тестів на ергономічність полягає в тому, щоб виявити будь-які проблеми, з якими може мати справу користувач, і усунути їх.

Уважне вивчення розглянутих методів тестування показує, що вони доповнюють один одного - різні методи дозволяють знаходити різні помилки. Відповідно, найбільш ефективними методиками тестування при розробці криптографічного програмного забезпечення є ті з них, які містять елементи цих методів, використовуючи позитивні сторони кожного з них. Зазвичай розробляється комплект тестів відповідно до критеріїв будь-якого з методів «чорного ящика», а потім отриманий комплект доповнюється тестами для перевірки найбільш складних вузлів і логіки програми. Такий підхід дозволяє протестувати програмний продукт в повному обсязі і дозволяє виявити найбільшу кількість помилок, допущених в тестованому програмному продукті.

4.3.2 Методологія тестування

Методологія тестування криптографічних програмних систем

Практичне застосування даного підходу до тестування у багато разів скорочує витрати як на оцінку якості програмного продукту, так і на пошук і виправлення його дефектів.

перше, таким способом неможливо знайти помилки, які взаємознищуються, по-друге, деякі помилки виникають досить рідко (помилки роботи з пам'яттю) і тому їх важко знайти і відтворити, по-третє, складно перевірити відповідність конкретного програмного продукту його специфікації і т.д .Основною ж перевагою тестування за методом «чорного ящика» є його об'єктивність. Потенційні помилки, виявлені методом «білого ящика», вимагають практичного підтвердження, що вони дійсно є помилками і що існує шлях реалізації цих помилок (тобто помилка не відсікається на будь-якому іншому рівні перевірок). Найчастіше це простіше виконувати саме функціональними методами, як у випадку застосування методу «чорного ящика».

Метод регресивного тестування

Обидва вищеописані процеси можуть слугувати основою регресивного тестування. Це метод повторного тестування зміненого програмного продукту з метою перевірки на відсутність помилок в попередньо нормально працюючих функціях, які були б викликані виправленням виявлених раніше помилок або додаванням в програму нових функціональних можливостей. Зберігши тестові набори для «чорного» і «білого» ящиків, можна використовувати їх для регресивного тестування, здійснюючи контроль цілісності коду в міру того, як він модифікується. При виконанні регресивного тестування можна відразу ж після зміни тексту визначати, чи не з'явилися нові помилки, і усунути їх негайно після виникнення, перешкоджаючи тим самим їх поширенню. Як правило, цей метод має на увазі повторне виконання всіх раніше використаних процедур тестування. Даний метод дозволяє повністю гарантувати те, що програмне забезпечення працює саме так, як планувалося.

Регресивне тестування має велике значення тому, що внесені в програму коди змін і виправлень помилок, як правило, більш схильні до спотворень, ніж вихідні коди програми. Наприклад, в книзі [2] наведені результати досліджень, проведених в одній великій компанії, що займається супроводом програм. В результаті цих досліджень з'ясувалося, що навіть однорядкова зміна програми з імовірністю 55% зберігала помилку або вносило нову.

Метод «сірого ящика»

Тестування з використанням методу «сірого ящика» (gray box) передбачає розгляд як об'єкта тестування не всієї програми в цілому, а її окремих інтерфейсів, як користувацьких, так і прикладних. У порівнянні з методикою «чорного ящика», витрати на розробку і запуск тестів тут значно вищі, тому застосування даного методу до додатків, які мають інтерфейс користувача, зазвичай обмежується тестуванням останнього, що пов'язано з частими його варіаціями. Основним недоліком даного методу є те, що програмний продукт не тестується в повному обсязі, і, отже, тільки частина помилок може бути виявлена.

Метод дослідної експлуатації

В основі тестування лежать реальні користувальницькі дії з програмою. Бета-тестування, а іноді приймально-здавальні випробування передбачають

	ються ті ж дії, що передбачено для дозволів «Читання» і «Список вмісту папки»
Зміна (*Modify*)	Дозволяються усі дії, передбачені для дозволів «Читання» і «Читання і виконання» + дозволяється видалення папки
Повний доступ (*Full control*)	Надає повний доступ до папки (дозволяються усі дії, передбачені усіма перерахованими вище дозволами). Додатково дозволяється ставати власником папки і змінювати її дозволи
Особливі дозволи (*Special Permission*)	Задає набір спеціальних дозволів, що відрізняється від стандартних

Стандартні дозволи NTFS для файлів

Дозвіл	Дії, що допускаються
Читання (*Read*)	Дозволяється читання файлу, а також перегляд його властивостей
Запис (*Write*)	Дозволяється перезапис файлу, зміна його атрибутів, а також перегляд його власника і дозволів
Читання івиконання (*Read&Execute*)	Те ж що і «Читання» + можливість запуску застосування (якщо файл виконуваний)
Зміна (*Modify*)	Дозволяється зміна і видалення файлу + те, що передбачено дозволами «Запис» і «Читання і виконання»
Повний доступ (*Full control*)	Надає повний доступ до файлу. Це означає, що дозволяються усі дії, передбачені усіма перерахованими вище дозволами. Додатково дозволяється ставати власником файлу і змінювати його дозволи
Особливі дозволи (*SpecialPermission*)	Задає набір спеціальних дозволів, що відрізняється від стандартних

Особливі дозволи NTFS для файлів і папок

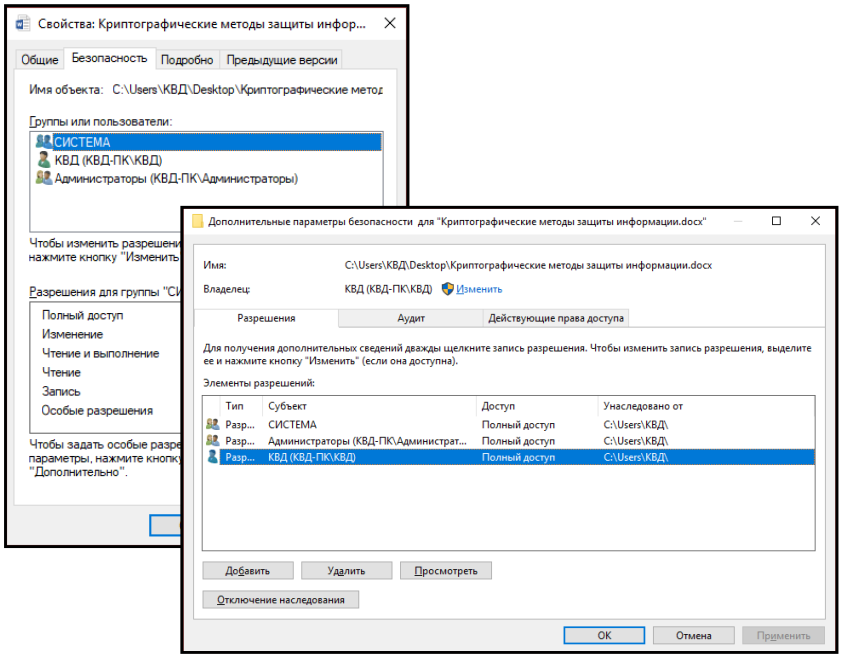
Якщо стандартні дозволи призначені для загальних випадків, для швидкого і зручного призначення прав доступу, то **спеціалізовані права доступу дозволяють підійти до цієї справи відповідальніше і скрупульозне**. При цьому можна змінити стандартний набір дозволів так, як необхідно.

До особливих дозволів відносяться наступні:

- ❑ Огляд папок / Виконання файлів
- ❑ Зміст папки / Читання даних
- ❑ Читання атрибутів
- ❑ Читання додаткових атрибутів
- ❑ Створення файлів / Запис даних
- ❑ Створення папок / Дозапис даних
- ❑ Запис атрибутів
- ❑ Запис додаткових атрибутів
- ❑ Видалення
- ❑ Видалення підпапок і файлів
- ❑ Читання дозволів
- ❑ Зміна дозволів
- ❑ Зміна власника
- ❑ Синхронізація

Призначення, перегляд і зміна дозволів виробляється на вкладці **Безпека** діалогового вікна **Властивості**. Щоб його викликати, необхідно клацнути по файлу або папці правою кнопкою миші і в контекстному меню вибрати **Властивості**.

Вже відразу на вкладці **Безпека** можна побачити, яким користувачам і групам доступ визначений і які їм надані дозволи. Інформацію про додаткові параметри доступу (у тому числі і про спеціальні дозволи) можна отримати, натиснувши кнопку **Додатково**.



2.2.2 Облікові записи і профілі користувачів

Створення **облікових записів і груп** займає важливе місце в забезпеченні безпеки Windows 10. Якщо на комп'ютері працюють декілька користувачів, то кожен з них має певні права, які задають операції, дозволені користувачеві на цьому комп'ютері.

Призначаючи права користувачам, адміністратор має можливість обмежити їх доступ до конфіденційної інформації мережі, дозволити або заборонити певні дії.

У Windows 10 для роботи з обліковими записами є оснащення **Локальні користувачі і групи** і утиліта **Облікові записи користувачів**. Права користувача призначаються шляхом додавання його в одну зі вбудованих груп, які містять набір вже призначених прав користувача. При необхідності можна створити нову групу і призначити їй певні права.

безпосередньо вам необхідно сформулювати вимоги до тестування, які будуть забезпечуватися різними наборами тестів. Кожна методика має на увазі використання наборів тестів, що дозволяють отримати необхідну інформацію відповідно до обраної методики тестування.

Ми розглянемо основні методи й методики, за допомогою яких здійснюється тестування в сучасних компаніях, що займаються розробкою криптографічних програмних систем.

Структурний метод (метод білого або скляного ящика)

Тестування методом «білого ящика» (white box) передбачає обробку програми, що тестується, як «прозорого об'єкта», на відміну від методу «чорного ящика», даний метод заснований на використанні певних знань програмного коду, необхідних для контролю коректності даних на виході. Тест є правильним тільки в тому випадку, коли фахівець, який тестує програмне забезпечення (ПЗ), знає, що конкретно повинна робити програма. До методів «білого ящика» відносяться методи покриття операторів, покриття рішень, покриття умов і метод комбінаторного покриття умов. Всі вони базуються на тому, що при виконанні тестів повинні виконуватися ті чи інші конструкції в тексті програми. Тестування методом «білого ящика» не обробляє випадкові помилки, але весь видимий код повинен бути зручним для читання. Основною складністю даної групи методів є складність відстеження часу виконання.

Поведінковий метод «чорного ящика»

Тестування методом «чорного ящика» (black box) передбачає обробку системи як «непрозорого об'єкта». Основна мета - з'ясування ситуацій, в яких поведінка програми не відповідає її специфікації. Тестові дані генеруються на основі функціональної специфікації програми, вони мають граничні значення. Тому для тестування функцій програми (у разі занадто великої кількості варіантів тестів) досить протестувати граничні значення і деякі випадкові проміжні значення вхідних даних. Таким чином, знання внутрішньої структури в явному вигляді не використовуються.

Тестування з застосуванням цього методу зазвичай має на увазі перевірку функціональних можливостей програми. Класичні приклади - це еквівалентне розбиття і тестування областей (domain testing). При тестуванні програмного забезпечення методом «чорного ящика» фахівець з тестування ПЗ знає тільки набір параметрів, що вводяться, і очікувані на виході результати. Яким чином програма досягає цих результатів, йому не відомо. Він також ніколи не перевіряє програмний код і не потребує додаткового знання програми, крім її технічного опису. До методів «чорного ящика» відносяться метод еквівалентного розбиття, аналіз граничних умов, метод функціональних діаграм.

Цей підхід вимагає найменших витрат на тестування і тому є найбільш поширеним в повсякденній практиці, але у нього є цілий ряд недоліків. По-

і **SHA-512** довжина блоку – 1024 біта). Поступові удосконалення алгоритму **SHA** ведуть до збільшення його криптостійкості.

Поява асиметричної криптографії не усуває потребу в криптографії з секретними ключами. Причина в тому, що криптографія з асиметричними ключами використовує математичні функції для шифрування і розшифрування набагато повільніше, ніж криптографія з симетричними ключами. Для шифрування великих повідомлень криптографія з симетричними ключами потрібна. З іншого боку, швидкість криптографії з симетричними ключами не усуває потреби в асиметричній криптографії, яка потрібна для встановлення достовірності цифрових підписів і роботи станцій розсилці ключів засекречування. Це означає здатність системи криптографічного захисту використовувати усі аспекти безпеки. Іншими словами, одна криптосистема доповнює іншу.

4.3 Методи тестування криптографічних програмних систем

4.3.1 Методи тестування

Методи тестування криптографічних програмних систем

Під самим тестуванням розуміється процес виконання програм з метою виявлення помилок [1].

Процедура тестування є одним з найбільш важливих компонентів оцінки якості програмного продукту і займає, як правило, майже половину від загального часу, який виділяється для розробки. В ході її реалізації відбувається виявлення більшості існуючих помилок в розроблюваному продукті.

На теперішній час широко відома оцінка розподілу затрат праці між фазами розробки програмного продукту на виявлення та виправлення дефектів в останньому: 40х20х40% (дизайн - розробка коду - тестування).

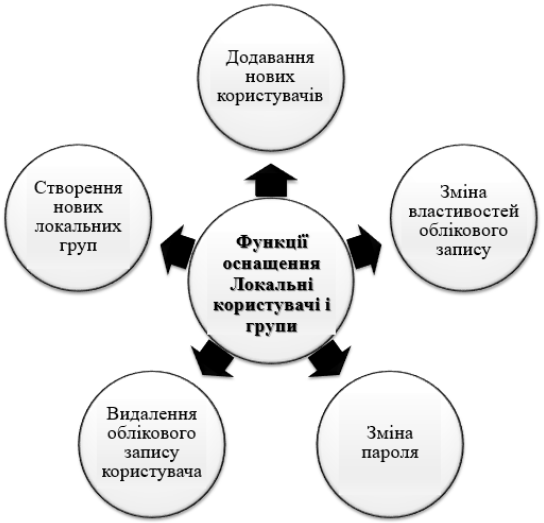
У зв'язку з постійним зростанням технічних вимог і функціональної складністю криптографічних систем процедура тестування значно ускладнилася. Як показує практика, всі помилки програми виявити неможливо, оскільки «повне» (вичерпне) тестування неможливе. Через це необхідна деяка методика, яка б дозволила розробити компактний, але досить продуктивний набір тестів, що дозволяє виявити більшість допущених помилок.

Використання певних методик і методів спрощують і систематизують процес тестування на різних його стадіях. Вони дають загальне розуміння процесу тестування, існуючого і застосовуваного сьогодні в більшості компаній, що розробляють програмне забезпечення. Так як ці методи в основному були розроблені в період початку розвитку інформаційних технологій, окремо вони не в змозі гарантувати високу якість продукту, що розробляється. Внаслідок цього в компаніях повинен використовуватися комбінований підхід у використанні цих методик.

В даний час розроблено і застосовується безліч методів і методик, які використовують різні підходи до процесу тестування. Для вибору потрібних

Оснащення Локальні користувачі і групи

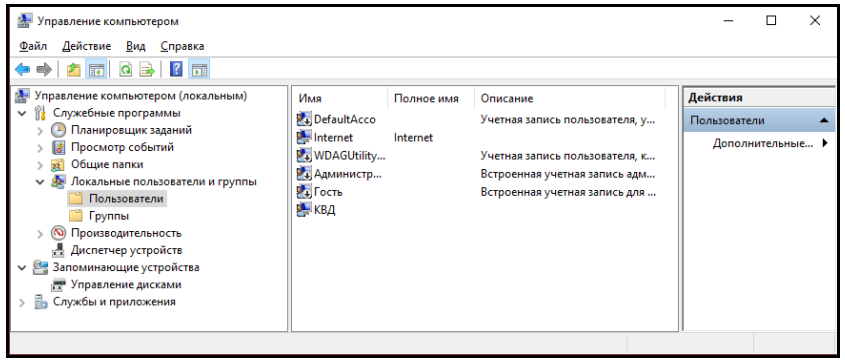
За її допомогою можна управляти локальними обліковими записами користувачів і груп як на локальному, так і на видаленому комп'ютері. Адміністрування користувачів за допомогою оснащення може виконувати тільки адміністратор.



Після установки MS Windows 10 в папці**Користувачі** розміщуються автоматично створювані облікові записи:

Обліковий запис	Призначення
Адміністратор	Використовується при установці і налаштуванні робочої станції або сервера, що є членом домена. Цей обліковий запис не може бути видалений або заблокований
Гість	Застосовується для реєстрації в комп'ютері без спеціально створеного облікового запису і введення пароля. Цей запис за замовчуванням заблокований і є членом групи **Гості**

Виклик:**Пуск ⇒ Панель управління ⇒ Адміністрування ⇒ Управління комп'ютером ⇒ Локальні користувачі і групи**



У процесі роботи створюються інші групи користувачів, яким надають різні права.

Права, що надаються групам користувачів

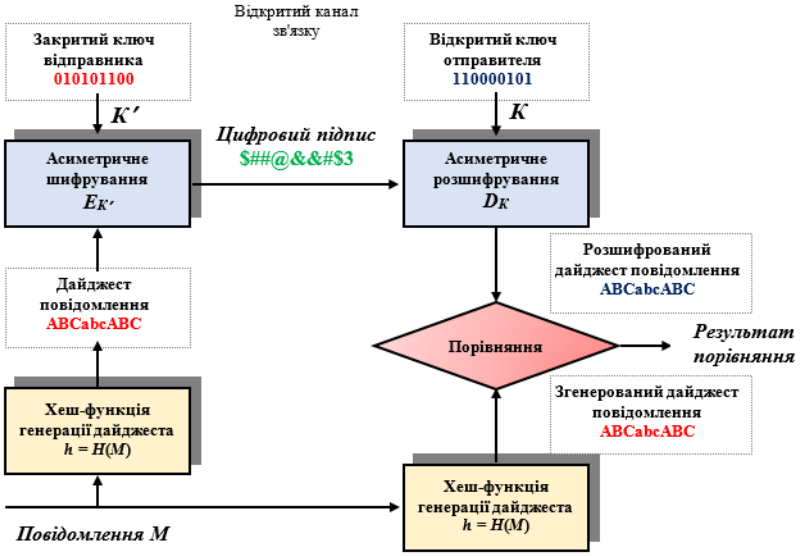
Група	Права
IIS_IUSRS	Група, використовувана службами **Internet Information Services (IIS)**
Адміністратори	Користувачам цієї групи надаються усі права і можливості в системі. Вони мають доступ до усіх ресурсів системи, право встановлювати програми і устаткування, вносити зміни на рівні системи, створювати, змінювати і видаляти облікові записи користувачів, переглядати усі загальні файли
Гості	Група призначена для запуску комп'ютера разовими користувачами. Членові цієї групи надаються обмежені можливості
Криптографічні оператори	Користувачам дозволено виконувати операції криптографії
Оператори архіву	Члени групи можуть архівувати і відновлювати файли на комп'ютері, незалежно від усіх дозволів, встановлених для цих файлів, входити в систему і завершувати її роботу. Члени групи не можуть змінювати налаштування безпеки
Оператори налаштування мережі	Членам групи надані деякі права з налаштування мережевих служб і параметрів
Оператори допомоги з контролю	Члени групи можуть видалено запрошувати атрибути авторизації і дозволу для ресурсів на цьому комп'ютері
Досвідчені користувачі	Категорія користувачів залишена для сумісності з попередніми версіями ОС і має обмежені адміністративні права.Досвідчені користувачі можуть створювати локальні групи і облікові записи користувачів, а також видаляти користувачів з локальних груп, які вони створили, змінювати і видаляти створені ними облікові записи. Вони можуть управляти додаванням і видаленням користувачів з груп **Досвідчені користувачі**, **Користувачі** і **Гості**. Вони не мають прав на архівацію і відновлення каталогів, завантаження і вивантаження драйверів пристроїв або управління журналами безпеки і аудиту
Користувачі	Члени цієї групи можуть виконувати найбільш поширені завдання, наприклад запуск застосувань, друк документів, копіювання файлів і т.п. Користувачі мають право створювати локальні групи і змінювати створені ними групи. Вони не можуть організовувати загальний доступ до ресурсів комп'ютера
Користувачі DCOM	Членам групи надані права з запуску, активації і використання об'єктів DCOM на цьому комп'ютері
Користувачі журналів продуктивності	Група призначена для управління журналами і сповіщеннями на локальному або видаленому комп'ютері
Користувачі системного монітора	Члени цієї групи можуть здійснювати моніторинг комп'ютера локально і видалено

з'явився на початку 90-х років XX століття, його автор – Р. Ривест (*R. Rivest*). В результаті використання **MD5** для довільного повідомлення формується 128-бітове хеш-значення. Вхідні дані обробляються блоками по 512 біт. В алгоритмі використовуються елементарні логічні операції (інверсія, кон'юнкція, складання за модулем 2, циклічні зрушення та ін.), а також звичайне арифметичне складання. Комплексне повторення цих елементарних функцій алгоритму забезпечує те, що результат після обробки добре перемішаний. Тому мало ймовірно, щоб два повідомлення, вибрані випадково, мали б однаковий хеш-код. Проте в 1996 р. алгоритм був практично зламаний німецьким криптографом Хансом Доббертином. У нім були виявлені настільки серйозні слабкі місця, що його використання на практиці є дуже ризикованим.

Алгоритм SHA (*Secure Hash Algorithm* – безпечний хеш-алгоритм) був розроблений національним інститутом стандартів і технології (NIST) США і опублікований як американський федеральний інформаційний стандарт **DSS** (*Digital Signature Standard*) в 1993 році.

SHA-1 формує 160-бітове хеш-значення на основі обробки вихідного повідомлення блоками по 512 біт. Використовуються прості логічні і арифметичні операції.

Структура криптосистеми цифрового підпису



У 2001 р. NIST прийняв як стандарт три хеш-функції з більшою довжиною хеш-коду, чим у **SHA-1**. Часто ці хеш-функції називають **SHA-2** або **SHA-256**, **SHA-384** і **SHA-512** (у назві вказується довжина створюваного алгоритмами хеш-коду). Ці алгоритми відрізняються не лише завдовжки створюваного хеш-коду, але і використовуваними внутрішніми функціями і завдовжки оброблюваного блоку (у **SHA-256** довжина блоку – 512 біт, а у **SHA-384**

$$\text{SIGN}(M,K') = D_{K'}(M) \text{ і } \text{CHECK}(K,M,S) = \begin{cases} 1, & \text{якщо } E_K(M); \\ 0, & \text{у протилежному випадку.} \end{cases}$$

Очевидно, що при передачі відкритого тексту ***M***, який не треба шифрувати, а вимагається тільки приєднати до нього цифровий підпис, розглянутий підхід не є оптимальним, оскільки довжина підпису ***S*** дорівнює довжині відкритого тексту ***M***, тобто час передачі інформації каналом зв'язку збільшується удвічі.

Для усунення цього недоліку використовуються так звані *хеш-функції*.

Хеш-функція*H*(*M*) – це математична або інша функція (алгоритм), яка з вихідного тексту ***M*** довільної довжини формує деяке ціле значення або деякий рядок фіксованої довжини $h = H(M)$ (так званий *дайджест*) (наприклад, 64, 128 або 160 біт).

Криптографічна хеш-функція ***H*(*M*)**, що формує дайджест, повинна задовольняти наступним **вимогам**:

1) бути швидко обчислюваною для будь-якого повідомлення ***M***;

2) при відомому повідомленні ***M*** повинно бути важко знайти інше повідомлення ***M'*** з таким же значенням хеш-функції, як у вихідного повідомлення;

3) не повинна призводити до колізій, тобто для різних повідомлень M_1 і M_2 не повинна виконуватися рівність $H(M_1) = H(M_2)$;

4) із заданого образу $h = H(M)$ неможливо ефективно знайти таке ***M'***, що $H(M') = H(M)$. Це не що інше, як властивість однобічності хеш-функції.

Створити хеш-функцію, яка задовольняє усім цим вимогам, – завдання непросте. Необхідно також пам'ятати, що на вхід функції поступають дані довільного розміру, а хеш-результат не повинен виходити однаковими для даних різного розміру.

На практиці застосовуються хеш-функції, що обробляють вхідне повідомлення блок за блоком і обчислюють хеш-значення h_i для кожного блоку M_i вхідного повідомлення рекурсивне:

$$h_t = H(M_t, h_{t-1}),$$

де h_{t-1} – результат, отриманий при обчисленні хеш-функції для попереднього блоку вхідних даних M_{t-1}.

В результаті вихід хеш-функції h_n є функцією від усіх ***n*** блоків вхідного повідомлення ***M***.

Як хеш-функцію можна узяти функцію ***H***, сконструйовану на основі RSA-функції:

$$H(M) = RSA(H(M_1, \cdots, M_{n-1}) \oplus H(M_n)),$$

де довге повідомлення ***M*** розбивається на блоки фіксованої довжини, що відповідає вибраній довжині дайджесту: $M = M_1 M_2 \cdots M_{n-1} M_n$.

Нині запропоновані і практично використовуються різні спеціальні алгоритми для обчислення хеш-функції. Найбільш відомими алгоритмами є **MD5**, **SHA** та ін.

Алгоритм MD5 (*Message Digest* – короткий дайджест повідомлення)

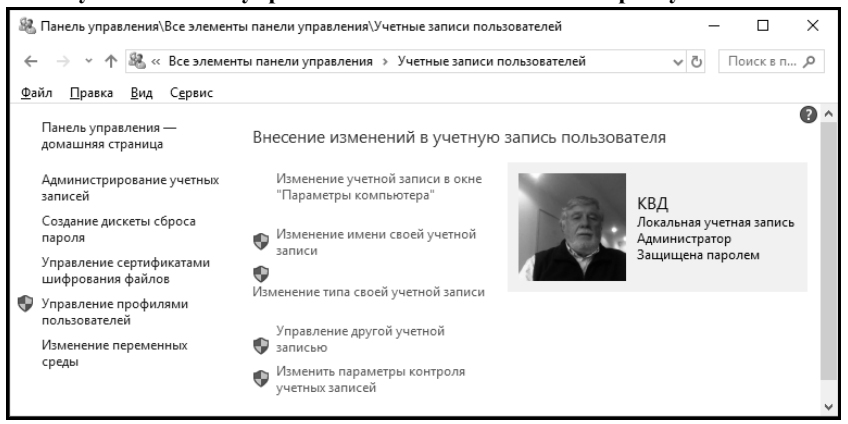
Користувачі видаленого робочого столу	Члени цієї групи мають право на виконання видаленого входу
Користувачі видаленого управління	Члени цієї групи можуть діставати доступ до ресурсів інструментарію WMI по протоколах управління (таким як WS Management в службі видаленогодоступу
Реплікатор	Група створена для підтримки функції реплікації в домені
Керована системою група облікових записів	Члени цієї групи управляються системою
Читачі журналу подій	Користувачі цієї групи можуть дистанційно читати журнали подій з локального комп'ютера

Призначення прав групам здійснюється за допомогою групової політики. Користувачам, доданим в групу, автоматично надаються усі права, призначені групі.

Утиліта Облікові записи користувачів

Призначена для управління користувачами (але не групами!!!).

Пуск ⇒ Панель управління ⇒ Облікові записи користувачів.

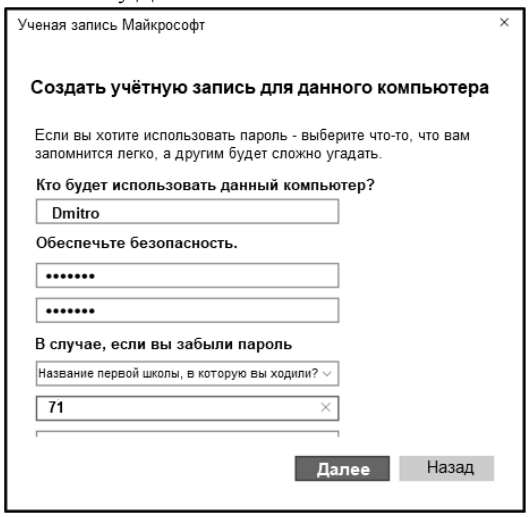


Створення нового облікового запису користувача

1. Виконаєте команди **Пуск ⇒ Параметри ⇒ Облікові записи ⇒ Сім'я і інші люди ⇒ Додати користувача для цього комп'ютера**.
2. У вікні **Виберіть спосіб входу користувача в систему** клацніть на посиланні **У мене немає даних для входу цієї людини**.
3. У наступному вікні **Створити обліковий запис Майкрософту** клацніть на посиланні **Додати користувача без облікового запису Майкрософту**.
4. У вікні **Створити обліковий запис для цього комп'ютера** заповните поля:
   - **Хто використовуватиме цей комп'ютер?** (вводиться логін користувача);

- **Забезпечити безпеку** (двічі вводиться пароль на вхід в систему);
- **У разі, якщо ви забули пароль** (вводяться три відповіді на поставлені системою питання).

5. Натисніть кнопку **Далі**.



Використання паролів

Пароль – це унікальний таємний набір дозволених символів, який має бути введений користувачем для перевірки його облікового імені та отримання доступу до ресурсів ПК.

Вимоги до вибору пароля:

1) довжина пароля не менш 7 символів (найбільш надійні паролі складаються з 8 - 14 символів);

2) містити символи кожної з трьох наступних груп:
   - букви (прописні та строкові) **A ... Z, a ... z, А … Я, а … я**;
   - цифри **0 … 9**;
   - спеціальні символи **~! @ # $ % & * ( ) _ + - =** та ін.

3) істотно відрізнятися від раніше використаних паролів;

4) не містити прізвища або імені користувача;

5) не бути поширеним словом або ім'ям.

Рекомендації з використання паролів:

- ❒ ніколи не записуйте свій пароль в якому-небудь осяжному місці;
- ❒ не повідомляйте пароль нікому;
- ❒ не використовуйте мережевий пароль для інших цілей;
- ❒ використовуйте різні паролі для входу в мережу і облікового запису адміністратора на комп'ютері;
- ❒ змінюйте свій пароль кожні 60-90 днів;
- ❒ змініть пароль негайно, якщо виникнуть підозри, що він був розкритий;

- ❒ для унеможливлення повторного використання застарілих повідомлень підпис повинен залежати від часу.

Загальна схема використання цифрового підпису включає:

1) *Імовірнісний алгоритм генерування ключів.* Кожен абонент **A** мережі генерує випадкову пару ключів (K_A, K_A'), де K_A – відкритий ключ, а K_A' – секретний ключ.

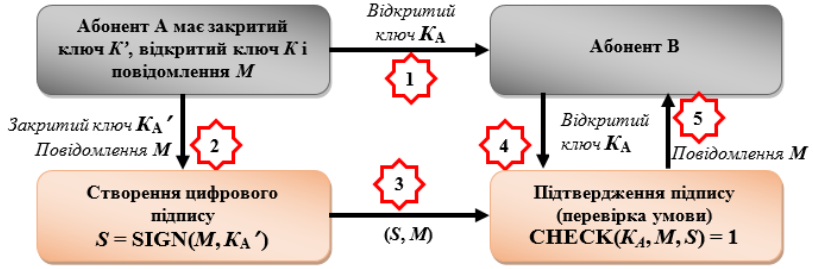
2) *Алгоритм цифрового підпису* ***SIGN***. Отримавши на вході довільне повідомлення M і секретний ключ K_A', цей алгоритм формує шифроване слово $S = \mathbf{SIGN}(M, K_A')$, яке називається **підписом абонента A на повідомленні** M. Коли **A** хоче послати комусь повідомлення M і при цьому переконати одержувача в тому, що воно дійсно належить **A**, те він передає парі (S, M).

3) *Алгоритм підтвердження підпису* **CHECK**. Одержувач повідомлення M, бажаючи переконатися в тому, що воно дійсно відправлене абонентом **A**, включає цей алгоритм, використовуючи загальнодоступний ключ K_A. Перевірка вважається успішною, якщо $\mathbf{CHECK}(K_A, M, S) = 1$.

Для будь-якого повідомлення M і для кожної пари ключів (K, K') повинне виконуватися співвідношення:

$$\mathbf{CHECK}(K, M, \mathbf{SIGN}(M, K')) = 1.$$

Виконання цієї умови означає *коректність* системи цифрового підпису. Усі вказані алгоритми мають бути *ефективними*.



Стійкість такої системи підпису означає, що лише законний власник секретного ключа K може для повідомлення M виробити такий підпис S, який пройшов би перевірку $\mathbf{CHECK}(K, M, S) = 1$. Якщо такий же підпис S знаходить порушник, то говорять, що він *підробляє* (*фальсифікує*) підпис легального абонента на повідомленні M.

До тих пір, поки абонент **A** надійно зберігає свій закритий ключ, його підписи достовірні. Крім того, неможливо змінити повідомлення, не маючи доступу до закритого ключа абонента **A**; тим самим забезпечується автентичність і цілісність даних.

Будь-яку асиметричну криптосистему можна перетворити в систему цифрового підпису таким чином: нехай E і D – відповідно алгоритми шифрування і розшифрування, K і K' – відкритий і секретний ключі, а M – довільне повідомлення. Тоді:

1. Абоненти **A** і **B** сформували наступні загальні параметри: $p = 11$, $g = 7$.
2. Абонент **A** згенерував секретний ключ $x_A = 3$, вичислив значення $y_A = 7^3 \bmod 11$ і відіслав його абонентові **B**.
3. Абонент **B** згенерував секретний ключ $x_B = 9$, вичислив значення $y_B = 7^9 \bmod 11$ і відіслав його абонентові **A**.
4. Кожен з абонентів обчислюєзагальний секретний ключ:

Абонент A	Абонент B
$k = (8)^3 \bmod 11 = 6$	$k = (2)^9 \bmod 11 = 6$

Тепер абоненти **A** і **B** мають загальний секретний ключ 6, який не передавався відкритим каналом зв'язку.

Протоколи цифрового підпису

Цифровий підпис – це унікальне числове доповнення до передаваної інформації, що дозволяє перевірити її авторство. Формується з відкритого тексту ***M*** за певним алгоритмом, потім шифрується і разом із зашифрованим або відкритим текстом передається адресатові.

Основним призначенням цифрового підпису є задача переконати одержувача повідомлення в тому, що воно поступило від відомого йому абонента і не підроблено.

Цифровий підпис повинен мати наступні властивості:

❒ підпис відтворюється тільки однією особою, а достовірність її може бути засвідчена багатьма;

❒ підпис нерозривно зв'язується з цим повідомленням і не може бути перенесена на інший документ;

❒ після того, як документ підписаний, його неможливо змінити;

❒ від поставленого підпису неможливо відмовитися, тобто особа, що підписала документ, не зможе потім стверджувати, що не ставило підпис.

Область використання цифрового підпису надзвичайно широка: від проведення фінансових і банківських безпаперових операцій до контролю за виконанням міжнародних договорів і охорони авторських прав.

Забезпечення захисту кожної сторони, що бере участь в обміні даними, здійснюється за допомогою ведення спеціальних протоколів. Для верифікації повідомлення протокол повинен містити наступні обов'язкові положення:

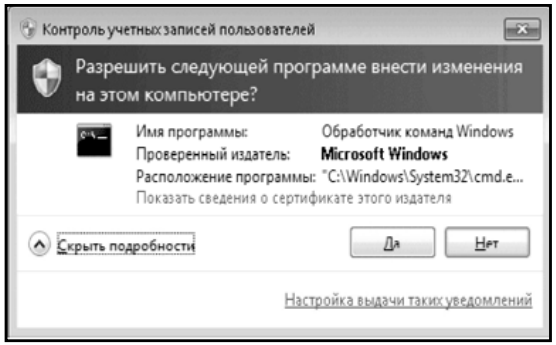
❒ відправник вносить до передаваного повідомлення свій цифровий підпис, що є додатковою інформацією, залежною від передаваних даних, імені одержувача повідомлення і деякої закритої інформації, яку має тільки відправник;

❒ одержувач повідомлення повинен мати можливість упевнитися, що отриманий у складі повідомлення підпис є правильний підпис відправника;

❒ отримання правильного підпису відправника можливе тільки при використанні закритої інформації, яку має відправник;

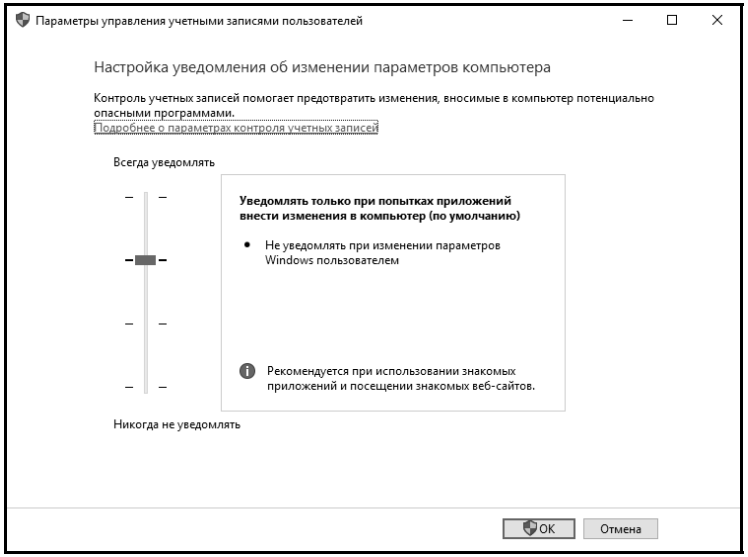
❒ будьте обережні при збереженні пароля на комп'ютері (у деяких діалогових вікнах є можливість зберегти пароль. Ніколи не встановлюйте цей параметр).

Контроль облікових записів користувачів



У Windows 10 є спеціальний засіб для запобігання несанкціонованому доступу до комп'ютера – **контроль облікових записів користувачів** (***User Account Control***, **UAC**), основне завдання якого – **запобігти несанкціонованому запуску шкідливих програм**.

Перед виконанням потенційно небезпечної дії служба контролю облікових записів просить дозвіл, і користувач повинен підтвердити запуск вибраної команди.



Усі облікові записи, що відносяться до групи **Адміністратори**, за замовчуванням працюють з правами звичайного користувача. Якщо користувач або застосування намагається виконати дію, для якої потрібно повноваження адміністратора, з'являється вікно UAC з вимогою підтвердити або відмінити ви-

брану команду. Після підтвердження запиту система тимчасово підвищує права користувача до рівня адміністратора.

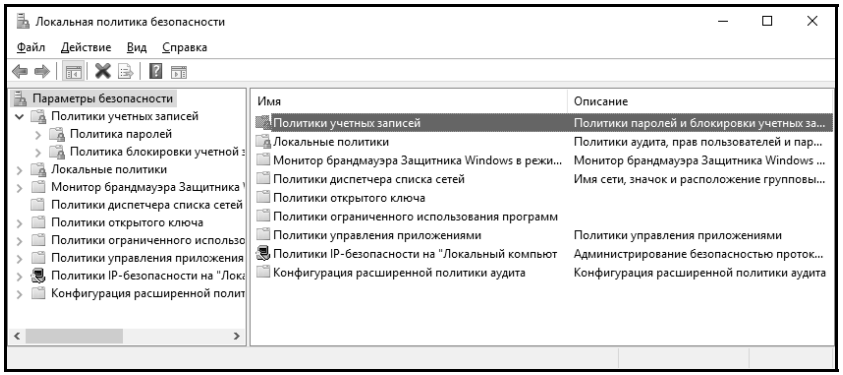
Є можливість **налаштування служби UAC**:

- ❑ відкрийте меню **Пуск** і клацніть на малюнку облікового запису;
- ❑ у вікні, що з'явилося, виберіть посилання **Змінити параметриконтролю облікових записів**;
- ❑ у вікні, що відкрилося, виберіть один з чотирьох рівнів **UAC**.

Політики управління обліковими записами

Існує можливість задати налаштування контролю облікових записів. Це можна зробити за допомогою оснащення **Локальна політика безпеки**:

Пуск ⇒ Панель управління ⇒ Адміністрування ⇒ Локальна політика безпеки



2.3 Розпізнавання користувачів

2.3.1 Програми внутрішнього захисту

Цей клас програм здійснює ЗІ безпосередньо в елементах АС. Сутність такого захисту зводиться до регулювання використання відповідних ресурсів АС (технічних засобів, даних, програм) в суворій відповідності з повноваженнями, наданими суб'єктам (користувачам) та об'єктам (терміналам, груповим пристроям, програмам). Кожен з видів регулювання зазвичай здійснюється у наступній послідовності.

1. Встановлення достовірності (розпізнавання) суб'єкта або об'єкта, що звертається до ресурсів системи.
2. Визначення відповідності характеру і змісту запиту повноваженням, які надані суб'єкту або об'єкту, що надсилає запит.
3. Прийняття та реалізація рішень відповідно до результатів перевірки повноважень.

Найбільш важливою з перерахованих процедур є перша, тобто встановлення достовірності (розпізнавання) суб'єкта або об'єкта, що звертається до

- ❒ просте число p вибирається таким, щоб виконувалася рівність $p = 2q + 1$, де q – також просте число;
- ❒ як g береться будь-яке ціле число, для якого справедливі нерівності

$$1 < g < p - 1 \quad \text{і} \quad g^q \bmod p \neq 1.$$

2. Абонент **A** вибирає випадкове число x_A ($x_A<p$) – свій секретний ключ, на його основі обчислює значення $y_A = g^{x_A} \bmod p$ і посилає його абонентові **B**.

3. Аналогічно поступає абонент **B**, вибираючи випадкове число x_B ($x_B<p$) – свій секретний ключ, - і обчислюючи $y_B = g^{x_B} \bmod p$.Це значення абонент **B** посилає абонентові **A**.

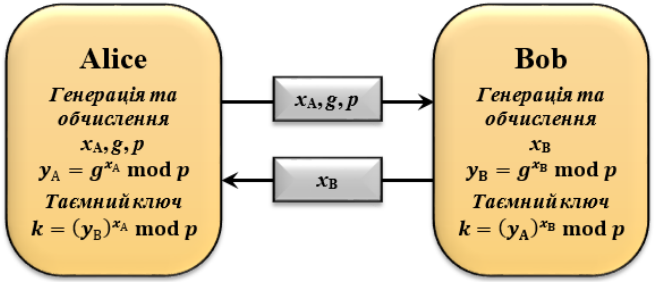
Таким чином, у абонентів сформувалися наступні дані:

	Загальні параметри	Відкритий ключ	Секретний ключ
Абонент **A**	p, g	y_A	x_A
Абонент **B**		y_B	x_B

4. З цих даних кожен з абонентів формує загальний секретний ключ k для сеансу симетричного шифрування, яким стали володіти обидва абоненти:

Абонент A	Абонент B
$k = (y_B)^{x_A} \bmod p$	$k = (y_A)^{x_B} \bmod p$

Порушник, не знаючи секретних чисел x_A і x_B, не зможе вичислити число k, навіть знаючи параметри p,g, y_A і y_B. Безпека формування загального ключа в алгоритмі Діффі-Хеллмана витікає з того факту, що, хоча відносно легко вичислити експоненти за модулем простого числа, дуже важко вичислити дискретні логарифми. Для великих простих чисел розміром сотні і тисячі біт задача вважається нерозв'язній, оскільки вимагає колосальних витрат обчислювальних ресурсів.



Розглянемо приклад використання протоколу.

нційної передачі повідомлень – історично перше завдання, яке вирішувалося криптографією.

2. **Протоколи автентифікації та ідентифікації**. Призначені для запобігання доступу до деякої інформації осіб, що не є її користувачами, а також запобігання доступу користувачів до тих ресурсів, на які у них немає повноважень. Типова сфера застосування – організація доступу користувачів до ресурсів деякої інформаційної системи.

3. **Протоколи розподілу ключів** потрібні для забезпечення секретними ключами учасників обміну зашифрованими повідомленнями.

4. **Протоколи електронного цифрового підпису** дозволяють ставити під електронними документами підпис, аналогічний звичайному підпису на паперових документах. В результаті виконання протоколу електронного цифрового підпису до передаваної інформації додається унікальне числове доповнення, що дозволяє перевірити її авторство.

5. **Протоколи забезпечення невідстежуваності** («Електронні гроші»). Під електронними грошима в криптографії розуміють електронні платіжні засоби, що забезпечують невідстежуваність, тобто неможливість прослідкувати джерело пересилки інформації.

Протокол розподілу ключів Діффі-Хеллмана

Схема обміну ключами **Діффі–Хеллман**а (*Diffie-Hellman*, *DH*), винайдена в 1976 р. Уітфілдом Діффі та Мартіном Хеллманом під впливом робіт Ральфа Меркле (*Ralph Merkle*), стала першим практичним методом для отримання загального секретного ключа при спілкуванні через незахищений канал зв'язку.

Алгоритм заснований на трудності обчислень дискретних логарифмів за модулем деякого великого простого числа ***p***. Спочатку спеціальним чином підбирається деяке натуральне число ***a<p***. Якщо потрібно зашифрувати значення ***x***, то обчислюється $y = a^x \bmod p$

Знаючи ***x***, вичислити ***y*** легко, але зворотна задача обчислення дискретного логарифма ***x*** з ***y*** ($x = \log_a y \bmod p$) є досить складною. Таким чином, знаючи про складність обчислення дискретного логарифма, число ***y*** можна відкрито передавати будь-яким каналом зв'язку, оскільки при великому модулі ***p*** вихідне значення ***x*** підібрати буде практично неможливо.

Отже, абоненти **A** і **B** бажають сформувати загальний секретний ключ, щоб далі здійснювати секретне листування, використовуючи алгоритм симетричного шифрування.

Протокол обміну наступний:

1. Абонент **A** (чи абонент **B**, це байдуже) вибирає велике просте число ***p*** і деяке спеціальне число ***g***. Пара чисел **(*p*, *g*)** відома усім абонентам системи і вибирається відкрито (це загальні параметри).

Зауваження. Щоб алгоритм Діффі-Хеллмана працював правильно, значення*p* і *g* повинні задовольняти наступним умовам:

ресурсів АС. Тому розробці ефективних засобів надійного розпізнавання незмінно приділяється підвищена увага.

Встановлення достовірності (аутентифікація, ідентифікація, розпізнавання) будь-якого об'єкта або суб'єкта полягає в підтвердженні того, що суб'єкт, який звертається, або пред'явлений об'єкт є саме тими, які повинні брати участь в даному процесі обробки інформації. Основними суб'єктами, автентичність яких підлягає встановленню у всіх системах, де обробляється інформація з обмеженим доступом, є різні користувачі. У деяких системах з підвищеними вимогами до забезпечення безпеки передбачається встановлення автентичності програмістів, що беруть участь в розробці і експлуатації програмного забезпечення, адміністраторів банків даних і навіть інженерно-технічного персоналу, залученого до технічного обслуговування системи в процесі обробки інформації, що захищається.

Складність і обсяг операцій по розпізнаванню можуть істотно відрізнятися для кожного конкретного випадку. Вони визначаються такими основними факторами:

— структурною і організаційною побудовою АС (розміри, складність архітектури, територіальний розподіл, розвиненість термінальної мережі, характер розміщення обладнання і т.п.);

— характером функціонування (наявність дистанційного доступу, режим роботи АС, обсяг і характер обміну інформацією по автоматизованим каналах зв'язку і т.д.);

— ступенем секретності інформації, що захищається і її об'ємом.

Залежно від *складності* операцій розпізнавання, фахівці виділяють три основні групи:

— просте;

— ускладнене;

— особливе розпізнавання.

За *величиною обсягу операцій* процедури розпізнавання також розбивають на три групи:

— контрольне;

— розширене;

— загальне розпізнавання.

Під *контрольним розпізнаванням* розуміють розпізнавання віддалених терміналів в моменти включення їх в роботу і при зверненні їх до системи під час обробки інформації, що захищається. При *розширеному розпізнаванні* зазвичай проводиться розпізнавання програмістів, віддалених кореспондентів, пристроїв групового управління вводом/виводом, елементів, що захищаються, баз даних і т.д. При *загальному розпізнаванні* забезпечується розпізнавання всіх суб'єктів і об'єктів, що мають відношення до обробки інформації, що захищається.

Просте розпізнавання, як правило, зводиться до порівняння коду (пароля), висунутого терміналом або користувачем, з еталонним кодом (паролем),

що зберігаються в ОП АС. При ускладненому розпізнанні зазвичай використовується додаткова інформація - система разових паролів, персональна інформація користувача і т.п.. Ускладнене розпізнавання здійснюється в режимі діалогу: система формує питання, на які розпізнаваний повинен дати відповіді. За змістом відповідей система приймає рішення про розпізнавання. При особливому розпізнаванні використовується така сукупність розпізнавальних характеристик, при якій має забезпечуватися надійне розпізнавання суб'єктів і об'єктів.

Існують також поняття прямого і *зворотного розпізнавання*. При цьому під прямим розпізнаванням розуміють розпізнавання системою суб'єктів, які до неї звертаються, і використовуваних об'єктів, а під зворотним - розпізнавання користувачем елементів системи, що надаються йому для обробки даних, які захищаються.

2.3.2 Просте розпізнаваннякористувача

Найбільш поширеною і простою процедурою є розпізнавання за кодом або паролем. Під кодом (паролем) мають на увазі деяку послідовність символів, що зберігається у таємниці та доступ до якої надається при зверненні до системи. Коди (паролі) усіх користувачів та пристроїв, що підлягають розпізнаванню, зберігаються в ОП тієї АС, в якій виконується процедура розпізнавання. Символи пароля (коду) обираються випадково. Протенайважливішою характеристикою пароля єйого довжина, оскількиза малого значення довжини пароля можнавиконати перебір усіх можливих значеньі, таким чином отримати несанкціонований доступ до системи.

Існує реальна можливість перехоплення пароля в процесійого передачі по лініях зв'язку. Для усунення такої небезпеки можна застосувати шифрування пароля (коду). Однак, в цьому випадку виникають додаткові складності, пов’язані із вибором, розподілом, зберіганням та використанням ключів з метою запобігання компрометації системи шифрування.

Під час роботи із паролями треба дотримуватись також такого запобіжного заходу, як унеможливлення їх роздрукування чи відображення на екрані дисплея. Очевидно, необхідно приділяти особливу увагу ретельним та ефективним засобам захисту пароліві кодів в ОП АС.

2.3.3 Ускладнена процедура розпізнавання

Для підвищення ефективності розпізнавання за паролем (кодом) можуть використовуватися різні ускладнені процедури: модифікація системи простих паролів, використання методів “запит — відповідь” та перехресного розпізнавання.

Найбільш поширенеми методами модифікації схеми простих паролів є *випадкова вибірка символів паролю* і *одноразове використання паролів*. При використанні першого методу кожному користувачу (пристрою) виділяється достатньо довгий пароль (код), причому кожного разу для розпізнавання використовується не весь пароль, а деяка його частина, що обирається довільним чином. В цьому випадку в процесі розпізнавання АС запитує у користувача не весь пароль, а деякійого символи, причому кількість символів та їх

У 2010 р. групі учених з Швейцарії, Японії, Франції, Нідерландів, Німеччини і США вдалося успішно вичислити дані, зашифровані за допомогою криптографічного ключа стандарту RSA завдовжки 768 біт. За словами дослідників, після їх праці надійною системою шифрування можна розглядати тільки RSA-ключи довжиною 1024 біта і більш. З 31 грудня 2013 р. браузери Mozilla перестали підтримувати сертифікати засвідчуючих центрів з ключами RSA менше 2048 біт.

4.2.3 Криптографічні протоколи

У сучасній криптографії велика увага приділяється не лише створенню і дослідженню шифрів, але і розробці криптографічних протоколів.

Криптографічний протокол – це така процедура (алгоритм) взаємодії двох або більш абонентів з використанням криптографічних засобів, в результаті якої абоненти досягають своєї мети, а їх супротивники – не досягають. В основі протоколу лежить набір правил, що регламентують використання криптографічних перетворень і алгоритмів в інформаційних процесах. Учасники протоколу повинні знати протокол і виконувати повністю усі його етапи.

Учасники протоколу можуть не довіряти один одному, тому криптографічні протоколи повинні захищати їх учасників не лише від зовнішнього супротивника, але і від нечесних дій партнерів.

Кожен криптографічний протокол призначений для вирішення певного завдання.

Будь-який протокол має наступні властивості:

- при виконанні протоколу важливий порядок дій; кожна дія повинна виконуватися у свою чергу і тільки після закінчення попередньої;
- протокол має бути несуперечливим;
- протокол має бути повним, тобто для кожної можливої ситуації має бути передбачено відповідна дія.

Нині розроблено декілька десятків різних типів криптографічних протоколів. Усі ці типи умовно розділяються на дві групи:

1) *прикладні протоколи*, що вирішують конкретні завдання, які виникають на практиці;

2) *примітивні протоколи* використовуються як своєрідні «будівельні блоки» при розробці прикладних протоколів.

До основних видів протоколів відносяться:

1. **Протоколи конфіденційної передачі повідомлень**. Завдання конфіденційної передачі повідомлень полягає в наступному. Є два учасники протоколу, які є абонентами мережі зв'язку. Учасники сполучені деякою лінією зв'язку, якою можна пересилати повідомлення в обидві сторони. Лінію зв'язку може контролювати порушник. У одного з абонентів є конфіденційне повідомлення ***M***, і завдання полягає в тому, щоб це повідомлення конфіденційним же чином передати другому абонентові. Протоколи цього типу, напевно, з'явилися раніше інших криптографічних протоколів, оскільки завдання конфіде-

❒ випадковим чином виберемо ***e***, взаємно просте з 20, наприклад ***e* = 7**. Використовуючи розширений алгоритм Евкліда, легко перевірити, що $НЗД(7, 20) = 1$;

❒ вичислимо число ***d***, що задовольняє умові $7 \times d \equiv 1 \pmod{20}$. Таким числом є ***d* = 3**;

❒ публікуємо відкритий ключ **(*e*, *n*) = (7, 33)**, а розшифровку повідомлень, що поступають, здійснюватимемо секретним ключем ***d* = 3**.

2. Шифрування:

❒ представимо шифроване повідомлення як послідовність цілих чисел за допомогою відповідності: **A = 1**, **B = 2**, .., **Z = 26**. Така криптосистема в змозі зашифрувати букви латинського алфавіту, що розглядаються як блоки;

❒ нехай передається повідомлення **CAB**, яке кодується у виді **(3, 1, 2)**. Абонент, передавальний це повідомлення, шифрує його відкритим ключем:

$$RSA(C) = RSA(3) = 3^7 \pmod{33} = 2187 \pmod{33} = 9;$$
$$RSA(A) = RSA(1) = 1^7 \pmod{33} = 1 \pmod{33} = 1;$$
$$RSA(B) = RSA(2) = 2^7 \pmod{33} = 128 \pmod{33} = 29.$$

Таким чином, криптотекст є **(9, 1, 29)**.

3. Розшифрування:

❒ спробуємо розшифрувати криптотекст **(9, 1, 29)**, використовуючи секретний ключ ***d*= 3**:

$$9^3 \pmod{33} = 729 \pmod{33} = 3;$$
$$1^3 \pmod{33} = 1 \pmod{33} = 1;$$
$$29^3 \pmod{33} = 24389 \pmod{33} = 2.$$

Таким чином, отриманий відкритий текст **(3, 1, 2)**.

Цей приклад носить чисто ілюстративний характер, тому в нім досить легко знайти секретний ключ шляхом перебору. На практиці це здійснити неможливо, оскільки рекомендуються наступні довжини шифрованих блоків:

❒ 512 – 768 біт – для приватних осіб;

❒ 1024 біт – для комерційної інформації;

❒ 2048 біт – для секретної інформації.

У серпні 1977 р. в журналі Scientific American з'явився перший опис криптосистеми RSA. Читачам було запропоновано дешифрувати англійську фразу, зашифровану описаним алгоритмом. Як відкриті параметри системи були використані числа ***n* = 1143816..6879541** (129 десяткових знаків, 425 біт) і ***e* = 9007**. За розшифровку була обіцяна нагорода в 100 доларів США. За заявою Рівеста, для факторизації числа було б потрібно більше 40×10^{15} (квадрильйонів) років. Через 15 років, 3 вересня 1993 року було оголошено про старт проекту розподілених обчислень з координацією через електронну пошту по знаходженню співмножників числа RSA-129 і рішенню головоломки. Впродовж півроку більше 600 добровольців з 20 країн жертвували процесорний час 1600 машин. В результаті були знайдені прості множники і розшифровано початкове повідомлення, яке є фразою «THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE» («Чарівні слова – це гидливий ягнятник»).

порядкові номери у паролі визначаються АС за допомогою генератора випадкових чисел.

При одноразовому використанні паролів кожному користувачу надається не один пароль, а більше, кожен з них використовується тільки один раз. Паролі можуть обиратися послідовно за списком абож випадковим чином. Недоліками цього методу є:

– користувач повинен пам'ятати всі паролітаїх послідовність (важко при значній кількості паролів) або мати при собіїх список (можлива йгог втрата чи компрометація);

– при великійкількості користувачів для генерації списку паролів необхідно використовувати генератори випадкових послідовностей, що надає змогу зловмиснику відновити паролі за допомогою статистичного аналізу.

При використанні метода "запит — відповідь" в пам'яті АС завчасно створюється и ретельно захищається масив запитань, що містить запитання загального характеру, та персональні запитання до конкретного користувача. Для розпізнавання користувача АС послідовно ставить перед ним ряд випадково обраних запитань, на які користувач повинен дати відповідь. Розпізнавання вважається позитивним, якщо у відповідях користувача число помилок не перевищує заданого.

Метод перехресного розпізнавання полягає в тому, що процедура розпізнавання повторюється періодично в процесі роботи користувача через випадкові моменти часу. Також, кожного разу можуть бути використані різні методи розпізнавання.

2.3.4 Методи особливо надійного розпізнавання

Особливо надійне розпізнавання повинно використовуватися в разі обробки інформації підвищеної секретності, особливо в разі роботи в режимі віддаленого доступу. При цьому використовуються суто індивідуальні характеристики людини: голос, відбитки пальців, сітківка ока, фотографія, особистий підпис і т.п..

Реалізація методів розпізнавання по перерахованим характеристикам пов'язана з вирішенням двох груп проблем: проблеми зняття індивідуальних характеристик людини в процесі розпізнавання і проблеми аналізу і обробки отриманих характеристик.

При розпізнанні користувача по голосу в пам'яті АС заздалегідь формується еталон його голосу, для чого користувач має виголосити перед мікрофоном задану сукупність фраз. В процесі впізнання АС порівнює вимовлені фрази зі збереженими еталонними і приймає рішення про розпізнавання.

Надійність розпізнавання по голосу в ідеальних умовах досить висока, проте на неї значно впливають такі чинники, як зміна голосу при застуді і деяких інших захворюваннях (а, можливо, і просто від втоми), можливість імітації голосу зловмисником. З цих причин розпізнавання по голосу на сьогоднішній день не отримало широкого поширення.

Розпізнавання за відбитками пальців і по сітківці ока – найбільш тради-

ційні методи розпізнавання, засновані на загальновідомому факторі, що відбитки і сітківка є строго індивідуальними характеристиками людини. При належній обробці відбитків і сітківки надійність розпізнавання може бути досить високою. Схема процедури розпізнавання для цього випадку зрозуміла і загальновідома. Основну складність при вирішенні цього завдання становить перетворення малюнків відбитків пальців і сітківки ока в цифрову форму для подальшої їх обробки на ЕОМ. Розробка і реалізація програмного забезпечення для вирішення цього завдання не є особливо важким.

Розпізнавання по довжині пальців грунтується на менш очевидному і менш відомому факті - довжина пальців і співвідношення довжин окремих пальців також є індивідуальними характеристиками людини. Вимірювання довжини чотирьох пальців (без великого) дозволяє впізнати людину з ймовірністю не нижче 95%. У той же час пристрій для вимірювання довжини пальців є настільки простим, що їм можна обладнати навіть невеликі термінали користувачів.

Розпізнавання по фотографії пов'язано з наявністю в будові особи стійких індивідуальних характеристик, сукупність яких не може бути імітована навіть при майстерному гримуванні. У цю сукупність входять: будова і розташування вух, геометричні співвідношення рис обличчя, знятого в анфас і в профіль, геометричні параметри положення очей і т.п..

Аналогічно наведеним вище методам, може здійснюватися розпізнавання по особистому підпису, причому в системах такого типу використовуються не тільки геометричні характеристики підпису, а й динамічні характеристики процесу його написання. Ці параметри також утворюють сукупність характеристик, що дозволяють досить надійно провести розпізнавання користувача. Слід зазначити, що високу надійність розпізнавання може забезпечити тільки комбінована система, що використовує кілька різних методів, хоча вона і буде досить складною і дорогою.

2.3.5 Методи розпізнавання АС і її елементів користувачем

Таке розпізнавання необхідно для того, щоб користувач міг переконатися в тому, що надані йому ресурси є саме тими, які призначені для роботи з ним, а не є хибними, фальсифікованими зловмисником для отримання секретних даних, в тому числі і паролів.

Розпізнавання користувачем системи і її окремих елементів також можна здійснити за допомогою паролів, тільки в цьому випадку сама система буде пред'являти свій код (пароль) користувачеві. Цілком очевидно, що користувач повинен знати такий пароль заздалегідь. Такий метод розпізнавання при великій кількості користувачів не може бути надійним.

Найбільш ефективним методом вирішення даної задачі в даний час вважається реалізація так званої *"схеми рукостискання"*. При її реалізації заздалегідь вибирається не дуже складне, але далеко не тривіальне перетворення $A(x,k_t)$, де x – аргумент, а k_t – ключ, що залежить від поточного часу. Це перетворення повинно водночас і бути секретним, і бути відомим

загального дільника $НЗД(e,\varphi(n))$ і $d = e^{-1} \pmod{\varphi(n)}$, які є поліноміальними, тобто ефективними;

❒ в алгоритмах шифрування і розшифрування піднесення до ступеня здійснюється за допомогою бінарного методу, який також є ефективним.

6) *Надійність:*

❒ криптосистема вважається стійкою до криптоатак у тому випадку, коли задача дешифруванняповідомлення за криптотекстом і відкритим ключем є важковирішувальною;

❒ **задача дешифрування RSA**:

- задано: e, n, y (перехоплений криптотекст);
- знайти: x (відкритий текст) таке, **що** $x^e \equiv y \pmod{n}$;

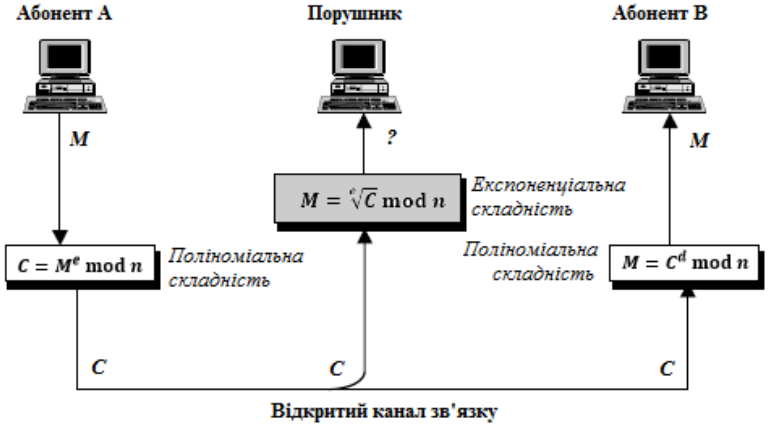
❒ таким чином, задача дешифрування RSA є завданням отримання із заданого цілого числа y кореня e-го ступеня за модулем n (на сьогодні для цієї задачі невідомо жодного ефективного алгоритму);

❒ можна спробувати розкрити RSA шляхом визначення секретного ключа d з відкритого ключа e, вирішивши наступну задачу:

- задано: e, n (відкритий ключ), де $n = p \times q$ і $НЗД(e,\varphi(n)) = 1$;
- знайти: x (відкритий текст) таке, що $x^{e \times d} \equiv x \pmod{n}$ для усіх x;

❒ ця задача зводиться до обчислення значення функції Ейлера $\varphi(n)$, яка, у свою чергу, еквівалентна завданню знаходження співмножників p і q числа n (так звана задача факторизації). На сьогодні вирішити подібну задачу – безнадійна справа для n близько 10^{200}; тому при генеруванні ключів p і q рекомендується вибирати приблизно з сотнею десяткових цифр кожне.

Складність операцій в RSA



Приклад побудови криптосистеми RSA:

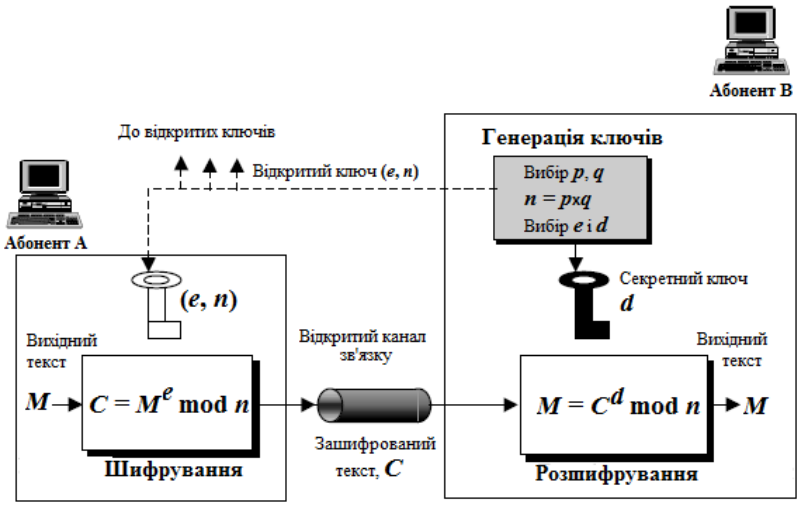
1. Генерування ключів:

❒ виберемо $p = 3$ і $q = 11$; вичислимо $n = p \times q = 3 \times 11 = 33$ і знайдемо $\varphi(33) = (3-1) \times (11-1) = 2 \times 10 = 20$;

❒ за допомогою алгоритму Евкліда для e знаходиться число d, зворотне до нього і таке, що $d < \varphi(n)$ і $e \times d \equiv 1 \pmod{\varphi(n)}$;

❒ як результат вважають: *відкритий ключ*: e, n; *секретний ключ*: d.

Асиметрична криптосистема RSA



2) *Шифрування*:

❒ вихідне повідомлення перетвориться в цифровий код (наприклад, двійковий) і розбивається на блоки так, щоб кожен блок означав ціле число, яке не перевищуєn (наприклад, якщо блок записаний в двійковому коді довжини m, то повинна виконуватися нерівність $2^m < n$);

❒ черговий блок M_i шифрується шляхом піднесення його до ступеня e:

$$C_i = E_{e,n}(M_i) = M_i^e \pmod{n};$$

❒ в результаті виходить блок криптотекста C_i тієї ж довжини, що і блок відкритого тексту M_i, який і передається відкритим каналом зв'язку.

3) *Розшифрування*:

❒ прийнятий з каналу зв'язку черговий блок криптотекста C_i підноситься до ступеня d (секретний ключ): $M_i = D_d(C_i) = C_i^d \pmod{n}$.

4) *Коректність*:

❒ можливість розшифрування криптотекста $D_d\left(E_{e,n}(M)\right)$ витікає з наступного доведеного твердження: якщо $n = p \times q$ є добутком двох простих числі і$e \times d = 1 \pmod{\varphi(n)}$, то для усіх $x \in Z_n$ виконується: $x^{e \times d} = x \pmod{n}$.

5) *Ефективність*:

❒ алгоритм генерування ключів використовує процедуру знаходження простих чисел і розширений алгоритм Евкліда для обчислення найбільшого

користувачеві та системі. Користувач разом із запитом на роботу посилає вибране ним значення аргументу x (наприклад, своєім'я). Система обчислює $A_c(x,k_t)$ і надсилає це значення користувачу. Користувач обчислює $A_п(x,k_t)$. Якщо $A_c = A_п$, розпізнавання вважається позитивним ("рукостискання відбулося").

Така схема розпізнавання може бути досить ефективною навіть при великому числі користувачів, оскільки для кожного користувача неважко підібрати окреме перетворення. Особливо просто реалізується режим "рукостискання" при наявності шифрувальної апаратури, що сполучається як із терміналом, так і з АС. Тоді як перетворення $A(x,k_t)$ може використовуватися криптографічне перетворення, що реалізовується в наявній криптографічного системі.

2.3.6 Проблеми регулювання використання ресурсів

Регулювання використання технічних засобів зазвичай здійснюється за такими параметрами, як загальне право на доступ, час доступу і виконувана функція.

Регулювання по загальному праву на доступ полягає в тому, що для кожного технічного пристрою з обмеженнями на доступ складається список суб'єктів і об'єктів, що мають право доступу до нього. Тоді регулювання полягатиме в дозволі доступу в тому випадку, коли суб'єкт або об'єкт, які звертаються, містяться в списку як ті, що мають право доступу, і заборони доступу в іншому випадку.

Регулювання доступу за часом полягає в тому, що для всіх суб'єктів або об'єктів може бути встановлено не загальне право доступу, а право доступу в певний час (дні тижня, число, години). Аналогічно, регулювання доступу до виконуваних функцій полягає в дозволі суб'єкту або об'єкту виконувати лише строго певні функції. На практиці можуть використовуватися і комбіновані системи регулювання доступом.

Регулювання доступу до баз (масивів) даних набуло широкого поширення при ЗІ в АС. Зауважимо, що даний вид регулювання доступу є одним з основних, який передбачається в будь-якій системі захисту.

За елементарну (найменшу) одиницю інформації, що захищається, найчастіше приймається файл, і це обумовлено двома обставинами: по-перше, саме файл найчастіше виступає одиницею інформаційного обміну, по-друге, на рівні файлу найпростіше вирішуються завдання регулювання доступу.

Всі файли, які захищаються, за ознакою приналежності зазвичай ділять на загальні, групові та особисті. До *загальних* відносяться файли сервісних програм: операційні системи, бібліотеки загального користування і т.п. До загальних файлів дозволено доступ всім користувачам, зареєстрованим в даній АС. *Груповими* зазвичай є файли даних довідкового характеру (пов'язані з певною сферою діяльності або ті, що належать будь-якій організації), бібліотеки програм групового користування та інші подібні файли. Доступ до

групових файлів дозволено деякій заздалегідь визначеній групі користувачів. *Особисті* файли належать одному користувачеві, який їх створює і має право доступу до них. Іншим особам доступ може бути наданий тільки з дозволу власника файлу.

Інформації, яка організована в файли і підлягає захисту, присвоюється відповідний гриф секретності. Порядок присвоєння грифу секретності регламентується законодавчими актами.

До теперішнього часу розроблено кілька способів розмежування доступу:

— розмежування за списками;
— матричне розмежування;
— розмежування за рівнями (кільцями) секретності;
— сторінкова організація пам'яті;
— мандатна система доступу.

Розмежуванняза списками здійснюється в тому випадку, якщо права користувачів на доступ задані у вигляді списків. При цьому або для кожного елемента бази наведено список користувачів, які мають право доступу до нього, або для кожного користувача заданий перелік тих елементів бази, до яких йому дозволено доступ. У будь-якому випадку процедура розмежування реалізується в такій послідовності.

1. За даними, що містяться в запиті, вибирається відповідний рядок списку: перелік користувачів, які мають право доступу до елементу або перелік елементів баз даних, до яких допущено користувача, що звернувся із запитом.

2. У вибраному рядку перевіряється наявність імені користувача, який звернувся із запитом, чи імені елемента бази даних, до якого звертається користувач.

3. За даними перевірки приймається рішення про допуск до даних. Крім того, можуть передбачатися санкції за спробу несанкціонованого доступу, причому в якості санкцій можуть бути вжиті такі заходи: попередження користувача про те, що їм допущені несанкціоновані дії; відключення користувача від системи в цілому або на деякий час; подача сигналу контрольним органам про спробу несанкціонованих дій.

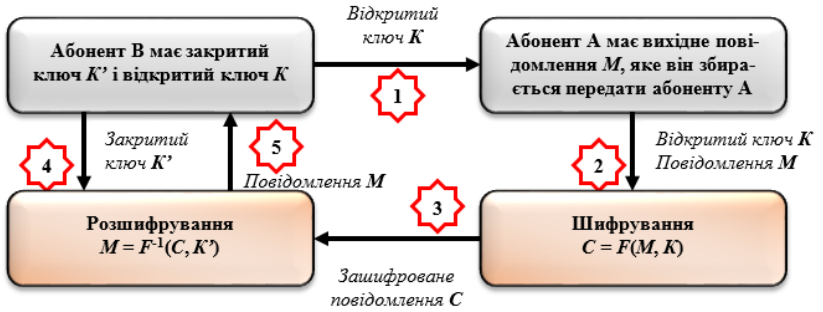
Матричне розмежування є більш гнучким у порівнянні з розмежуванням за списками, оскільки воно дозволяє не тільки регулювати доступ до даних, але і характер виконуваних процедур (читання, запис, реконструювання даних і т.д.). Забезпечується це тим, що права користувачів задаються у вигляді матриці, по рядках якої представлений список користувачів, а по стовпцях - перелік імен елементів бази даних. Елементами матриці є коди, кожен з яких містить інформацію про повноваження відповідних користувачів щодо відповідних елементів бази даних. Безліч можливих прав визначається розрядністю коду.

2. Абонент ***A*** шифрує своє повідомлення ***M*** отриманим відкритим ключем ***K*** і отримує шифрограму ***C***.

3. Шифрограма ***C*** пересилається абонентові ***B***.

4. Абонент ***B*** розшифровує отриману шифрограму ***C*** своїм закритим ключем ***K'***.

5. Абонент ***B*** читає отримане секретне повідомлення ***M***.



Загальні вимоги до алгоритмів шифрування з відкритим ключем:

1. Обчислювальне легко створювати пару (відкритий ключ, секретний ключ).

2. Обчислювальне легко зашифрувати повідомлення відкритим ключем.

3. Обчислювальне легко розшифрувати повідомлення, використовуючи секретний ключ.

4. Обчислювальне неможливо, знаючи відкритий ключ, визначити відповідний секретний ключ.

5. Обчислювальне неможливо, знаючи тільки відкритий ключ і зашифроване повідомлення, відновити вихідне повідомлення.

Криптосистема RSA

Як приклад асиметричної криптосистеми розглянемо **систему RSA**, запропоновану в 1977 р. Рональдом Райвестом, Аді Шаміром і Леонардом Адлеманом (назва системи складається з перших букв прізвищ її авторів – Rivest, Shamir і Adleman)[2]. У грудні 1997 р. була обнародувана інформація, згідно якої британський математик Кліффорд Кокс, що працював в центрі урядового зв'язку Великобританії, описав криптосистему, аналогічну RSA в 1973 р.

1) *Генерування ключів*:

❒ вибираються два досить великих простих числа ***p*** і ***q***, і знаходиться їх добуток $n = p \times q$;

❒ знаходиться функція Ейлера $\varphi(n) = (p-1) \times (q-1) = n - p - q + 1$;

❒ випадковим чином вибирається ціле число ***e***, що не перевищує $\varphi(n)$ є взаємно простим з ним;

[2] *Rivest R.L., Shamir A., Adleman L.M.* A method for obtaiming digital signatures and public-key cryptosystems. Communications of the ACM, № 21, 1978, p. 120-126.

❒ при будь-якому K існує поліноміальний алгоритм обчислення значень
$y = F_K(x)$;

❒ при невідомому K не існує поліноміального алгоритму зворотного обчислення значення $x = F_K^{-1}(y)$;

❒ при відомому K існує поліноміальний алгоритм зворотного обчислення значення $x = F_K^{-1}(y)$.

Для практичних цілей криптографії було побудовано декілька функцій, які гіпотетично вважаються односторонніми:

1) **функція множення:** $\mathrm{MULT}(x, y) = x \times y$ визначена на парах натуральних чисел x і y з однаковою кількістю значущих цифр (складність розкладання цілого числа на прості множники);

2) **RSA функція:** $\mathrm{RSA}(x, e, m) = \langle x^e \bmod m, e, m \rangle$, визначепна для добутку
$m = p \times q$ двох різних простих чисел і такого e, що **НЗД**$(e,\varphi(m)) = 1$, $x \in Z_m$ (НЗД – найбільший загальний дільник);

3) **квадратична функція Рабина:** $\mathrm{SQUARE}(x, m) = \langle x^2 \bmod m, m \rangle$, визначена для добутку $m = p \times q$ двох різних простих чисел і $x \in Z_m$;

4) **експоненціальна функція:** $\mathrm{EXP}(x, g, p) = \langle g^x \bmod p, g, p \rangle$, визначена для довільного простого p, первинного кореня g за модулем p і $x \in Z_p^*$;

5) **еліптичні функції виду** $y^2 = x^3 + ax + b$.

У чому ж полягають переваги нової криптографічної схеми над класичною? Розглянемо комунікаційну мережу, якою користуються Nабонентів. Кожен з них хоче встановити конфіденційний зв'язок з кожним. Якщо використовувати класичну криптосистему, то буде потрібно $N \times (N - 1)/2$ ключів, які мають бути передані за такою ж кількістюзакритих каналів зв'язку. У разі застосування криптосистеми з відкритим ключем кожен з абонентів самостійно генерує свою пару ключів (K_i, K_i'), $i = 1, 2, ..., N$, причому тільки N ключів (K_i) поступають в загальне користування і при цьому з них не роблять секрету.

Застосування функцій з лазівкою в криптографії дозволяє:

1) організувати обмін шифрповідомленнямиз використанням тільки відкритих каналів зв'язку, тобто відмовитися від секретних каналів для попереднього обміну ключами;

2) включити в завдання криптоаналізу («злому» шифру) важке математичне завдання і тим самим підвищити обґрунтованість стійкості шифру;

3) вирішувати нові криптографічні завдання, відмінні від шифрування (цифровий підпис, розподіл ключів та ін.).

Загальна схема відкритого шифрування

Абонент A хоче передати секретне повідомлення абоненту B так, щоб ніхто інший не смог його прочитати. Для цього необхідно виконатинаступні дії:

1. Абонент B посилає абонентові A свій відкритий ключ K будь-яким відкритим каналом зв'язку, наприклад, електронною поштою.

Недоліками методу розмежування по матриці повноважень вважаються дві обставини: для великих систем з великим об'ємом даних, що захищаються, матриці повноважень виявляються громіздкими, динамічне ведення матриць в процесі функціонування системи є досить складною процедурою.

Розмежування доступу за рівнями (кільцями) секретності полягає в тому, що бази (масиви) даних, які захищаються, діляться на частини відповідно до рівнів їх секретності. Повноваження кожного користувача задаються максимальним рівнем секретності даних, доступ до яких йому дозволено. Відповідно до цього, користувачеві дозволяється доступ до всіх даних, рівень секретності яких не вище рівня його повноважень. Неважко помітити, що таке розмежування є найменш гнучким з усіх розглянутих.

Сторінкова організація пам'яті полягає в поділі обсягу ОП АС на блоки (сторінки) фіксованого розміру. При цьому засобами операційної системи організовується управління використанням сторінок програмами користувача. Будь-яка спроба несанкціонованого входження в поле сторінки буде викликати переривання.

Мандатна система доступу (або доступ за перепустками) полягає в тому, що користувачеві видається мандат (пропуск) на доступ до відповідних масивів даних або сегментів пам'яті. При кожному зверненні здійснюється перевірка наявності мандата. Сама процедура розмежування є досить простою: пред'явлений мандат порівнюється з еталонним і за результатами порівняння приймається рішення про допуск. Однак при цьому виникають ті ж труднощі, що і при роботі з паролями - можливі перехоплення, розгадування мандатів і т.п.

Основним засобом розмежування доступу в великих банках даних є програмний механізм замків управління доступом. Цей механізм дозволяє оголосити будь-який елемент бази закритим і привласнити йому персональний замок. Після цього доступ до даного елементу бази буде дозволений тільки в тому випадку, якщо в запиті буде представлений ключ саме до цього замку. Використовувана мова опису даних дозволяє закрити замком будь-яку структуру на всіх ієрархічних рівнях. Сам замок може бути заданий у вигляді постійного коду, значеннями змінної або результатом деякої процедури. Якщо замок заданий константою або значенням змінної, то для доступу до даних необхідний простий збіг замку і пред'явленого ключа. Якщо ж замок заданий процедурою, то доступ до даних буде дозволений тільки в разі отримання заздалегідь визначеного результату процедури.

Розмежування доступу за допомогою механізму замків управління доступом вважається досить ефективним методом захисту даних. Однак лише цього захисту недостатньо. В сучасних автоматизованих банках даних, орієнтованих на колективне використання і довготривале зберігання інформації, механізм захисту повинен бути розвиненим і багато-функціональним. Такий механізм повинен мати наступні характеристики:

1. Мати засоби розпізнавання терміналів і користувачів, причому система розпізнавання повинна бути розвиненою і надійною.
2. Забезпечувати захист по різним аспектам і на різних рівнях:
   — за компонентами банку даних, до яких відносять компоненти структур даних, компоненти структур пам'яті, службові дані, і т.д .;
   — за операціями розмежування доступу і виконання програм і процедур, розмежування повноважень переміщення даних в оператив-ній пам'яті, контролю санкціонованості реорганізації баз і т.п .;
   — за умовами виконання операції в залежності від вмісту даних про об'єкти, в залежності від вхідних даних, в залежності від частоти звернень і т.п.
3. Забезпечувати розмежування по ієрархічній системі повноважень, коли користувач має свої повноваження і повноваження всіх користувачів, які є підлеглими.
4. Мати можливість криптографічного закриття даних в базах.
5. Мати розвинену систему реагування на спроби несанкціонованого доступу (повідомлення користувача, зняття завдання, відключення терміналу, видалення порушника зі списку користувачів, подача сигналу тривоги).
6. Мати засоби специфікації правил захисту як за допомогою мови опису даних, так і за допомогою автономної мови.

Висновки

1. **Захист інформації** – сукупність методів і засобів, що забезпечують цілісність, конфіденційність і доступність інформації за умов впливу на неї загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації.

2. **Комп'ютерна безпека** – це сукупність проблем у галузі телекомунікацій та інформатики, пов'язаних з оцінкою і контролюванням ризиків, що виникають при користуванні комп'ютерами та комп'ютерними мережами і розглядуваних з точки зору конфіденційності, цілісності і доступності.

3. **Загроза безпеки КС** – сукупність умов і чинників, що визначають потенційну або реально існуючу небезпеку порушення конфіденційності, цілісності, (правомірної) доступності комп’ютерної інформації, спостереженості та керованості КС, і/або зниження надійності (безвідмовності і автентичності) реалізації функцій КС.

4. **Політика безпеки інформації** – сукупність законів, правил, обмежень, нормативних документів, рекомендацій, інструкцій тощо, які регламентують порядок обробки інформації в КС і спрямованих на захист інформації від певних видів загроз.

вуваний в шифруванні, а ***K′*** – *секретний ключ*, використовуваний для розшифрування.

❒ **Поліноміальний детермінований алгоритм шифрування *E***, який отримує на вході повідомлення ***M*** і відкритий ключ ***K***, а видає криптограму $C = E_K(M)$.

❒ **Поліноміальний детермінований алгоритм розшифрування *D***, одержуючий на вході криптотекст ***C*** і секретний ключ ***K′***, і видаючий відкритий текст$M = D_{K'}(C)$.

Криптосистеми з відкритим ключем повинні задовольняти наступним **умовам**:

1) якщо пара ключів (K, K') породжена алгоритмом генерування ключів, то із $C = E_K(M)$ витікає $M = D_{K'}(C)$ для будь-якого відкритого тексту ***M***;

2) немає (чи, принаймні, невідомо) жодного ефективного алгоритму, який по відомих $C = E_K(M)$ і ***K*** міг би знайти ***M*** – це умова *стійкості криптосистеми*.

Центральним поняттям «новій криптографії» є поняття *односторонньої функції*.

Одностороння функція $F: X \to Y$ – функція, що має дві властивості:

❒ існує поліноміальний алгоритм обчислення значень $y = F(x)$;

❒ не існує поліноміального алгоритму зворотного обчислення значення $x = F^{-1}(y)$.

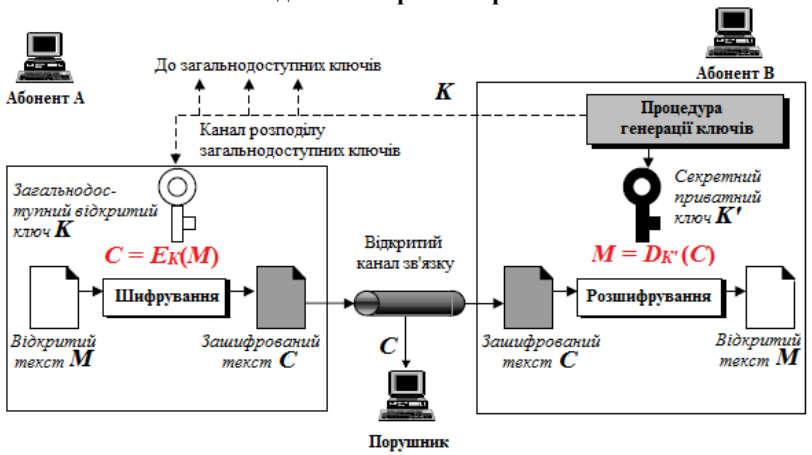
Існування поліноміального алгоритму обчислення односторонньої функції означає наступне. Нехай шифроване повідомлення включає ***n*** букв. Якщо час на шифрування цього повідомлення задовольняє нерівності $t(n) \le c \cdot n^c$для деякої константи $c > 0$, то говорять, що алгоритм вирішує задачу шифрування за *поліноміальний* час. Такі алгоритми відносяться до класу *ефективних*, оскільки реалізують вирішувані завдання за реальний час. Якщо час реалізації алгоритму $t(n) > c \cdot 2^{n^c}$для деякої константи $c > 0$, алгоритм називається *експоненціальним*. Такий алгоритм є *неефективним*, оскільки він не може бути реалізований за розумний час. Прикладом такого алгоритму є повний перебір можливих ключів в завданні криптоаналізу (так звана «лобова атака»).

Існування односторонніх функцій досі не доведене. Проте, сучасна асиметрична криптографія ґрунтується на припущенні, що односторонні функції все-таки існують.

У чистому вигляді односторонні функції в криптографії не використовуються, оскільки неможливо за розумний час отримати зворотну функцію, тому для розшифрування повідомлення, зашифрованого за допомогою односторонньої функції, у одержувача має бути ключ, що допомагає йому вирішити завдання. Функцію, що реалізує таку можливість, називають ***функцією з лазівкою*** (***функцією з секретом***, ***функцією з пасткою***).

Функція з лазівкою $F_K: X \to Y$ – це функція, залежна від деякого параметра ***K***, що має властивості:

Загальна ідея асиметричної криптосистеми



2. Асиметрична криптографія означає, що абоненти не можуть використовувати одну і ту ж множину ключів для двостороннього зв'язку. Кожен об'єкт в співтоваристві створює свій власний секретний і відкритий ключі доступу.

3. Асиметрична криптографія означає, що абонент ***B*** потребує тільки одного секретного ключа, щоб отримувати усю кореспонденцію від будь-якого учасника співтовариства. А ось абонент ***A*** потребує зв'язки ключів, щоб зв'язатися з ***N*** об'єктами в співтоваристві – один ключ доступу для кожного, тобто ***A*** потребує кільця ключів доступу.

На відміну від криптографії з симетричними ключами, в асиметричній криптографії вихідний текст і зашифрований текст обробляються як цілі числа. Повідомлення повинне перед шифруванням кодуватися як ціле число (чи множина цілих чисел). Після розшифрування воно має бути також цілим числом (чи множиною цілих чисел). Асиметрична криптографія зазвичай зашифровує або розшифровує маленькі частини інформації, визначувані завдовжки ключа шифру.

Шифрування і розшифрування в асиметричній криптографії – математичні функції, які застосовуються до чисел, що представляють вихідний текст і зашифрований текст.

Поняття криптосистем з відкритим ключем включає наступні об'єкти:

- **Алфавіт** ***A***, в якому записується повідомлення (відкритий текст), і **алфавіт** ***B***, в якому записується криптограма.
- **Простір ключів** ***K*** – множина слів в деякому алфавіті.
- **Алгоритм генерування ключів** – поліноміальний імовірнісний алгоритм, що видає випадкову пару **(*K, K'*) ∈*K***: ***K*** – *відкритий ключ*, використо-

5. Основними механізмами, що забезпечують безпеку інформації в КС, які реалізовані в ОС сімейства MS Windows є засоби контролю і управління доступом в комп'ютерну систему, до файлів і папок, що зберігаються і оброблюються; засоби, що забезпечують безпеку інформації в місцях її зберігання, в процесі обробки і в ході передачі каналами зв'язку (криптографічний захист); засоби захисту від шкідливого програмного забезпечення (антивірусний захист); засоби захисту периметра комп'ютерної системи (міжмережеві екрани – брандмауери).

6. **Дозвіл**– правило, пов'язане з об'єктом і використовуване для управління доступом користувачів до цього об'єкту. Передбачений як стандартний набір дозволів (для загальних випадків), так і спеціалізований набір – для «тонкого» налаштування.

Питання для самоконтролю

1. Які засоби захисту інформації реалізовані в ОС Windows?
2. Якими засобами налаштовується безпека в ОС Windows 10?
3. Для яких цілей використовується брандмауер Windows 10?
4. Для чого призначена програма Захисник Windows?
5. Як здійснюється шифрування файлів і папок в Windows 10?
6. Якими засобами шифруються розділи жорсткого диска і USB-носіїв в Windows 10?
7. У чому полягають стандартні дозволи до файлів і папок?
8. Для чого служать облікові записи користувачів і як вони встановлюються?
9. У чому сенс парольного захисту? Як вибираються паролі?
10. Як здійснюються опізнавання користувачів?
11. Як здійснюються регулювання використанням ресурсів?

Література

1. Введение во внутреннее устройство Windows. [Електронний ресурс]. – Режим доступу: http://www.intuit.ru/studies/courses/10471/1078/info
2. Грайворонський М.В. Безпека інформаційно-комунікаційних систем / М.В. Грайворонський, О.М. Новіков. — К: Видавнича група BHV, 2009. – 608 с
3. Информатика. Базовый курс. 3-е издание / Под ред. С.В. Симоновича. — СПб.: Питер, 2011. — 640 с.
4. Основы организации операционных систем Microsoft Windows. [Електронний ресурс]. – Режим доступу: http://www.intuit.ru/studies/courses/1089/217/info
5. Руссинович М. Внутреннее устройство Windows / М. Руссинович, Д. Соломон, А. Ионеску, П. Йосифович; 7-е изд. – СПб.: Питер, 2018. – 944 с.

6. Шаньгин В.Ф. Защита информации в компьютерных системах и сетях/ В.Ф.Шаньгин. – М.: ДМК Пресс, 2012. – 592 с.
7. Ленков С.В. Методы и средства защиты информации. В 2-х томах / С.В. Ленков, Д.А. Перегудов, В.А. Хорошко – К: Арий, 2008.

❒ розглянуті алгоритми мають високу криптостійкість, особливо коли використовуються комбіновані методи шифрування/розшифрування.

В той же час «вузьким» місцем цього класу криптосистем є необхідність у високонадійному каналі зв'язку для передачі секретного ключа. При невеликому числі абонентів ця проблема не викликає серйозних перешкоди, але коли кількість абонентів зростає і між кожним з них слід організувати такий канал, то ця проблема стає практично нерозв'язною. В зв'язку з цим на допомогу можуть прийти криптосистеми з відкритим ключем.

4.2.2 Криптосистеми з відкритим ключем

Виниклі труднощі використання криптографічних засобів (особливо пов'язані з розподілом і використанням ключів шифрування) і нові завдання, які вимагають масового застосування криптографічних засобів (наприклад, завдання ефективної реалізації електронного підпису, розподілу ключів) поставили як актуальною проблему пошуку нових методів криптографічного закриття.

1976 рік відкрив сучасний етап в криптографії. Він пов'язаний з іменами американських математиків Уайтфилда Діффі і Мартина Хеллмана, а також Ральфа Меркля, які висунули ідею відкритого ключа [1].

У криптосистемі з відкритим ключем процедура шифрування є загальнодоступною, тобто ключ шифрування (***K***) не ховається в таємниці (він називається ***відкритим ключем***), але це не означає, що загальнодоступною є процедура розшифрування. Саме ключ, за допомогою якого здійснюється розшифрування шифрограм (***K′***), є ***секретним***. Причому знаходження секретного ключа ***K′*** по загальнодоступному шифруючому відкритому ключу ***K*** є дуже складним завданням, практично нерозв'язним за розумний час. У зв'язку з тим, що ***K ≠ K′,*** такі криптосистеми ще називають ***асиметричними***.

Особливості асиметричних криптосистем:

1. Відповідальність за забезпечення безпеки знаходиться, головним чином, на плечах приймача – абонента ***B***, який повинен створити два ключі: один секретний (приватний) і один відкритий (загальнодоступний). Він не несе відповідальність за розподіл відкритого ключа доступу усьому співтовариству. Це може бути зроблено через канал розподілу відкритого ключа доступу. Хоча цей канал не зобов'язаний забезпечувати секретність, він повинен забезпечити встановлення достовірності і цілісність інформації про ключ.

[1] *Диффи У., Хеллмэн М.Э.* Новые направления в криптографии. ТИИЭР, № 22, 1976, с. 644-654; *Merkle R.C.* Secure communication over insecure channels. Communications of the ACM, № 21, 1978, p. 294-299.

Імітовставка виробляється для $M \geq 2$ блоків відкритого тексту по 64 біт. Алгоритм наступний:

1. Блок відкритих даних записується в регістри N_1 і N_2 , після чого піддається перетворенню, відповідному першим 16 циклам шифрування в режимі простої заміни.
2. До отриманого результату побітно по модулю 2 додається наступний блок відкритих даних. Останній блок при необхідності доповнюється нулями. Сума також шифрується відповідно до пункту 1.
3. Після додавання і шифрування останнього блоку з результату вибирається імітовставка довжиною Lбіт: з біта номер $32 - L$ до 32 (відлік починається з 1). Стандарт рекомендує вибирати Lвиходячи з того, що ймовірність нав'язування помилкових даних дорівнює 2^{-L}. Имитовставка передається по каналу зв'язки після зашифрованих блоків.

Для перевірки приймаюча сторона після розшифровки тексту проводить аналогічну процедуру формування імітовставки. У разі неспівпадання результату з переданою імітовставкою усе відповідні блоки вважаються помилковими.

Достоїнства ДСТУ ГОСТ 28147:2009:

- безперспективність атаки повним перебором;
- ефективність реалізації і, відповідно, висока швидкодія на сучасних комп'ютерах;
- наявність захисту від нав'язування помилкових даних (вироблення імітовставки) і однаковий цикл шифрування в усіх чотирьох алгоритмах.

Основні проблеми ГОСТу пов'язані з неповнотою стандарту в частині генерації ключів і таблиць замін. Вважається, що у ГОСТу існують «слабкі» ключі і таблиці замін, але в стандарті не описуються критерії вибору і відсіву «слабких». Також стандарт не специфікує алгоритм генерації таблиці замін (*S*-блоків). З одного боку, це може бути додатковою секретною інформацією (крім ключа), а з іншого, піднімає ряд проблем:

- не можна визначити криптостійкість алгоритму, не знаючи заздалегідь таблиці замін;
- реалізації алгоритму від різних виробників можуть використовувати різні таблиці замін і можуть бути несумісні між собою;
- можливість навмисного надання слабких таблиць замін органами, що проводять ліцензування;
- потенційна можливість (відсутність заборони в стандарті) використання таблиць заміни, в яких вузли не є перестановками, що може привести до надзвичайного зниження стійкості шифру.

Підводячи підсумок розгляду криптосистем з секретним ключем, слід зауважити наступне:

- можливість реалізації алгоритмів шифрування/розшифрування як апаратними, так і програмними засобами робить їх досить швидкими;

3. ШКІДЛИВІ ПРОГРАМИ ТА ЗАСОБИ ЗАХИСТУ ВІД НИХ

Тема захисту ПК від шкідливих програм є дуже актуальною і важливою. Незважаючи на появу досить потужних антивірусних пакетів, небезпека зараження комп'ютерів не лише не зменшується, але продовжує зростати.

Проблема забезпечення безпеки інформаційних ресурсів також вирішується шляхом архівації важливих даних.

3.1 Шкідливі програми: поняття, класифікація, способи поширення

3.1.1 Поняття шкідливого програмного забезпечення

Шкідлива програма (*malware* – з'єднання слів ***mal****icious* (зловмисний) і *soft****ware*** (програмне забезпечення)) – будь-яке програмне забезпечення, призначене для діставання несанкціонованого доступу до обчислювальних ресурсів комп'ютера або до інформації, що зберігається і оброблюється в комп'ютері, з метою несанкціонованого використання ресурсів ЕОМ або спричинення шкоди (завдання збитку) власникові інформації, власникові ЕОМ, власникові мережі ЕОМ шляхом копіювання, спотворення, видалення або підміни інформації, порушення умов порушення умов спостережливості і керованості комп'ютерних систем.

Crimeware (***crime*** – злочинність і *soft****ware*** – програмне забезпечення) – клас шкідливих програм, спеціально створений для кіберзлочинів. Не усі програми, що відносяться до **crimeware**, є шкідливими, оскільки поняття злочину суб'єктивне і залежить від законодавства конкретної країни, а шкода, що наноситься власникові і/або користувачеві комп'ютера, – об'єктивна.

За основним визначенням, шкідливі програми призначені *для діставання несанкціонованого доступу* до інформації в обхід існуючих правил розмежування доступу.

- **Санкціонований доступ до інформації** (*authorized access to information*) – доступ до інформації, що не порушує правила розмежування доступу.
- **Несанкціонований доступ до інформації** (*unauthorized access to information*) – доступ до інформації, що порушує правила розмежування доступу з використанням штатних засобів, що надаються засобами обчислювальної техніки або автоматизованими системами. Під штатними засобами розуміється сукупність програмного, мікропрограмного і технічного забезпечення засобів обчислювальної техніки або автоматизованих систем.

❒ **Правила розмежування доступу** (*access mediation rules*) – сукупність правил, що регламентують права доступу суб'єктів доступу до об'єктів доступу.

Шкідливі програми свідомо призначені для несанкціонованого знищення, блокування, модифікації, копіювання комп'ютерної інформації або нейтралізації засобів захисту комп'ютерної інформації.

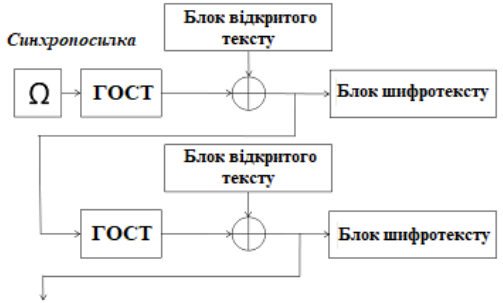
Корпорація Microsoft трактує термін «шкідлива програма» таким чином: «***Шкідлива програма*** *– це зазвичай використовуване як загальноприйнятий термін для позначення будь-якого програмного забезпечення, спеціально створеного для того, щоб заподіювати збиток окремому комп'ютеру, серверу або комп'ютерній мережі незалежно від того, чи є воно вірусом, шпигунською програмою і т.д.*».

Шкоди, що завдаються:

1. Перешкоди в роботі зараженого комп'ютера: починаючи від відкриття-закриття піддону CD/DVD-ROM і закінчуючи знищенням даних і поломкою апаратного забезпечення (поломками відомий, зокрема, **Win32.CIH**):
   - ❒ вандалізм – знищенняданих і устаткування;
   - ❒ шифрування файлів при кодовірусній атаці;
   - ❒ блокування антивірусних сайтів, антивірусного ПЗ і адміністративних функцій ОС з метою ускладнити лікування;
   - ❒ саботаж промислових процесів, керованих комп'ютером (цим відомий хробак **Stuxnet**).
3. Інсталяція іншого шкідливого ПЗ:
   - ❒ завантаження з мережі (*downloader*);
   - ❒ розпаковування іншої шкідливої програми, що вже міститься усередині файлу (*dropper*).
4. Крадіжка, шахрайство, здирство і шпигунство за користувачем. Для крадіжки може застосовуватися сканування жорсткого диска, реєстрація натиснень клавіш (*Keylogger*) і перенаправлення користувача на підробні сайти, в точності повторюючи вихідні ресурси:
   - ❒ постановка помилкових посилань, що ведуть на підробні веб-сайти з реєстрацією;
   - ❒ викрадення даних, що представляють цінність або таємницю;
   - ❒ крадіжка акаунтів різних служб (електронної пошти, месенджерів, ігрових серверів.). Акаунти застосовуються для розсилки спаму. Також через електронну пошту частенько можна дістати паролі від інших акаунтів, а віртуальне майно в MMOG – продати;
   - ❒ крадіжка акаунтів платіжних систем (реєстрація натиснень клавіш (*Keylogger*) з метою крадіжки паролів і номерів кредитних карток);
   - ❒ блокування комп'ютера, шифрування файлів користувача з метою шантажу і здирства грошових коштів. В більшості випадків після оплати комп'ютер або не розблоковується, або незабаром блокується другий раз;

попередній режим, проте гама формується на основі попереднього блоку зашифрованих даних, так що результат шифрування поточного блоку залежить також і від попередніх блоків. Тут при зміні одного біта шифртекста у відповідному блоці розшифрованого тексту міняється тільки один біт, так само зачіпається наступний блок відкритого тексту. При цьому усі інші блоки залишаються незмінними.

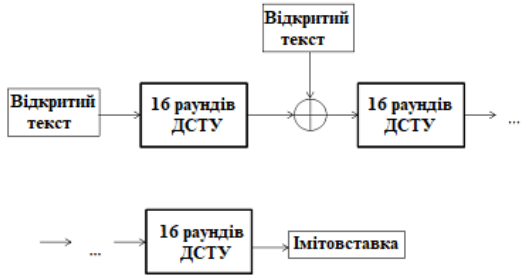
Алгоритм шифрування наступний:



1. Синхропосилка заноситься в регістри N_1 і N_2.
2. Вміст регістрів N_1 і N_2 шифрується відповідно до алгоритму простої заміни. Отриманий результат є 64-бітовим блоком гами.
3. Блок гами побітно складається по модулю 2 з блоком відкритого тексту. Отриманий шифротекст заноситься в регістри N_1 і N_2.
4. Операції 2-3 виконуються для блоків шифрування тексту, що залишилися.

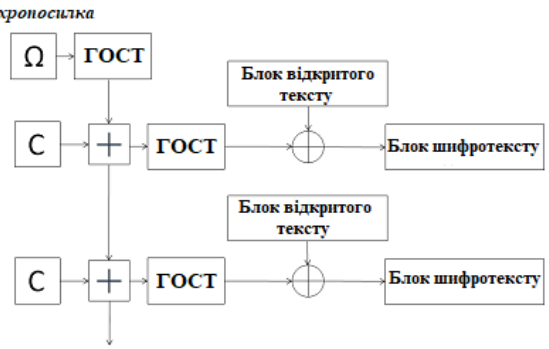
При використанні цього режиму слід мати на увазі, що синхропосилку не можна використовувати повторно (наприклад, при шифруванні логічне роздільних блоків інформації – мережевих пакетів, секторів жорсткого диска і т.п). Це обумовлено тим, що перший блок шифротекста отриманий усього лише складанням по модулю 2 із зашифрованою синхропосилкою; таким чином, знання усього лише 8 перших байт вихідного і шифрованого тексту дозволяють читати перші 8 байт будь-якого іншого шифртекста після повторного використання синхропосилки.

При роботі в **режимі вироблення імітовставки** створюється деякий додатковий блок, залежний від усього тексту і ключових даних. Цей блок використовується для перевірки того, що в шифртекст випадково або навмисно не були внесені спотворення. Це особливо важливо для шифрування в режимі гамування, де порушник може змінити конкретні біти, навіть не знаючи ключа; проте і при роботі в інших режимах ймовірні спотворення не можна виявити, якщо в передаваних даних немає надлишкової інформації. Імітовставка передається каналом зв'язку після зашифрованих блоків.



❒ при шифруванні однакових блоків відкритого тексту виходять однакові блоки шифртекста, що може дати певну інформацію криптоаналітику, тобто застосування ГОСТ 28147-89 в цьому режимі бажано лише для шифрування ключових даних.

При роботі ГОСТ 28147-89 в **режимі гамування** формується криптографічна гама, яка потім побітно складається за модулем 2 з вихідним відкритим текстом для отримання шифртекста. Таке шифрування позбавлене недоліків, властивих режиму простої заміни. Так, навіть ідентичні блоки відкритого тексту дають різний шифртекст. Крім того, гамма може бути вироблена заздалегідь, що відповідає роботі шифру в потоковому режимі.



Вироблення гами відбувається на основі ключа і так званою *синхропосилки*, яка задає початковий стан генератора. Алгоритм її вироблення наступний:

1. Синхропосилка шифрується з використанням алгоритму простої заміни, набутих значень записуються в допоміжні 32-розрядні регістри N_3 і N_4 – молодші й старші біти відповідно.
2. N_3 підсумовується по модулю 2^{32} з константою $C_2 = 1010101_{(16)}$.
3. N_4 підсумовується по модулю $2^{32} - 1$ з константою $C_1 = 1010104_{(16)}$.
4. N_3 і N_4 переписуються відповідно в N_1 і N_2, які потім шифруються з використанням алгоритму простої заміни. Отриманий результат є 64 бітами гамми.
5. Кроки 2-4 повторюються відповідно до довжини шифрованого тексту.

Для розшифровки необхідно виробити таку ж гаму, після чого побітно скласти її по модулю 2 із зашифрованим текстом. Очевидно, для цього треба використовувати ту ж синхропосилку, що і при шифруванні. При цьому, виходячи з вимог унікальності гами, не можна використовувати одну синхропосилку для шифрування декількох масивів даних. Як правило, синхропосилка тим або іншим чином передається разом з шифротекстом.

Особливість роботи ГОСТ 28147-89 в режимі гаммирования полягає в тому, що при зміні одного біта шифротекста змінюється тільки один біт розшифрованого тексту. З одного боку, це може чинити позитивний вплив на завадозахищеність; з іншої – порушник може внести деякі зміни до тексту, навіть не розшифровувавши його.

Алгоритм **шифрування гамуванням із зворотним зв'язком** схожий на

❒ використання телефонного модему для здійснення дорогих дзвінків, що спричиняє за собою значні суми в телефонних рахунках;

❒ платне ПЗ, імітуюче, наприклад, антивірус, але нічого корисного не робить (*fraudware* або *scareware* – помилкові антивіруси).

5. Використання ресурсів зараженого комп'ютера в злочинних цілях:

❒ отримання несанкціонованого (і/або дармового) доступу до ресурсів самого комп'ютера або третіх ресурсів, доступних через нього, у тому числі пряме управління комп'ютером (*backdoor*);

❒ організація на комп'ютері відкритих релеєв і загальнодоступних проксі-серверів;

❒ використання заражених комп'ютерів (у складі ботнету) для проведення DDoS-атак;

❒ збір адрес електронної пошти і поширення спаму, у тому числі у складі ботнета;

❒ накрутка електронних голосувань, клацань по рекламних банерах;

❒ генерація монет платіжної системи **Bitcoin**.

6. Інші види протиправної діяльності:

❒ поширення інших шкідливих програм (наприклад, «троянських коней», що поширюють віруси);

❒ пряме управління комп'ютером;

❒ дезактивація антивірусів і брандмауерів.

Ознаки зараження комп'ютерної системи:

❒ автоматичне відкриття вікон з незнайомим вмістом при запуску комп'ютера;

❒ блокування доступу до офіційних сайтів антивірусних компаній або до сайтів, що роблять послуги з «лікування» комп'ютерів від шкідливих програм;

❒ поява нових невідомих процесів у виведенні диспетчера завдань (наприклад, вікні «Процеси» диспетчера завдань Windows);

❒ поява в гілках реєстру, що відповідають за автозапуск, нових записів;

❒ заборона на зміну налаштувань комп'ютера в обліковому записі адміністратора;

❒ неможливість запустити виконуваний файл (видається повідомлення про помилку);

❒ поява спливаючих вікон або системних сполучень з незвичним текстом, у тому числі що містять невідомі веб-адреси і назви;

❒ перезапуск комп'ютера під час старту якої-небудь програми;

❒ випадкове і/або безладне відключення комп'ютера;

❒ випадкове аварійне завершення програм.

Проте, слід враховувати, що незважаючи на відсутність симптомів, комп'ютер може бути заражений шкідливими програмами.

Комп'ютерним хуліганством займаються:

1) студенти і школярі, які тільки що вивчили мову програмування і хочуть спробувати свої сили. Такі віруси пишуться тільки для самоствердження їх авторів;

2) молодь (частіше – студенти), які ще не повністю оволоділи мистецтвом програмування. Єдина причина, що штовхає їх на написання вірусів, це комплекс меншовартості, який компенсується комп'ютерним хуліганством. З-під пера подібних «умільців» часто виходять віруси вкрай примітивні й з великим числом помилок («студентські» віруси);

3) професійні програмісти, що створюють ретельно продумані та налагоджені «професійні» віруси, які використовують досить оригінальні алгоритми проникнення в системні області даних, помилки в системах безпеки операційних середовищ, соціальний інжиніринг та інші хитрощі;

4) «дослідники» – досить кмітливі програмісти, які займаються винаходом принципово нових методів зараження, приховування, протидії антивірусам і т.д. Вони ж придумують способи впровадження в нові операційні системи.

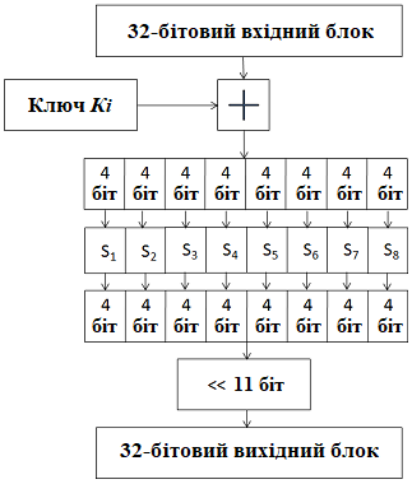
Дрібне злодійство.З появою і популяризацією платних інтернет-сервісів (пошта, веб, хостинг) комп'ютерний андеграунд починає проявляти підвищений інтерес до отримання доступу в мережу за чужий рахунок, тобто за допомогою крадіжки логінів і паролів шляхом застосування спеціально розроблених троянських програм. «Дрібними злодюжками» створюються троянські програми інших типів: крадуть реєстраційні дані та ключові файли різних програмних продуктів, що використовують ресурси заражених комп'ютерів в інтересах свого «господаря» і т.п. В останні роки постійно збільшується число троянських програм, що крадуть персональну інформацію з мережевих ігор (ігрову віртуальну власність) з метою її несанкціонованого використання або перепродажу.

Кримінальний бізнес. Найбільш небезпечну категорію вірусописьменників становлять групи хакерів, які усвідомлено створюють шкідливі програми з корисливою метою. Для цього вони створюють вірусні і троянські програми, які крадуть коди доступу до банківських рахунків, нав'язливо

32-бітних блоків: $K_1 \ldots K_8$. Ключі $K_9 \ldots K_{24}$ є циклічним повторенням ключів $K_1 \ldots K_8$ (нумеруються від молодших бітів до старших). Ключі $K_{25} \ldots K_{32}$ є ключами $K_8 \ldots K_1$, що йдуть у зворотному порядку.

Після виконання усіх 32 раундів алгоритму, блоки A_{33} і B_{33} склеюються в шифрограму.

Розшифрування виконується так само, як і зашифровування, але інвертується порядок підключівK_i.



Функція$f(A_i, K_i)$обчислюється таким чином: A_iі K_iскладаються по модулю 2^{32}. Результат розбивається на вісім 4-бітових підпослідовностей, кожна з яких надходить на вхід свого *вузла таблиці замін* (у порядку зростання старшинства бітів), званого *S-блоком*. Загальна кількість *S*-блоків – вісім, тобто стільки ж, скільки і підпослідовностей. Кожен *S-блок* являє собою перестановку чисел від 0 до 15. Перша 4-бітова підпослідовність потрапляє на вхід першого *S*-блоку, друга – на вхід другого і т.д.

Якщо *S*-блок виглядає так: **1, 15, 13, 0, 5, 7, 10, 4, 9, 2, 3, 14, 6, 11, 8, 12** і на вході *S*-блоку 0, то на виході буде 1, якщо 4, то на виході буде 5, якщо на вході 12, то на виході 6 і т.д. Виходи всіх восьми S-блоків об'єднуються в 32-бітове слово, потім все слово циклічно зсувається вліво (до старших розрядів) на 11 бітів.

Усі вісім *S*-блоків можуть бути різними. Фактично, вони можуть бути додатковим ключовим матеріалом, але частіше є параметром схеми, загальним для певної групи користувачів. Наприклад, в ГОСТ Р 34.11-94 для цілей тестування наведені такі *S*-блоки:

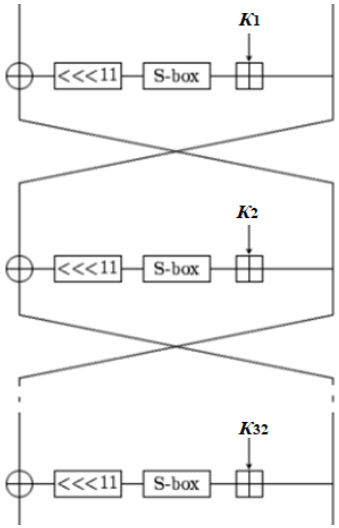
Номер S-блоку	Значення															
1	4	10	9	2	13	8	0	14	6	11	1	12	7	15	5	3
2	14	11	4	12	6	13	15	10	2	3	8	1	0	7	5	9
3	5	8	1	13	10	3	4	2	14	15	12	7	6	0	9	11
4	7	13	10	1	0	8	9	15	14	4	6	12	11	2	5	3
5	6	12	7	1	5	15	13	8	4	10	9	14	0	3	11	2
6	4	11	10	0	7	2	1	13	3	6	8	5	9	12	15	14
7	13	11	4	1	3	15	5	9	0	10	14	7	6	8	2	12
8	1	15	13	0	5	7	10	4	9	2	3	14	6	11	8	12

Режим простої заміни має наступні недоліки:

❒ може застосовуватися тільки для шифрування відкритих текстів з довжиною, кратною 64 біт;

Нині одним із найпоширеніших алгоритмів симетричного шифрування є **AES** (*Advanced Encryption Standard*), також відомий під назвою **Rijndael** і запропонований бельгійськими криптографами Й.Дайменом та В.Рейменом – симетричний алгоритм блочного шифрування (розмір блока 128 біт, ключ 128/192/256 біт, число раундів 10/12/14 (залежить від розміру ключа)), прийнятий як американський стандарт шифрування урядом США 26 травня 2002 року. Підтримка AES введена фірмою Intel в сімейство процесорів x86 починаючи з Intel Core i7-980X Extreme Edition, а потім на процесорах Sandy Bridge.

У червні 2003 р. Агентство національної безпеки США ухвалило, що шифр AES є досить надійним, щоб використовувати його для захисту відомостей, що становлять державну таємницю. Аж до рівня SECRET було дозволено використовувати ключі завдовжки 128 біт, для рівня TOP SECRET вимагалися ключі завдовжки 192 і 256 біт.



В країнах колишнього СРСР (у тому числі, в Україні ДСТУ ГОСТ 28147:2009) використовуєтьсясиметричний блоковий шифр, визначений стандартом «**ГОСТ 28147-89***Системи обробки інформації. Захист криптографічний. Алгоритм криптографічного перетворення*». Він рекомендований до використання для захисту будь-яких даних, представлених у вигляді двійкового коду. Формувався з урахуванням світового досвіду, і зокрема, при його розробці були взяті до уваги недоліки алгоритму DES.

Характеристики алгоритму:

- розмір ключа – 256 біт;
- розмір шифруємого/розшифруємого блоку – 64 біт;
- число раундів – 32/16;
- тип – мережа Фейстеля;
- режими роботи:
  - простої заміни;
  - гамування;
  - гамування із зворотним зв'язком;
  - режим вироблення імітовставки.

У режимі **простої заміни** 64-бітовий блок відкритого тексту спочатку розбивається на дві половини (молодші біти – ***A***, старші біти – ***B***). На *i*-му циклі використовується підключ K_i:

$$A_{i+1} = B_i \oplus f(A_i, K_i); \quad B_{i+1} = A_i.$$

Для генерації підключів вихідний 256-бітний ключ розбивається на вісім

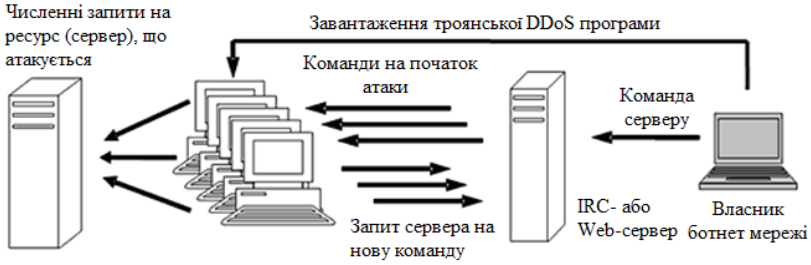
рекламують будь-які товари або послуги, несанкціоноване використовують ресурси зараженого комп'ютера.

Кримінальний бізнес

- **Обслуговування спам-бизнеса**
- **Розподілені мережеві атаки (DDoS -атаки)**
- **Створення мереж "зомбі-машин"**
- **Дзвінки на платні телефонні номери або посилка платних SMS-повідомлень**
- **Крадійство інтернет-грошей**
- **Крадійство банківської інформації**
- **Крадійство конфіденційної інформації**
- **Кібершантаж**
- **Розробка «засобів доставки»**
- **Точкові таргетовані атаки**
- **Інші види кримінальної діяльності**

Як приклад кримінального бізнесу розглянемо організацію розподіленої мережевої атаки (**DDoS-атаки**, *Distributed Denial of Service* - розподілена відмова в обслуговуванні). Мережеві ресурси (наприклад, веб-сервера) мають обмежені можливості по кількості одночасно обслуговуваних запитів – ця кількість обмежена як потужностями самого сервера, так і шириною каналу, з яким він підключений до Інтернету. Якщо кількість запитів перевищує допустиму, то робота з сервером значно затримується або взагалі запити користувачів будуть враховані.

Використовуючи цей факт, комп'ютерні порушники ініціюють «шкідливі» запити на ресурс, що атакується, причому кількість таких запитів багаторазово перевищує можливість ресурсу-жертви. За допомогою «зомбі-мережі» достатнього розміру організовується масована DDoS-атака на один або декілька інтернет-ресурсів, що призводить до відмови атакованих вузлів мережі. В результаті звичайні користувачі не в змозі отримати доступ до атакованого ресурсу. Зазвичай під удар потрапляють інтернет-магазини, букмекерські контори, інші компанії, бізнес яких безпосередньо залежить від працездатності своїх інтернет-сервісів. Частіше всього розподілені атаки здійснюються або з метою «завалити» бізнес-конкурента, або з наступною вимогою грошової винагороди за припинення атаки.



Створення мереж «зомбі-машин». Для розгортання подібних мереж створюються спеціалізовані троянські програми – «боти», які централізовано управляються віддаленим «господарем». Цей троянець впроваджується в тисячі, десятки тисяч або навіть мільйони комп'ютерів. В результаті «господар зомбі-мережі» (або «ботнет-мережі») отримує доступ до ресурсів всіх заражених комп'ютерів і використовує їх в своїх інтересах. Іноді такі мережі «зомбі-машин» надходять на чорний інтернет-ринок, де купуються спамерами або здаються їм в оренду.

Умови існування шкідливих програм

Операційна система або застосування можуть піддатися вірусному нападу, якщо вони *мають можливість запустити програму, яка не є частиною самої системи або застосування*. Даної умові задовольняють усі популярні «настільні» ОС, багато офісних застосувань, графічні редактори, системи проектування та інші програмні комплекси, що мають вбудовані скриптові мови.

Комп'ютерні віруси, хробаки, троянські програми існують для десятків ОС і застосувань. У той же час є величезна кількість інших ОС і додатків, для яких шкідливі програми поки не виявлені. Що є причиною існування шкідливих програм в одних системах і відсутності їх в інших?

Причиною появи подібних програм в конкретній ОС або застосуванню є одночасне виконання таких умов:

1) популярність, широке поширення даної операційної системи;
2) документування – наявність різноманітної та досить повної документації по системі;
3) незахищеність системи або існування відомих уразливостей в її безпеки і застосуваннях.

Кожна перерахована умова є *необхідною*, а виконання всіх умов одночасно є *достатнім* для появи різноманітних шкідливих програм.

Чим **популярніше ОС або застосування**, тим частіше вона буде жертвою вірусної атаки. Практика це підтверджує – розподіл кількості шкідливого програмного забезпечення для Windows, Linux і MacOS практично збігається з частками ринку, які займають ці операційні системи.

Наявність повної документації необхідно для існування вірусів з природної причини: створення програм (включаючи вірусні) неможливо без технічного опису використання сервісів операційної системи і правил створення програмного забезпечення.

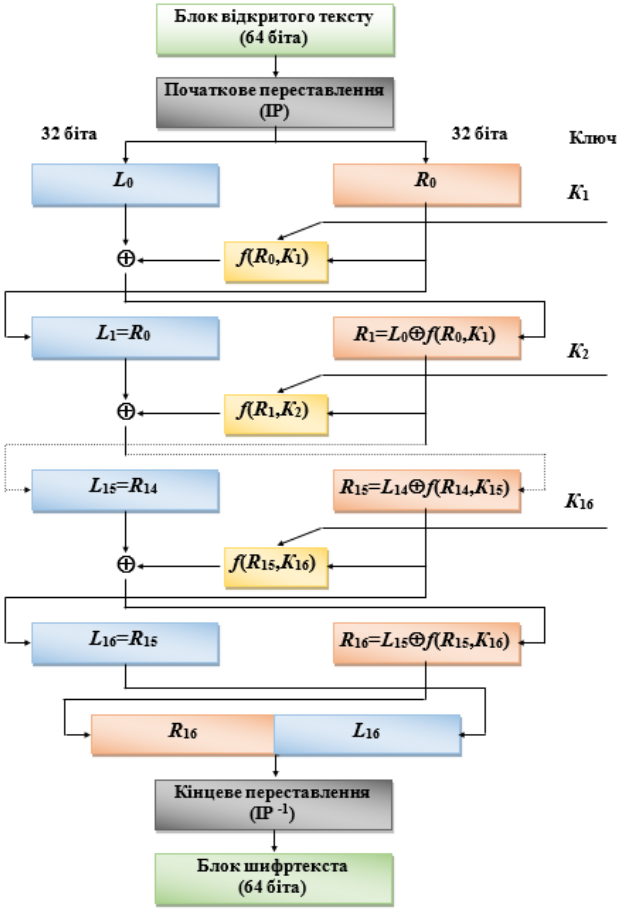
Уразливостями називають помилки («дірки») в ПЗ, як програмістські (помилка в коді програми, що дозволяє вірусу «пролізти в дірку» і захопити контроль над системою), так і логічні (можливість проникнення в систему легальними, іноді навіть документально оформленими методами). Якщо ОС або в її додатках існують відомі уразливості, то така система відкрита для вірусів, який би захищеної вона не була.

Під **захищеністю** системи розуміються архітектурні рішення, які не дозволяють новому (невідомому) застосуванню отримати повний або достатньо

❒ **DES – EDE3**: 3 DES операцій шифрування-розшифрування-шифрування з 3 різними ключами;

❒ **DES – EEE2** і **DES – EDE2**: як і попередні, за винятком того, що перша і третя операції використовують однаковий ключ.

Детальна схема шифрування алгоритму DES



Найпопулярніший тип при використанні **3DES** - це **DES – EDE3**, для нього алгоритм виглядає так:

❒ **зашифрування**: $C = E_{k3}\left(E_{k2}^{-1}\left(E_{k1}(M)\right)\right)$

❒ **розшифрування**: $M = E_{k1}^{-1}\left(E_{k2}\left(E_{k3}^{-1}(C)\right)\right)$

DES був національним стандартом США в 1977-2000 рр., але нині **DES** використовується (з ключем довжини 56 біт) тільки для застарілих систем, найчастіше використовують його криптостійкий вид (**3DES**, **DESX**).

3) *стійкість*: якщо ключ невідомий, то немає способу розкриття шифру, окрім «лобової атаки».

При шифруванні даних в криптосистемі DES може бути використаний один з чотирьох режимів роботи :

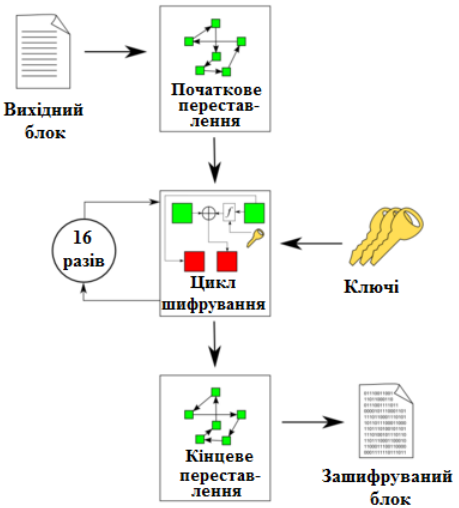
1) електронна кодова книга (проста заміна) (**ECB** – *Electronic Codebook Mode*);

2) зворотний зв'язок за шифртекстом (**CFB** – *Cipher Feedback Mode*);

3) зчеплення блоків шифру (**CBC** – *Cipher Block Chaining Mode*);

4) зворотний зв'язок за виходом (**OFB** – *Output Feedback Mode*).

Схема шифрування алгоритму DES у режимі простої заміни (ECB)



У режимах **ECB** і **OFB** спотворення при передачі одного 64-бітового блоку шифротекста C_i призводить до спотворення після розшифрування тільки відповідного відкритого блоку M_i, тому такі режими використовується для передачі каналами зв'язку з великим числом спотворень.

Із-за невеликого числа можливих ключів (всього 2^{56}), з'являється можливість їх повного перебору на швидкодіючій обчислювальній техніці за реальний час. У 1998 році **Electronic Frontier Foundation** використовуючи спеціальний комп'ютер **DES-Cracker**, вдалося зламати DES за 3 дні.

Щоб збільшувати криптостойкість **DES** з'являються декілька варіантів: **2DES**, **3DES**, **DESX**, **G-DES**.

Так, методи **2DES** і **3DES** засновані на DES, але збільшують довжину ключів (**2DES** – 112 біт, **3DES** – 168 біт) і тому збільшується криптостойкість.

Наприклад, схема **3DES** має вигляд $DES(k_3, DES((k_2, DES(k_1, M)))$, де k_1, k_2, k_3 ключі для кожного шифру **DES**. Це варіант відомий як в **EEE** оскільки три **DES** операції є шифруванням. Існує 3 типи алгоритму **3DES**:

- **DES – EEE3**: шифрується три рази з 3 різними ключами;

широкий доступ до файлів на диску (включно з іншими програмами) і потенційно небезпечним сервісам системи. Подібне обмеження фактично блокує будь-яку вірусну активність, але при цьому, природно, накладає суттєві обмеження на можливості звичайних програм.

Прикладів широко відомих захищених багатофункціональних і відкритих операційних систем і застосувань, на жаль, немає.

3.1.2 Способи проникнення шкідливих програм в систему

Необхідним для вірусописьменників і кіберзлочинців завданням є впровадження вірусу, хробака або троянської програми в комп'ютер-жертву або мобільний телефон. Досягається ця мета різними способами, які діляться на дві основні категорії:

- *соціальна інженерія* (*social engineering*);
- *технічні прийоми впровадження* шкідливого коду в систему, що заражається, без відома користувача.

Часто ці способи використовуються одночасно. При цьому так само часто використовуються спеціальні заходи з протидії антивірусним програмам.

Соціальна інженерія – методи несанкціонованого доступу до інформації або систем зберігання інформації без використання технічних засобів. Засновані на використанні слабкостей людського чинника: примушують користувача запустити заражений файл або відкрити посилання на заражений вебсайт.

Завдання хакерів – притягнути увагу користувача до зараженого файлу (чи HTTP-посиланню на заражений файл), зацікавити користувача, змусити його кликнути по файлу (чи по посиланню на файл). «Класикою жанру» є поштовий хробак **LoveLetter** (травень 2000 р.), що досі зберігає лідерство за масштабом завданого фінансового збитку. Повідомлення, яке хробак виводив на екран, виглядало таким чином:

На визнання «**I LOVE YOU**» зреагували дуже багато, і в результаті поштові сервера великих компаній не витримали навантаження – хробак розсилав свої копії по усіх контактах з адресної книги при кожному відкритті вкладеного VBS-файлу.

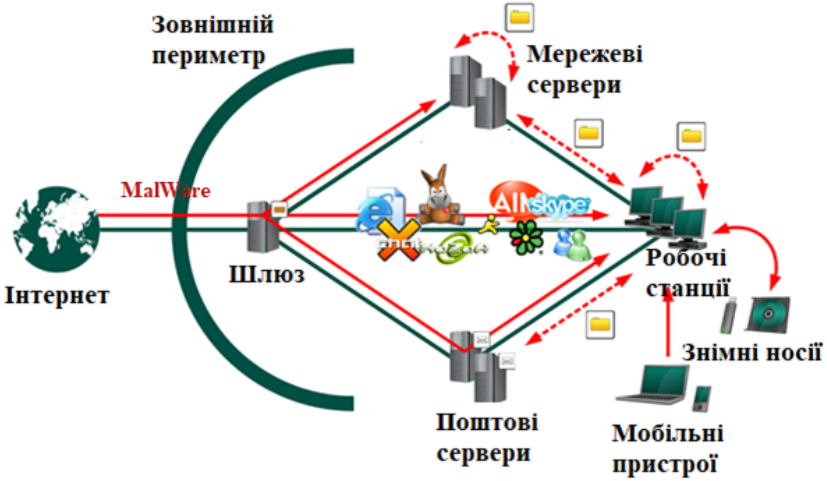
Поштовий хробак **Mydoom**, що «рвонув» в Інтернеті в січні 2004 р., використовував тексти, що імітують технічні повідомлення поштового сервера.

Варто також згадати хробак **Swen**, який видавав себе за повідомлення від компанії Microsoft і маскувався під патч, що усуває ряд нових уразливостей в Windows

Останнім часом особливу популярність отримали не файли, вкладені в лист, а посилання на файли, розташовані на зараженому сайті. Потенційній жертві вирушає повідомлення – поштове, через ICQ або інший пейджер, (у

разі мобільних вірусів звичайним способом доставки служить SMS-повідомлення). Повідомлення містить який-небудь привабливий текст, що примушує користувача кликнути на посилання. Цей спосіб проникнення в комп'ютери-жертви на сьогодні є найпопулярнішим, оскільки дозволяє обходити пильні антивірусні фільтри на поштових серверах.Використовуються також можливості файлообмінних мереж (P2P-мережи).

Типова схема зараження шкідливим ПЗ



Технології впровадженнявикористовуються порушниками для впровадження в систему шкідливого коду потайно, не привертаючи уваги власника комп'ютера. Здійснюється це через уразливості в системі безпеки ОС і в програмному забезпеченні. Наявність уразливостей дозволяє мережевому хробаку або троянській програмі проникнути в комп'ютер-жертву і самостійно запустити себе на виконання.

Наприклад, уразливостями в поштових клієнтах **Outlook** користувалися поштові хробаки **Nimda** і **Aliz**. Для того, щоб запустити файл хробака, досить було відкрити заражений лист або просто навести на нього курсор у вікні попереднього перегляду.

Шкідливі програми активно використовують уразливості в мережевих компонентах операційних систем. Для свого поширення такими можливостями користувалися хробаки**CodeRed**, **Sasser**, **Slammer**, **Lovesan** (**Blaster**) та ін.

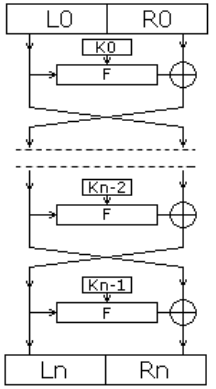
Останніми роками одним з найбільш популярних способів зараження стало впровадження шкідливого коду через веб-сторінки. При цьому часто використовуються уразливості в Інтернет-браузерах. На веб-сторінку поміщається заражений файл і скрипт-програма, яка використовує уразливість в браузері. При заході користувача на заражену сторінку спрацьовує скрипт-програма, яка через уразливість закачує заражений файл на комп'ютер і запус-

5. Результат буде використаний в наступному раунді в ролі «лівого підблоку» L_1: $L_1 := x$.

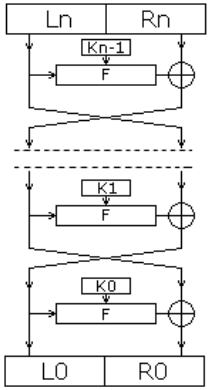
6. «Лівий підблок» L_0 поточного раунду буде використаний в наступному раунді в ролі «правого підблоку» R_1: $R_1 := L_0$.

7. За яким-небудь математичним правилом обчислюється раундовий ключ K_1 – ключ, який використовуватиметься в наступному раунді.

8. Перераховані операції виконуються $N - 1$ раз, де N – кількість раундів у вибраному алгоритмі шифрування. При цьому між переходами від одного раунду до іншого змінюються ключі: K_0 замінюється на K_1, K_1 – на K_2 і т.д.



Шифрування



Розшифрування

Розшифрування

Розшифрування інформації відбувається так само, як і шифрування, з тим лише виключенням, що ключі використовуються в зворотному порядку, тобто не від першого до *N*-го, а від *N*-го до першого.

Прикладом застосування комбінованих шифрів на основі мережі Фейстеля з 16 раундами є **DES** (*Data Encryption Standard*) – національний стандарт США криптографічного закриття даних, розроблений компанією IBM в 1974 р. за замовленням Національного бюро стандартів (НБС) США і прийнятий як стандарт шифрування даних в державних і приватних установах в 1976 р. У 1987 р. був прийнятий міжнародний стандарт на основі DES (**ISO 8372**). Однією з головних вимог НБС була можливість публікації алгоритму без збитку для його стійкості.

У DES відкритий блок тексту M, криптограма C і ключ K– двійкові послідовності завдовжки відповідно $M = 64$, $C = 64$ і $K = 56$. DES є підстановкою в гігантському алфавіті, що містить $2^{64} \approx 10^{19}$ «символів».

При виборі алгоритму шифрування розробники виходили з трьох вимог:

1) *можливість розшифрування*: для будь-якого ключа Kрізним блокам відкритого повідомлення M' і M'' відповідають різні блоки криптотекста C'і C'';

2) *ефективність*: шифрування і розшифрування здійснюються швидко;

❒ **перемішування** – це використання таких шифруючих перетворень, які ускладнюють відновлення взаємозв'язку статистичних властивостей відкритого і шифрованого текстів.

Поширений спосіб розсіювання і перемішування полягає у використанні *складеного шифру*, реалізованого у вигляді послідовності простих шифрів, кожен з яких вносить свій вклад до сумарного розсіювання і перемішування. При багатократному чергуванні простих переставлень і підстановок можна отримати дуже стійкий шифр з хорошим розсіюванням і перемішуванням.

Стійкість комбінованого шифрування S_kне нижча за добуток стійкості використовуваних способів S_i, тобто

$$S_K \geq \prod_t S_t$$

Комбінувати можна будь-які методи шифрування і в будь-якій кількості, проте на практиці найбільшого поширення набули наступні комбінації:

1) підстановка + гамування;
2) переставлення + гамування;
3) гамування + гамування;
4) підстановка + переставлення.

У симетричному шифруванні широкого поширення отримала **мережа Фейстеля** – один з методів побудови блокових шифрів. У 1971 р. **Горст Фейстель (1915-1990)** – співробітник компанії IBM (США) запатентував пристрої, що реалізовують алгоритми шифрування, які пізніше дістали назву «*Lucifer*».

Мережа є структурою, що складається з однотипних **комірок Фейстеля**, які багаторазово повторюються. На вхід кожної комірки поступають дані і ключ, а на виході отримують змінені дані і змінений ключ. Ключ вибирається залежно від алгоритму шифрування/розшифрування і міняється при переході від однієї комірки до іншої. При шифруванні і розшифруванні виконуються одні і ті ж операції; відрізняється тільки порядок застосування ключів. Ітеративна структура алгоритму дозволяє створювати прості програмні й апаратні реалізації.

Алгоритм шифрування

1. Інформація розбивається на блоки однакової (фіксованою) довжини. Отримані блоки називаються вхідними. Як правило довжина блоку є ступеню 2, наприклад, складає 64 біта або 128 біт.

2. Вибраний блок ділиться на два підблоки однакового розміру – «лівий» (L_0) і «правий» (R_0).

3. «Лівий підблок» змінюється функцією f з використанням раундового ключа K_0: $x := f(L_0, K_0)$.

4. Результат складається по модулю 2 («XOR») з «правим підблоком»: $x := x \oplus R_0$.

кає його там на виконання. В результаті для зараження великого числа комп'ютерів досить заманити як можна більше число користувачів на таку вебсторінку. Це досягається різними способами, наприклад, розсилкою спаму з вказівкою адреси сторінки, розсилкою аналогічних повідомлень через Інтернет-пейджери, іноді для цього використовують навіть пошукові машини. На зараженій сторінці розміщується різноманітний текст, який рано чи пізно обраховується пошуковими машинами, – і посилання на цю сторінку опиняється в списку інших сторінок в результатах пошуку.

Окремим класом коштують троянські програми, які призначені для скачування і запуску інших троянських програм. Зазвичай цей троян тим або іншим чином (наприклад, використовуючи чергову уразливість в системі) «підсовується» на комп'ютер-жертву, а потім вже самостійно викачують з Інтернету і встановлюють в систему інші шкідливі компоненти.

Досить часто порушники використовуються відразу обидва методи: метод соціальної інженерії – для привертання уваги потенційної жертви, а технічний – для збільшення ймовірності проникнення зараженого об'єкту в систему.

Наприклад, поштовий хробак **Mimail** поширювався як вкладення в електронний лист. Для того, щоб користувач звернув увагу на лист, в нього вставлявся спеціально оформлений текст, а для запуску копії хробака з вкладеного в лист ZIP-архіву – уразливість в браузері Internet Explorer. В результаті при відкритті файлу з архіву хробак створював на диску свою копію і запускав її на виконання без яких або системних попереджень або додаткових дій користувача.

Способи протидії антивірусним програмам з боку шкідливих програм

Оскільки мета порушника – впровадити шкідливий код в комп'ютери-жертви, то для цього їм необхідно не лише змусити користувача запустити заражений файл або проникнути в систему через яку-небудь уразливість, але і непомітно проскочити повз встановлений антивірусний фільтр. Тому порушники цілеспрямовано борються з антивірусними програмами. Використовувані ними технічні прийоми дуже різноманітні.

❒ **Упаковка і шифрування коду.** Значна частина сучасних комп'ютерних хробаків і троянських програм упаковані або зашифровані тим або іншим способом. Для цього створюються спеціальні утиліти упаковки і шифрування. Наприклад, шкідливими виявилися абсолютно усі файли, що оброблені утилітами **CryptExe**, **Exeref**, **PolyCrypt** і деякими іншими.

Для детектування подібних шкідливих програм антивірусам доводиться або додавати нові методи розпаковування і розшифровки, або додавати сигнатури на кожен зразок шкідливої програми, що знижує якість детектування, оскільки не завжди усі можливі зразки модифікованого коду опиняються в руках антивірусної компанії.

❒ **Мутація коду** – розбавлення коду трояна «сміттєвими»» інструкціями. В результаті функціонал троянської програми зберігається, але значно міняється її «зовнішній вигляд». Періодично зустрічаються випадки, коли мутація коду відбувається в режимі реального часу – при кожному скачуванні троянської програми із зараженого веб-сайту. Прикладом застосування цієї технології є поштовий хробак **Warezov**, декілька версій якого викликали значні епідемії в другій половині 2006 р.

❒ **Приховання своєї присутності.**Руткит-технології, використовувані в троянських програмах, здійснюють перехоплення і підміну системних функцій, завдяки яким заражений файл не видно ні штатними засобами ОС, ні антивірусними програмами. Іноді також ховаються гілки реєстру, в яких реєструється копія трояна, та інші системні області комп'ютера. Ці технології активно використовуються, наприклад, троянцем-бекдором **HacDef**.

❒ **Зупинка роботи антивіруса і системи отримання оновлень антивірусних баз.** Багато троянських програм і мережеві хробаки роблять спеціальні дії проти антивірусних програм – шукаютьїх в списку активних застосувань і намагаються зупинити їх роботу, псують антивірусні бази даних, блокують отримання оновлень і т.п. Антивірусним програмам доводиться захищати себе адекватними способами – стежитиза цілісністю баз даних, ховати від троянів свої процеси і т.п.

❒ **Приховання свого коду на веб-сайтах.** Адреси веб-сторінок, на яких присутні троянські файли, рано чи пізно стають відомі антивірусним компаніям. Природно, що подібні сторінки потрапляють під пильну увагу антивірусних аналітиків – вміст сторінки періодично викачується, нові версії троянських програм заносяться в антивірусні оновлення. Для протидії цьому веб-сторінка модифікується спеціальним чином – якщо запит йде з адреси антивірусної компанії, то викачується який-небудь нетроянський файл замість троянського.

дкритого тексту здійснюється згідно з вибраним аналітичним виразом.

Досить надійне закриття інформації може бути забезпечене при використанні для шифрування деяких аналітичних перетворень, зокрема, методів алгебри матриць. Наприклад, множення матриці на вектор за правилом

$$\left\|a_{ij}\right\|\cdot b_j = c_j = \sum_i a_{ij}\cdot b_j .$$

Якщо матрицю $\left\|a_{ij}\right\|$ використовувати як ключ, а замість компонента вектору b_j підставити символи вихідного тексту, то компоненти вектору c_j будуть символами зашифрованого тексту.

Наприклад, як ключ вибрана квадратна матриця:

$$\begin{Vmatrix} 14 & 8 & 3 \\ 8 & 5 & 2 \\ 3 & 2 & 1 \end{Vmatrix}.$$

Замінимо букви алфавіту цифрами, відповідними їх порядковому номеру в алфавіті, починаючи з нуля (**А – 0, Б – 1, ..., Я – 33**) і зашифруємо уривок тексту **ГАПАЙА...**, якому відповідає числова послідовність 3, 0, 19, 0, 12, 0.

За алгоритмом шифрування виконаємо необхідні дії:

$$\begin{Vmatrix} 14 & 8 & 3 \\ 8 & 5 & 2 \\ 3 & 2 & 1 \end{Vmatrix} \times \begin{Vmatrix} 3 \\ 0 \\ 19 \end{Vmatrix} = \begin{Vmatrix} 99 \\ 62 \\ 28 \end{Vmatrix}, \quad \begin{Vmatrix} 14 & 8 & 3 \\ 8 & 5 & 2 \\ 3 & 2 & 1 \end{Vmatrix} \times \begin{Vmatrix} 0 \\ 12 \\ 0 \end{Vmatrix} = \begin{Vmatrix} 96 \\ 60 \\ 24 \end{Vmatrix}$$

Таким чином, зашифрований текст виглядатиме таким чином:

99, 62, 28, 96, 60, 24.

Дешифрування здійснюється з використанням того ж правила множення матриці на вектор, тільки як основа береться матриця, зворотна тій, за допомогою якої здійснювалося закриття, а як вектор-співмножник – відповідна кількість символів шифртекста. Результатом будуть цифрові еквіваленти знаків відкритого тексту.

Комбіновані методи шифрування

Підвищення стійкості криптосистеми за рахунок збільшення довжини ключа приводить, як правило, до істотного ускладнення самого процесу шифрування і збільшення витрат ресурсів (часу, апаратних засобів), зменшення пропускної спроможності каналів зв'язку і т.п. Для побудови стійкого шифру алгоритм шифрування повинен забезпечити два загальні принципи, сформульованих К.Шенноном: *розсіювання* і *перемішування*:

❒ **розсіювання** – це поширення впливу одного знаку відкритого тексту на багато знаків шифртекста, що дозволяє приховати статистичні властивості відкритого тексту;

За переказами, першим криптоаналітиком, що зламав шифр «Скитала» був старогрецький філософ Арістотель (384 – 322 до Р.Х.)

ЦЕШИФРДРЕВНЬОЇСПАРТИ

	Ц	Е	Ш	И	Ф	
	Р	Д	Р	Е	В	
	Н	Ь	О	Ї	С	
	П	А	Р	Т	И	

ЦРНПЕДЬАШРОРИЕЇТФВСИ

Шифрування простим переставленням здійснюється таким чином:

1) вибирається ключове слово з символами, що не повторюються;

2) шифрований текст записується послідовними рядками під символами ключового слова;

3) зашифрований текст виписується колонками в тій послідовності, в якій розташовуються в алфавіті букви ключа (чи в порядку дотримання цифр в натуральному ряду, якщо ключ цифровий).

Приклад: зашифрувати повідомлення: «БУДЬТЕ ОБЕРЕЖНІ З ПРЕДСТАВНИКОМ ФІРМИ «ФЕНІКС», використовуючи ключ 5 8 1 3 7 4 6 2. У вихідному тексті замість пропусків використовується буква *α*.

	Ключ							
	5	**8**	**1**	**3**	**7**	**4**	**6**	**2**
Відкритий текст, записаний рядами	**Б**	**У**	**Д**	**Ь**	**Т**	**Е**	***α***	**О**
	Б	**Е**	**Р**	**Е**	**Ж**	**Н**	**І**	***α***
	З	***α***	**П**	**Р**	**Е**	**Д**	**С**	**Т**
	А	**В**	**Н**	**И**	**К**	**О**	**М**	***α***
	Ф	**І**	**Р**	**М**	**И**	***α***	**Ф**	**Е**
	Н	**І**	**К**	**С**	***α***	***α***	***α***	***α***

Записати текст по колонках, використовуючи порядок старшинства букв (чи чисел) в ключі – отримаємо шифрований текст:

ДРПНРК О*α*Т*α*Е*α* ЬЕРИМС ЕНДО*αα*ББЗАФН
**
*α*ІСМФ*α*ТЖЕКИ*α*УЕ*α*ВІІ**

Розшифрування виконується в наступному порядку:

1) підрахувати число знаків в зашифрованому тексті та розділити його на число знаків ключа;

2) виписати ключове слово і під його знаками у відповідній послідовності виписати символи зашифрованого тексту в певній вище кількості;

3) по рядках таблиці прочитати вихідний текст.

Слабкість шифрування простим переставленням обумовлюється тим, що при великій довжині вихідного тексту в зашифрованому тексті можуть проявитися закономірності символів ключа. Для усунення цього недоліку можна міняти ключ після зашифровування певного числа знаків. При досить частій зміні ключа стійкість шифрування можна істотно підвищити.

Шифрування за допомогою аналітичних перетворень

У шифрах, заснованих на **аналітичних перетвореннях**, шифрування ві-

❒ **Атака кількістю** – генерація і поширення в Інтернеті великої кількості нових версій троянських програм за короткий проміжок часу. В результаті антивірусні компанії виявляються «завалені» новими зразками, на аналіз яких вимагається час, що дає зловмисному коду додатковий шанс для успішного впровадження в комп'ютери.

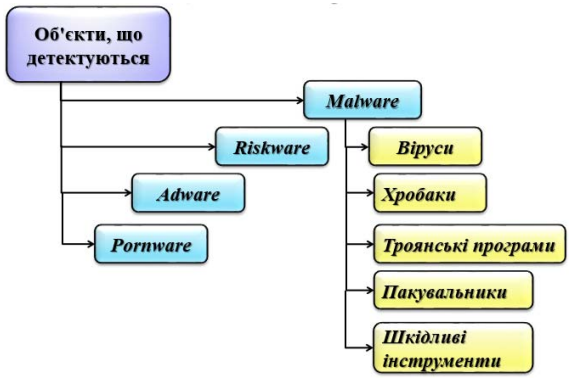
3.1.3 Класифікація шкідливого програмного забезпечення

Необхідність створення класифікації об'єктів, що детектуються, виникла одночасно з появою першої антивірусної програми. Шкідливі програми необхідно відрізняти одну від однієї по шкідливому навантаженню, методам поширення, небезпеці та іншим ознакам.

Перші спроби упорядкувати процес класифікації були зроблені на початку 1990-х років у рамках альянсу антивірусних фахівців **CARO** (*Computer AntiVirus Researcher's Organization*). Альянсом був створений документ «**CARO malware naming scheme**», який на якийсь період став стандартом для індустрії.

Але з часом стрімкий розвиток шкідливих програм, поява нових платформ і зростання числа антивірусних компаній привели до того, що ця схема фактично перестала використовуватися. Ще важливішою причиною відмови від неї стали істотні відмінності в технологіях детектування кожної антивірусної компанії і, як наслідок, неможливість уніфікації результатів перевірки різними антивірусними програмами.

Періодично робляться спроби виробити нову загальну класифікацію об'єктів, що детектуються антивірусними програмами, проте вони залишаються безуспішними. Останнім значним проектом подібного роду було створення організації **CME** (*Common Malware Enumeration*), яка присвоює однаковим об'єктам, що детектуються, єдиний унікальний ідентифікатор.



Комп'ютерні віруси

Вірус – програмний код, що мимоволі поширює свої копії по ресурсах локального комп'ютера з метою подальшого запуску свого коду при яких-небудь

діях користувача і подальшого впровадження в інші ресурси комп'ютера.

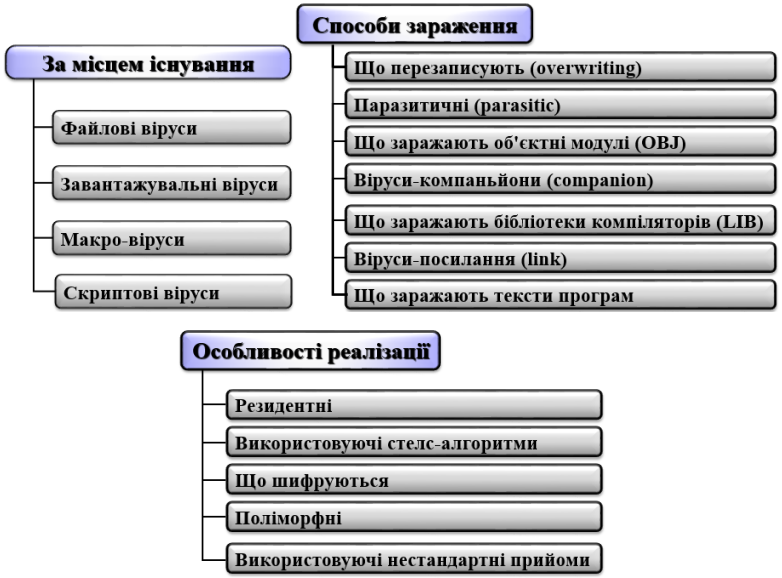
Виконує різні небажані дії: псування файлів, каталогів, спотворення результатів обчислень, знищення або спотворення інформації і т.д.

Відмітні особливості:

- несанкціоноване саморозмноження;
- прив'язка до інформаційних об'єктів.

Віруси розрізняються за наступними ознаками:

- місцю існування;
- способу зараження;
- особливостям алгоритмів реалізації.



Мережеві хробаки

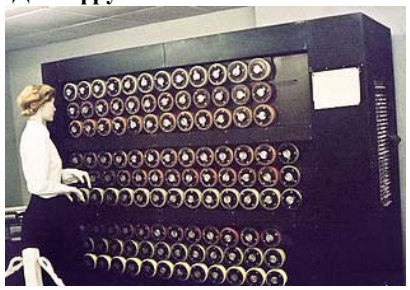
Мережеві хробаки– програми, що поширюють свої копії по локальних і/або глобальних мережах з метою:

- проникнення на видалені комп'ютери;
- запуску своєї копії на видаленому комп'ютері;
- подальшого поширення на інші комп'ютери мережі;
- транспортування іншого шкідливого ПЗ.

❒ **Email-Worm** – має здатність до несанкціонованого саморозмноженню по каналах електронної пошти. В процесі розмноження хробак посилає або свою копію у вигляді вкладення в електронний лист, або посилання на свій

Шифрувальна машина «Енігма»

Дешифрувальна машина «Bombe»



Дешифрувальна машина «Бомба» еквівалентна 36 робочим шифрувальним машинам «Енігма». Довжина 3 м, висота 2,1 м, ширина 0,61 м, вага 2.5 т. Серійно випускалася до вересня 1944 року. Всього в Блетчлі-парк було встановлено 210 машин типу «Bombe», що дозволило щодня розшифровувати до 3 тисяч повідомлень.

Загальна модель шифрування заміною може бути представлена у виді:

$$c_i \equiv m_i + w \pmod{n-1},$$

де c_i – буква зашифрованого тексту; m_i– буква відкритого тексту; w – ціле число в діапазоні від 0 до (n–1); n– число букв використовуваного алфавіту.

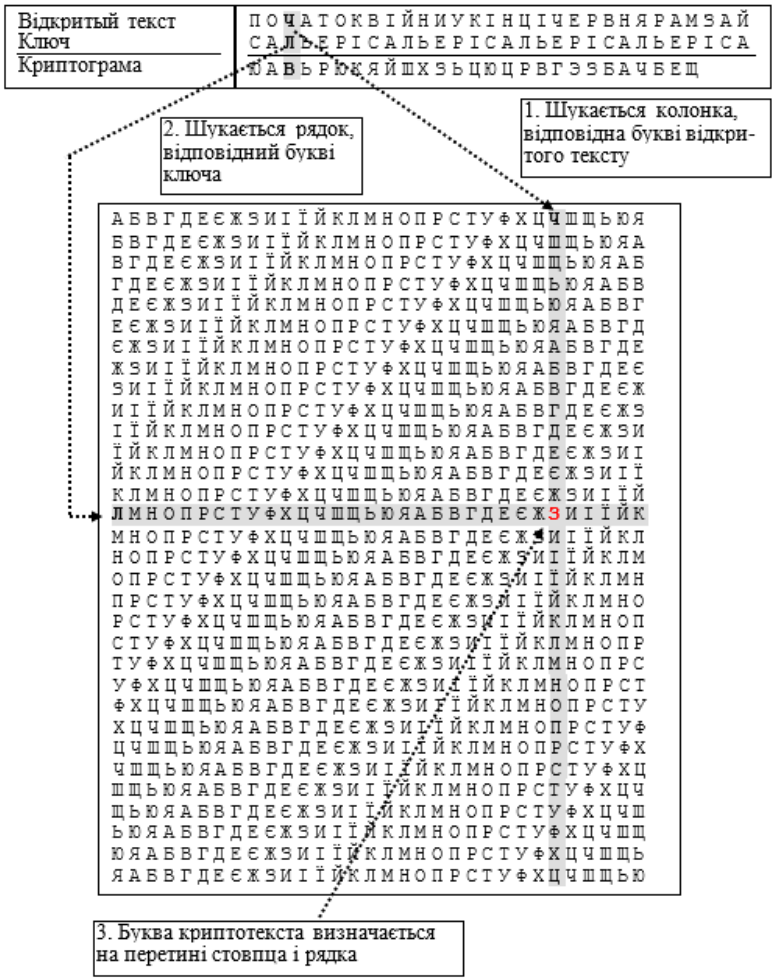
Якщо w фіксоване, то формула описує моноалфавітну заміну, якщо w вибирається з послідовності $w_1, w_2, \ldots, w_n$, то вийде поліалфавітна заміна з періодом n. Якщо в поліалфавітної заміні $n>m$ (де m – число букв шифрованого тексту) і будь-яка послідовність $w_1, w_2, \ldots, w_n$використовується тільки один раз, то такий шифр є теоретично нерозкриваним, якщо, звичайно, криптоаналітик не має доступу до вихідного тексту.

Шифри переставлення

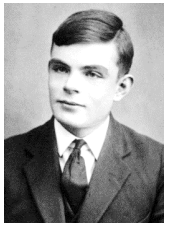
Переставлення – це метод шифрування, при якому зберігаються усі букви відкритого тексту, але вони розміщуються в криптограмі в іншому порядку, тобто переставляють елементи відкритих даних (біти, букви, символи) в деякому новому порядку.

Розрізняють шифри горизонтального, вертикального, подвійного переставлення, лабіринти та ін.

Історичним прикладом шифру переставлення є **скитала** (чи **сцитала** від грецького *σκυτάλη*, жезл) – шифр Древньої Спарти (III століття до Р.Х.) – пристрій, що складається з циліндра і вузької смужки пергаменту, яка обмотувалася навколо нього по спіралі, на якій писалося повідомлення. Античні греки і спартанці використовували цей шифр для зв'язку під час військових кампаній.



В усіченому вигляді шифр Віженера був реалізований в шифрувальній машинці «Енігма», широко використовуваною в німецьких військах під час Другої світової війни на тактичному і оперативному рівнях. Зусиллями британських криптоаналітиків шифр «Енігми» був успішно зламаний. Велика роль в цьому процесі належить видатному англійському математикові **Алану Т'юрингу (1912-1954)**, що запропонував використовувати для розтину шифру пристрій «Bombe».



файл, розташований на якому-небудь мережевому ресурсі (наприклад, URL на заражений файл, розташований на зламаному або хакерському веб-сайті).

❒ **IM-Worm** – має здатність до несанкціонованого саморозмноження по каналах систем миттєвого обміну повідомленнями (наприклад, **ICQ**, **MSN Messenger**, **AOL Instant Messenger**, **Yahoo Pager**, **Skype** та ін.). Розсилає на виявлені контакти (з контакт-листа) повідомлення, що містятьURL на файл з тілом хробака, розташований на якому-небудь мережевому ресурсі.

❒ **IRC-Worm** – має здатність до несанкціонованого саморозмноження через **Internet Relay Chats**. Способи поширення: 1) відсилання URL на копію хробака; 2) відсилання зараженого файлу якому-небудь користувачеві IRC-каналу. При цьому користувач, що атакується, повинен підтвердити прийом файлу, потім зберегти його на диск і відкрити (запустити на виконання).

❒ **Net-Worm** – має здатність до несанкціонованого саморозмноження в комп'ютерних мережах. Відмітна особливість: відсутність необхідності в користувачі як ланцюжку поширення (тобто безпосередньо для активації шкідливої програми).При поширенні хробак шукає в мережі комп'ютери, на яких використовується ПЗ, що містить критичні уразливості. Для зараження таких комп'ютерів хробак посилає спеціально сформований мережевий пакет (експлойт), внаслідок чого код хробака проникає на комп'ютер-жертву і активується.

❒ **P2P-Worm** – має здатність до несанкціонованого саморозмноження по каналах файлообмінних пірингових мереж (наприклад, **Kazaa**, **Grokster**, **eDonkey**, **FastTrack**, **Gnutella** та ін.).Для впровадження в P2P-мережу хробаку досить скопіювати себе в каталог обміну файлами, який зазвичай розташований на локальній машині. Усю іншу роботу по поширенню вірусу P2P мережа бере на себе – припошуку файлів в мережі вона повідомить видалених користувачів про цей файл і надає увесь необхідний сервіс для скачування файлу із зараженого комп'ютера.

Троянські програми

Троянські програми– програми, що здійснюють різні несанкціоновані користувачем дії:

- збір інформації та її передачу порушникові,
- її руйнування або зловмисну модифікацію,
- порушення працездатності комп'ютера,
- використання ресурсів комп'ютера в непристойних цілях.

❒ **Backdoor** – для прихованого видаленого управління ураженого ПК. По своїй функціональності нагадує легальні системи адміністрування. Дозволяють робити з комп'ютером усе, що в них заклав автор: приймати або посилати файли, запускати і знищувати їх, виводити повідомлення, стирати інформацію, перезавантажувати комп'ютер.

❒ **Rootkit** – для приховання в системі певних об'єктів, або активності. Прихованню піддаються ключі реєстру (наприклад, що відповідають за авто-

запуск шкідливих об'єктів), файли, процеси в пам'яті зараженого комп'ютера, шкідлива мережева активність.

❒ **Trojan-ArcBomb** – архіви, спеціально сформовані так, щоб викликати нештатну поведінку архіваторів при спробі розархівувати дані - зависання або істотне уповільнення роботи комп'ютера або заповнення диска великою кількістю "порожніх" даних.

❒ **Trojan-Dropper** – програма для прихованої інсталяції на комп'ютер-жертву шкідливих програм, що містяться в її тілі. В результаті використання хакери досягають цілей: 1) потайній інсталяції троянських програм і вірусів; 2) захист від детектування відомих шкідливих програм антивірусами.

❒ **Trojan-DDoS** – програма для проведення DoS -атаки з ураженого комп'ютера на комп'ютер-жертву за заздалегідь визначеною адресою. Суть атаки зводиться до посилки жертві численних запитів, що призводить до відмови в обслуговуванні, якщо ресурси видаленого комп'ютера, що атакується, недостатні для обробки усіх запитів, що поступають.

❒ **Trojan-downloader** – програма для завантаження і установки на комп'ютер-жертву нових версій шкідливих програм, установки троянських програм або рекламних систем.

❒ **Trojan-Proxy** – програма для здійснення зловмисником анонімного доступу до різних інтернет-ресурсів через комп'ютер-жертву. Використовується при розсилці спаму через заражені комп'ютери.

❒ **Trojan-Spy** – програма для ведення електронного шпигунства за користувачем (інформація, що вводиться з клавіатури, знімки екрану, список активних застосувань і так далі). Знайдена інформація передається зловмисникові по каналах електронної пошти, FTP, HTTP.

Підозрілі пакувальники

Шкідливі програми часто стискуються різними способами упаковки, поєднаними з шифруванням утримуваного файлу для того, щоб виключити зворотну розробку програми і ускладнити аналіз поведінки проактивними і евристичними методами. Антивірусом детектуються результати роботи підозрілих пакувальників – упаковані об'єкти.

Шифр зсуву (до якого, зокрема відноситься шифр Цезаря) є окремим випадком загального шифру заміни на сукупності лише n ключів, в якому нижній ряд є циклічним зсувом верхнього ряду. Ключ такого типу повністю визначається завдовжки зсуву s, $0 \le s < n$ (s і $s + n$ дають однаковий результат). Алгоритм шифрування полягає в наступному:

$$c_i \equiv m_i + s \pmod{n},$$

де m_i і c_i – відповіднобукви відкритого тексту і шифрограми.

Аналогічно здійснюється дешифрування:

$$m_i \equiv c_i + s \pmod{n}.$$

У зв'язку з низькою стійкістю шифрів простої заміни, обумовленої тим, що частотні властивості шифрограми співпадають з частотними властивостями відкритого тексту, на практиці частіше застосовуються складніші алгоритми. Зокрема, **поліалфавітні шифри** – шифри заміни, в яких позиція букви у відкритому тексті впливає на те, за яким правилом ця буква замінюватиметься, тобто для заміни деякого символу вихідного повідомлення в кожному випадку його появи послідовно використовуються різні символи з деякого набору. Зрозуміло, що цей набір не нескінченний, через якусь кількість символів його треба використовувати знову. У цьому слабкість чисто поліалфавітних шифрів.

Як приклад розглянемо **шифр Віженера**. Він є квадратною таблицею $n \times n$, в якій вихідний алфавіт і наступнішифралфавіти послідовно циклічно зсуваються вліво на одну позицію.

Шифр Віженера використовується для відкритого тексту, записаного в ряд без пропусків між словами і розділових знаків. Ключем є слово в тому ж алфавіті. Якщо ключ коротший за повідомлення, то його записують кілька разів підряд доки не утворюється ряд тієї ж довжини, що і відкритий текст. Ряд з розмноженим ключем розміщується під поряд з повідомленням, і букви, які виявилися одна над іншою, складаються. В результаті виходить ще один ряд тієї ж довжини, який і є криптотекстом.

За допомогою таблиці Віженера зашифруємо текст «ПОЧАТОК ВІЙНИ У КІНЦІ ЧЕРВНЯ. РАМЗАЙ» за допомогою ключа «САЛЬЄРІ».

На відміну від шифру простої заміни при використанні шифру Віжинера однаковим буквам у відкритому тексті можуть відповідати різні букви в шифртексте. Ця обставина ускладнює частотний криптоаналіз, оскільки статистичні характеристики відкритого тексту практично не проявляються в криптограмі.

Розшифрування шифрограми виробляється в наступній послідовності:

1) визначається рядок таблиці, відповідний букві ключа;
2) у цьому рядку шукається буква шифртекста;
3) піднімаючись стовпцем, відповідним цій букві, в першому рядку таблиці визначається буква відкритого тексту.

❒ *поліалфавітні* (наприклад, шифр Віженера, циліндр Джефферсона, диск Уетстоуна, Enigma).

У **моноалфавітних шифрах заміни** буква вихідного тексту замінюється на іншу, заздалегідь визначену букву. Наприклад, в шифрі Цезаря буква замінюється на букву, віддалену від неї в латинському алфавіті на деяке число позицій. Очевидно, що такий шифр зламується зовсім просто. Треба підрахувати, як часто зустрічаються букви в зашифрованому тексті, і зіставити результат з відомою для кожної мови частотою букв, що зустрічаються у мові.

Приклад: **шифр Цезаря** реалізує наступне перетворення відкритого тексту – кожна буква відкритого тексту замінюється третьою після неї буквою в алфавіті, який вважається написаним по кругу.

Вихідний алфавіт	**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**
Шифралфавіт	**D E F G H I J K L M N O P Q R S T U V W X Y Z A B C**

Зашифруємо повідомлення ***ULIYCEZAR***. Відкритій букві ***U*** (верхній рядок таблиці) відповідає шифрована буква ***X*** (нижній рядок таблиці), букві ***L*** – шифрбуква ***O*** і т.д. Як ключ використовується величина циклічного зсуву вихідного алфавіту на три позиції вліво. У результаті отримаємо шифртекст ***XOLBFHCDU***.

Другий приклад: частина зашифрованого повідомлення з оповідання Артура Конан Дойля «Чоловічки, що танцюють» про пригоди Шерлока Холмса. У шифрі використовуються малюнки чоловічків, кожна поза яких відповідає окремій букві звичайного алфавіту.

Нехай X і Y – два алфавіти (відкритого і шифрованого текстів відповідно), що складаються з однакового числа символів. Нехай також $g: X \to Y$— взаємно-однозначне відображення Xв Y. Тоді шифр заміни діє так: відкритий текст $x_1x_2\ldots x_m$ перетвориться в шифрований текст $g(x_1)g(x_2)\ldots g(x_m)$.

Якщо в алфавіт входить n символів, то загальна кількість ключів рівна $n!$. Так, при $n = 26$ маємо $26! > 10^{26}$ (це число дуже велике: так вік нашої планети складає 10^9 років). Усе це вказує на безперспективність лобової атаки на шифр заміни, але це не означає, що він є надійним. Виявляється, що успішний криптоаналіз можливий за допомогою **частотного методу**: частота символу в тексті дорівнює кількості його входжень в цей текст, ділена на загальну кількість символів в тексті. Для кожної мови справедливий наступний емпіричний факт:

> ***у досить довгих текстах кожна буква зустрічається приблизно з однаковою частотою, залежною від самої букви і незалежною від конкретного тексту***.

Існують прийоми боротьби з розпаковуванням: наприклад, пакувальник може розшифровувати код не повністю, а лише у міру виконання або розшифровувати і запускати шкідливий об'єкт цілком тільки в певний день тижня.

Основними ознаками, по яких диференціюють поведінку об'єктів цього підкласу є вид і кількість пакувальників, використаних при стискуванні файлу.

❒ **MultiPacked** – багаторазово упаковані різними програмами упаковки файлові об'єкти. Антивірусний продукт видає даний вердикт при виявленні виконуваних файлів, упакованих одночасно трьома і більше пакувальниками.

❒ **SuspiciousPacker** – файлові об'єкти, стислі пакувальниками, створеними спеціально для захисту шкідливого коду від детектування антивірусними продуктами.

❒ **RarePacker** – файлові об'єкти, стислі різними пакувальниками, що рідко зустрічаються, наприклад, що реалізовують яку-небудь концептуальну ідею.

Шкідливі утиліти

Шкідливі утиліти(*Malicious tools*) – програми, розроблені для автоматизації створення вірусів, хробаків або троянських програм, організації DoS - атак на видалені сервера, злому інших комп'ютерів.

❒ **Constructor** – програма для виготовлення нових комп'ютерних вірусів, хробаків і троянських програм. Відомі конструктори шкідливих програм для DOS, Windows і макро-платформ. Дозволяють генерувати вихідні тексти шкідливих програм, об'єктні модулі і безпосередньо заражені файли.

❒ **DoS** – програма для проведення DoS-атаки (*Denial of Service*– відмова в обслуговуванні) з відома користувача на комп'ютер-жертву.

❒ **Exploit** – програма, в якій містяться дані або виконуваний код, що дозволяють використовувати одну або декілька уразливостей в ПЗ на локальному або видаленому комп'ютері зі свідомо шкідливою метою. Використовується зловмисниками для проникнення на комп'ютер-жертву з метою подальшого впровадження шкідливого коду.

❒ **HackTool** – програма, використовувана зловмисниками при організації атак на локальний або видалений комп'ютер (наприклад, внесення нелегального користувача в список дозволених відвідувачів системи; очищення системних журналів з метою приховання слідів присутності в системі і т.д.).

❒ **VirTool** – програма, що дозволяє зловмисникові модифікувати інші шкідливі програми так, щоб вони не детектувалися антивірусним програмним забезпеченням.

❒ **Spoofer** – програми, що дозволяють відправляти повідомлення і мережеві запити з підробною адресою відправника.Програми цього типу можуть бути використані з різними цілями (наприклад, утруднити виявлення відправника або видати повідомлення за повідомлення, відправлене оригіналом).

❒ **Hoax** – програми, які не заподіюють комп'ютеру якої-небудь прямої шкоди, проте виводять повідомлення про те, що така шкода вже причинна, або буде причинний за яких-небудь умов, або попереджають користувача про неіснуючу небезпеку.До «злих жартів» відносяться, наприклад, програми, які «лякають» користувача повідомленнями про форматування диска (хоча ніякого форматування насправді не відбувається), виводять дивні вірусоподібні повідомлення і т.д.

❒ **Flooder** – програми, функцією яких є «забивання сміттям» (даремними повідомленнями) мережевих каналів, відмінних від поштових, інтернет-пейджерів і SMS (наприклад, IRC). Програми, що «забивають» канали поштових служб, інтернет-пейджерів і SMS-канали, відносяться відповідно до **Email-Flooder**, **IM-Flooder** і **SMS-Flooder**.Ці програми можуть використовуватися спамерами.

Потенційно небажані програми

Потенційно небажані програми(*PUPs, Potentially Unwanted Programs*) – програми, які розробляються і поширюються легально і можуть використовуватися в повсякденній роботі, наприклад, системних адміністраторів. Проте вони мають функції, які можуть завдати шкоди користувачеві при виконанні ряду умов.

Adware – рекламне ПЗ, призначене для показу рекламних повідомлень (у вигляді графічних банерів); перенаправлення пошукових запитів на рекламні веб-сторінки, а також для збору даних маркетингового характеру про активність користувача.Потрапляють на комп'ютер двома способами:

1) шляхом вбудовування реклами у безкоштовне і умовно-безкоштовне ПЗ (*freeware*, *shareware*);

2) шляхом несанкціонованої постановки реклами при відвідуванні заражених веб-сторінок.

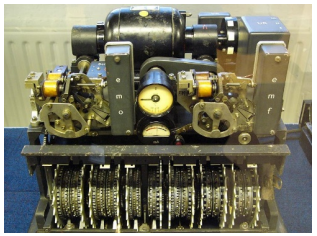
Riskware – легальні програми, які в руках зловмисника здатні завдати шкоди користувачеві (викликати знищення, блокування, модифікацію або копіювання інформації, порушити роботу комп'ютерів або комп'ютерних мереж).

У списку програм цієї категорії можна виявити комерційні утиліти віддаленого адміністрування, програми-клієнти IRC, програми дозвону, програми для завантаження («скачування») файлів, монітори активності комп'ю-

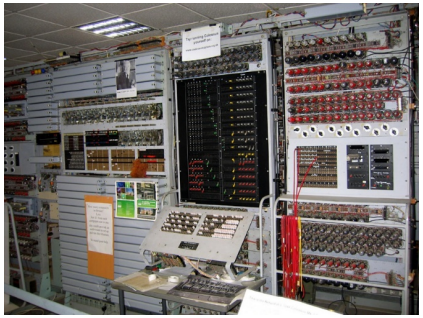
завжди можливий. Тому частіше використовують гаму, що отримується за допомогою **генератора псевдовипадкових чисел (ГПВЧ)**. В цьому випадку ключем виступає *число, що породжує запуск ГПВЧ* (початкове значення, вектор ініціалізації). Кожен ГПВЧ має *період*, після якого генерована послідовність повторюється. Очевидно, що період псевдовипадкової гами повинен перевищувати довжину шифрованої інформації.

Прикладом криптосистеми, що застосовувала потокові шифри на основі гамування була німецька шифрувальна машина **«Лоренц»**, що використалася під час Другої світової війни на стратегічному рівні для передачі інформації по телетайпу. Британські аналітики називали шифри «Лоренца» і її саму **«Танни»** (англ. *tunny* - **тунець**).

Шифрувальна машина «Лоренц»



ЕОМ «Colossus»



Для її злому криптоаналітикам з Блетчлі-парку потрібно було створити в 1943 році програмовану електронну обчислювальну машину (ЕОМ) «Colossus» (Макс Н'юмен Томмі Флауерс за участю Алана Т'юринга). Це була перша у світі ЕОМ. Вона включала 1600 електронних ламп і дозволила скоротити час, потрібний на злом повідомлень, з шести тижнів до декількох годин.

При **блоковому шифруванні** початковий відкритий текст розбивається на блоки фіксованої довжини і шифрується поблоково. Розрізняють наступні основні види блокових шифрів:

- ❒ *шифри заміни* (*підстановки, S-блоки*);
- ❒ *шифри переставлення* (*P-блоки*);
- ❒ *шифри, засновані на аналітичних перетвореннях.*

***Шифри підстановки* (*заміни*)**

Підстановка (заміна) – це метод шифрування, при якому кожен знак вихідного тексту взаємно-однозначно замінюється шифрпозначенням – одним або декількома знаками деякого набору символів (алфавіту).

Розрізняють шифри простої, складної, парної заміни, буквено-складове шифрування і шифри колонної заміни. Шифри заміни діляться на дві групи:

- ❒ *моноалфавітні* (наприклад, шифр Цезаря);

У **потокових шифрах**, тобто при шифруванні повідомлення, представленого у вигляді безперервної послідовності бітів, кожен біт відкритого тексту шифрується незалежно від інших за допомогою гамування.

Гамування – це накладення на відкриті дані гами шифру (випадковій або псевдовипадковій послідовності одиниць і нулів) за певним правилом. Зазвичай використовується операція «виключне АБО» (зване також складанням за модулем 2 і що реалізовується командою **XOR**). Для розшифрування та ж гамма накладається на зашифровані дані.

Процедуру накладення гами на відкритий текст можна здійснити двома способами:

1) символи вихідного тексту і гами замінюються цифровими еквівалентами, які потім складаються за модулем n:

$$c_{ш} \equiv m_{в} + g_{г} (\mathrm{mod}\ n),$$

де n – число символів в алфавіті; $c_{ш}$, $m_{в}$, $g_{г}$ – відповідно символи зашифрованого, вихідного тексту і гами;

2) символи вихідного тексту і гами представляються у вигляді двійкового коду, потім відповідні розряди складаються за модулем 2.

Замість складання за модулем 2 при гамуванні можна використовувати інші логічні операції, наприклад, перетворення за правилом *логічної еквівалентності* (а) або *логічної нееквівалентності* (б). Така заміна рівносильна вступу ще одного ключа, яким є вибір правила формування символів зашифрованого повідомлення з символів вихідного тексту і гами.

Гама \ Текст	0	1
0	0	1
1	1	0

а)

Гама \ Текст	0	1
0	1	0
1	0	1

б)

Вихідний текст	Б	У	Д	Ь
	010010	100000	110010	100000
Знаки гами	7	1	8	2
	000111	000001	001000	000010
Шифртекст	010101	100001	111011	100010

Стійкість шифрування методом гамування визначається головним чином властивостями гами: *тривалістю періоду* і *рівномірністю статистичних характеристик*. Остання властивість забезпечує відсутність закономірностей в появі різних символів в межах періоду.

При одноразовому використанні випадкової гами однакового розміру із зашифровуваними даними злом шифру неможливий (так звані криптосистеми з одноразовим або нескінченним ключем). В даному випадку «нескінченний» означає, що гамма не повторюється.

Зрозуміло, що обмін ключами розміром з шифровану інформацію не

терних систем, утиліти для роботи з паролями, а також численні інтернет-сервери служб **FTP**, **Web**, **Proxy** і **Telnet**.

Усі ці програми не є шкідливими самі по собі, проте мають функціонал, яким можуть скористатися порушники для спричинення шкоди користувачам.

До цієї категорії об'єктів, що детектуються, відносяться:

❒ **Client-IRC** – програми, використовувані для спілкування в Internet Relay Chats. Детектування додане унаслідок частого використання зловмисниками розширеного функціонала цих програм – із завидною періодичністю виявляються шкідливі програми, що встановлюють Client-IRC на призначені для користувача комп'ютери із зловмисними цілями.

❒ **Client-P2P** – програми, використовувані для роботи в peer-to-peer мережах. Детектування додане по проханнях користувачів, оскільки ряд програм подібного роду стали причиною просочування конфіденційної інформації.

❒ **Client-SMTP** – програми, що використовувані для відправки електронної пошти і мають прихований режим роботи. Ці програми можуть включатися зловмисниками до складу пакету шкідливих програм для розсилки спаму або іншого шкідливого контенту з комп'ютерів користувачів.

❒ **Dialer** – програми, що дозволяють встановлювати в прихованому режимі телефонні з'єднання через модем.

❒ **Downloader** – програми, що дозволяють здійснювати в прихованому режимі завантаження різного контенту з мережевих ресурсів. Подібні програми можуть використовуватися зловмисниками для завантаження шкідливого контенту на комп'ютер-жертву.

❒ **FraudTool** – програми, які видають себе за інші програми, хоча такими не є. Часто пропонують користувачеві перерахувати фінансові кошти на певні рахунки для оплати «послуг».Як приклад таких програм можна привести псевдоантивіруси, які виводять повідомлення про «виявлення» шкідливих програм, але насправді нічого не знаходять і не лікують.

❒ **Monitor** – програми, що містять функції спостереження за активністю на комп'ютері користувача (активні процеси, мережева активність і т.д.). У цих же цілях можуть бути використані зловмисниками.

❒ **NetTool** – програми, що мають різну мережеву функціональність (наприклад, видалене перезавантаження комп'ютера, сканування відкритих мережевих портів, видалений запуск довільних застосувань і т.д.), що дозволяє використання їх кіберзлочинцями із зловмисними цілями.

❒ **PSWTool** – програми, що дозволяють переглядати або відновлювати забуті (часто – приховані) паролі. З таким же успіхом в подібних цілях цей тип програм може бути використаний зловмисниками.

❒ **RemoteAdmin** – програми, використовувані для видаленого управління комп'ютером. Будучи встановленими порушником дають йому можливість повного контролю над комп'ютером-жертвою.

❒ **RiskTool** – програми, що мають різну функціональність (наприклад, приховання файлів в системі, приховання вікон запущених застосувань, знищення активних процесів і т.д.), що дозволяє використання їх кіберзлочинцями із зловмисними цілями.

❒ **Server-FTP** – програми, що містять функціональність FTP-сервера. З цієї причини включаються зловмисниками в пакети шкідливих програм, наприклад, для організації видаленого доступу до комп'ютера-жертви, де встановлена ця програма.

❒ **Server-Proxy** – програми, що містять функціональність Proxy-сервера. З цієї причини включаються зловмисниками в пакети шкідливих програм, наприклад, для розсилки спаму або іншого шкідливого контенту від імені комп'ютера-жертви.

❒ **Server-Telnet** – програми, що містять функціональність Telnet-сервера. З цієї причини включаються зловмисниками в пакети шкідливих програм, наприклад, для організації видаленого доступу до комп'ютера-жертви, де встановлена ця програма.

❒ **Server-Web** – програми, що містять функціональність веб-сервера. З цієї причини включаються зловмисниками в пакети шкідливих програм, наприклад, для організації видаленого доступу до комп'ютера-жертви, де встановлена ця програма.

❒ **WebToolbar** – програми, які з дозволу користувача розширюють можливості призначеного для користувача ПЗ шляхом установки панелей інструментів, що дозволяють використовувати одну або декілька пошукових систем при роботі в Інтернеті. Детектування додане унаслідок частого поширення подібних панелей за допомогою різних шкідливих програм у вигляді вкладених в них файлів.

Pornware – програми, які так чи інакше пов'язані з показом користувачеві інформації порнографічного характеру. Ці програми можуть бути встановлені на комп'ютер користувача зловмисниками – через використання уразливостей ОС і Інтернет-браузера або за допомогою шкідливих троянських програм класів Trojan-Downloader або Trojan-Dropper. Робиться це звичайно з метою «насильницької» реклами платних порнографічних сайтів і сервісів, на які користувач сам по собі ніколи не звернув би уваги.

3.2 Методи і технології захисту від шкідливих програм

3.2.1 Методи і способи захисту від шкідливого програмного забезпечення

Захист комп'ютерних систем від дії різних шкідливих програм повинен будуватися на основі *комплексного* використання юридичних, організаційних (освітніх) і технічних методів.

1. Юридичні (поліцейські) методи. В усіх комп'ютеризованих країнах

Процедура шифрування наступна:

M **= 00010 01010 01100 10000 00000** **ВІЙНА** (Відкритий текст)
⊕⊕
K **= 00001 00000 10000 00000 10000** **БАНАН** (Ключ)

C **= 00011 01010 11100 10000 10000** **ГІЩНН** (Шифртекст)

Процедура розшифрування наступна:
C **= 00011 01010 11100 10000 10000** **ГІЩНН** (Шифртекст)
⊕⊕
K **= 00001 00000 10000 00000 10000** **БАНАН** (Ключ)

M **= 00010 01010 01100 10000 00000** **ВІЙНА** (Відкритий текст)

Чому ж шифр Вернама не розкривний? Річ у тому, що, якщо відома криптограма, та її довжина рівна *n* бітів, то, перебираючи усі можливі ключі (тобто усі можливі двійкові послідовності довжини *n* бітів (а таких послідовностей 2^n)) і складаючи їх посимвольно за модулем 2 з криптограмою, можна отримати усі можливі двійкові тексти довжини *n* бітів. Який з них був справжнім повідомленням, встановити неможливо.

Проте обмеженість сфери застосування шифру очевидна, оскільки він вимагає ключа тієї ж довжини, як і само повідомлення. З цією обставиною пов'язано дві проблеми:

1) генерування довгої послідовності випадкових біт;
2) необхідність надійного каналу для регулярного обміну довгими ключами.

Для абсолютної стійкості істотною є кожна з наступних вимог до стрічки одноразового використання:

❒ повна випадковість (рівноймовірність) ключа (це, зокрема, означає, що ключ не можна виробляти за допомогою якого-небудь детермінованого пристрою, або програми);

❒ рівність довжини ключа і довжини відкритого тексту;

❒ одноразовість використання ключа.

У разі порушення хоч би одного з цих умов шифр перестає бути абсолютно стійким, і з'являються принципові можливості для його розкриття.

Симетричні криптосистеми можуть бути побудовані одним з наступних способів:

❒ *потокова схема* – для шифрперетворення використовується гама, не залежна від шифрованої інформації;

❒ *блокова схема* – гама для шифрперетворення виробляється шляхом маніпулювання з інформацією, яку потрібно зашифрувати;

❒ *комбінована схема* – для шифрперетворення використовуються як потокові, так і блокові методи, а також їх різновиди.

Аналіз існуючих шифрів дозволив К.Шеннону сформулювати висновку, що навіть в складних шифрах як типові компоненти можна виділити шифри *заміни*, *переставлення* або їх поєднання. Але найважливішим для розвитку криптографії був результат про існування і єдність *абсолютно стійкого шифру*: так званої ***стрічки одноразового використання***, в якій відкритий текст «об'єднується» з повністю випадковим ключем такої ж довжини.

Прикладом реалізації абсолютно стійкого шифру є **шифр одноразового блокнота**, запропонованого в 1917 р. американським інженером з телекомунікацій **Гілбертом Вернамом(1890-1960)** з компанії AT&T. В цьому шифрі здійснюється побітове складання *n*-бітового відкритого тексту і *n*-бітового випадкового ключа:

$$c_i = m_i \oplus k_i, \quad i = 1, 2, \cdots, n,$$

$m_1 \ldots m_n$– двійкові розряди відкритого тексту;

$k_1 \ldots k_n$– двійкові розряди ключа;

$c_1 \ldots c_n$– шифртекст в двійковому коді.

$\oplus$ – операція складання за модулем 2: $0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$.

Розшифрування в шифрі одноразового блокнота співпадає з шифруванням – щоб отримати вихідне повідомлення M, слід скласти з криптотекстом C той самий ключ K:

$$C \oplus K = (M \oplus K) \oplus K = M \oplus (K \oplus K) = M \oplus 0 = M.$$

Шифр одноразового блокнота є *абсолютно надійним* або, як ще говорять, *надійним в теоретико-інформаційному сенсі*. Якщо порушник не знає ключа K, то з перехопленої криптограми C він зовсім нічого не може дізнатися про повідомлення M.

Назва шифру походить від того, що агент, який здійснював шифрування вручну, отримував свої копії ключів, записаними в блокноті. Як тільки ключ використовувався, сторінка з ним знищувалася.

Як приклад, спробуємо зашифрувати слово ВІЙНА, використовуючи як ключ слово БАНАН. Для шифрування букв використовується кодова таблиця, в якій кожній букві зіставляється п'ятизначний двійковий код.

Буква	Код	Буква	Код	Буква	Код	Буква	Код
А	00000	З	01000	Н	10000	Х	11000
Б	00001	И	01001	О	10001	Ц	11001
В	00010	І	01010	П	10010	Ч	11010
Г	00011	Ї	01011	Р	10011	Ш	11011
Д	00100	Й	01100	С	10100	Щ	11100
Е	00101	К	01101	Т	10101	Ь	11101
Є	00110	Л	01110	У	10110	Ю	11110
Ж	00111	М	01111	Ф	10111	Я	11111

ухвалені закони, що забороняють створення і поширення вірусів та інших типів шкідливих програм.

Так до Кримінального Кодексу України включені статті 361^1 і 363^1, що передбачають кримінальне переслідування за злочини у сфері створення і поширення шкідливих програм.

Ст. 361^1. Створення з метою використання, розповсюдження або збуту шкідливих програмних чи технічних засобів, а також їх розповсюдження або збут призначених для несанкціонованого втручання в роботу ЕОМ (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку,

- караються штрафом від 500 до 1000 неоподатковуваних мінімумів доходів громадян або виправними роботами на строк до двох років, або позбавленням волі на той самий строк, з конфіскацією програмних чи технічних засобів;
- ті самі дії, вчинені повторно або за попередньою змовою групою осіб, або якщо вони заподіяли значну шкоду, караються позбавленням волі на строк до п'яти років з конфіскацією програмних чи технічних засобів.

Ст. 363^1. Перешкоджання роботі електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку шляхом масового розповсюдження повідомлень електрозв'язку

- карається штрафом від 500 до 1000 неоподатковуваних мінімумів доходів громадян або обмеженням волі на строк до трьох років;
- ті самі дії, вчинені повторно або за попередньою змовою групою осіб, якщо вони заподіяли значну шкоду, караються обмеженням волі на строк до п'яти років або позбавленням волі на той самий строк, з позбавленням права обіймати певні посади або займатися певною діяльністю на строк до трьох років та з конфіскацією програмних або технічних засобів.

На жаль, критерії, за якими програмні продукти можуть бути віднесені до категорії «шкідливих програмних засобів», а також «повідомлень електрозв'язку, які масово розповсюджуються» досі ніде чітко не обумовлені. Відповідно, для того, щоб твердження про шкідливість програмного засобу мало юридичну силу, необхідне проведення *програмно-технічної експертизи* з дотриманням всіх встановлених чинним законодавством формальностей.

2. Організаційні (освітні) методипов'язані з організацією служб антивірусного захисту, планування і контролю їх діяльності, навчання персоналу і користувачів правилам поведінки при роботі з комп'ютерною технікою:

- обов'язковість використання антивірусного захисту;
- ретельність перевірки усієї інформації, що поступає на комп'ютер;
- регулярність оновлення операційної системи, прикладних програм, з якими здійснюється робота, і антивірусних засобів;

- звернення серйозної уваги на інформацію від антивірусних компаній і від експертів з комп'ютерній безпеці.

3.Технічні методи пов'язані з вибором і використанням антивірусних програмних, апаратних і апаратно-програмних засобів.

Способи захисту від шкідливих програм

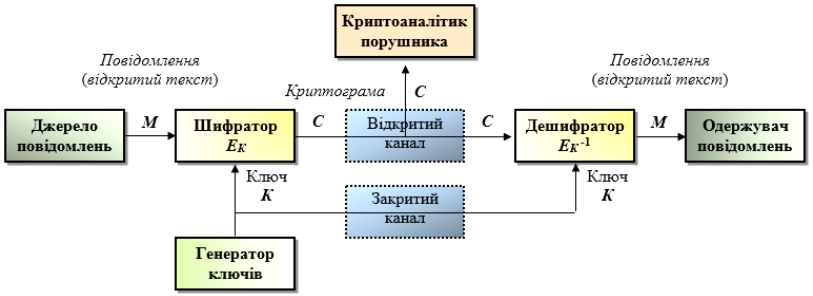
Абсолютного захисту від шкідливих програм не існує: від «експлойтів нульового дня», тобто шкідливих програм, проти яких ще не створені антивірусні засоби, на кшталт **Sasser** або **Conficker** не застрахований ніхто. Але за допомогою деяких заходів можна істотно понизити ризик зараження шкідливими програмами.

Основні і найбільш ефективні заходи для підвищення безпеки:

- ❒ використовувати операційні системи, що не дають змінювати важливі файли без відома користувача;
- ❒ своєчасно встановлювати оновлення;
  - якщо існує режим автоматичного оновлення, включити його;
  - для проприетарного ПЗ використовувати ліцензійні копії. Оновлення для двійкових файлів іноді конфліктують із зломщиками;
- ❒ окрім антивірусних продуктів, що використовують сигнатурні методи пошуку шкідливих програм, використовувати ПЗ, що забезпечує проактивний захист від загроз (необхідність використання проактивного захисту обумовлюється тим, що сигнатурний антивірус не помічає нові загрози, ще не внесені до антивірусних баз). Проте, його використання вимагає від користувача великого досвіду і знань;
- ❒ постійно працювати на персональному комп'ютері виключно під правами користувача, а не адміністратора, що не дозволить більшості шкідливих програм інсталюватися на персональному комп'ютері і змінити системні налаштування. Але це не захистить персональні дані від шкідливих (**Trojan-Clicker**, **Trojan-DDoS**, **Trojan-Downloader**, **ransomware** [що шифрує файли], шпигунського ПЗ) і потенційно-небажаних програм (**Adware**, **Hoax**), що мають доступ до файлів користувача, до яких обмежений обліковий запис має дозвіл на запис і читання (наприклад, домашній каталог – підкаталоги **/home** в **GNU/Linux**, **Documents and settings** в Windows, папка **Users** в Windows 7, 8, 8.1, 10), до будь-яких папок, в які дозволений запис і читання файлів, або інтерфейсу користувача (як роблять призначені для користувача програми для створення знімків екрану або зміни розкладки клавіатури);
- ❒ обмежити фізичний доступ до комп'ютера сторонніх осіб;
- ❒ використовувати зовнішні носії інформації тільки від перевірених джерел на робочому комп'ютері;
- ❒ не відкривати комп'ютерні файли, отримані від ненадійних джерел, на робочому комп'ютері;

вивчати криптографію, застосовуючи науковий підхід. Важливою заслугою є дослідження абсолютно стійких систем і доказ їх існування, а також існування криптостійких шифрів, і потрібні для цього умови. Шеннон сформулював основні вимоги, що пред'являються до надійних шифрів, ввів поняття розсіювання і перемішування, а також вказав методи створення криптостійких систем шифрування на основі простих операцій.

Модель Шеннона криптосистеми з секретним ключем (симетричної криптосистеми)



На передавальній стороні є *джерело повідомлень* і *генератор ключів*. Останній відбирає конкретний ключ K серед усіх можливих ключів цієї системи, який, з одного боку, подається на шифратор, а з іншого, – передається деяким способом одержувачу. При цьому передбачається, що його не можна перехопити (наприклад, ключ передається за допомогою посильного). Секретні ключі для джерела і одержувача повідомлень можуть також генеруватися і роздаватися довіреними організаціями (наприклад, центрами генерації і управління ключами). Але при цьому їх доставка абонентам також повинна здійснюватися закритим каналом.

Джерело повідомлень формує відкрите повідомлення M, яке зашифровується по заданому алгоритму f за допомогою ключа K в шифраторі, і готова криптограма C передається на приймальну сторону відкритим каналом зв'язку (це може бути локальна або глобальна мережа, дротяна, оптична, радіорелейна або інша лінія зв'язку), в якому вона може бути перехоплена криптоаналітиком порушника.

На приймальній стороні за допомогою дешифратора за допомогою ключа K за алгоритмом зворотного перетворення f^{-1} криптограма C перетвориться в вихідне повідомлення M (розшифровує його).

Формально вказані процедури описуються так:

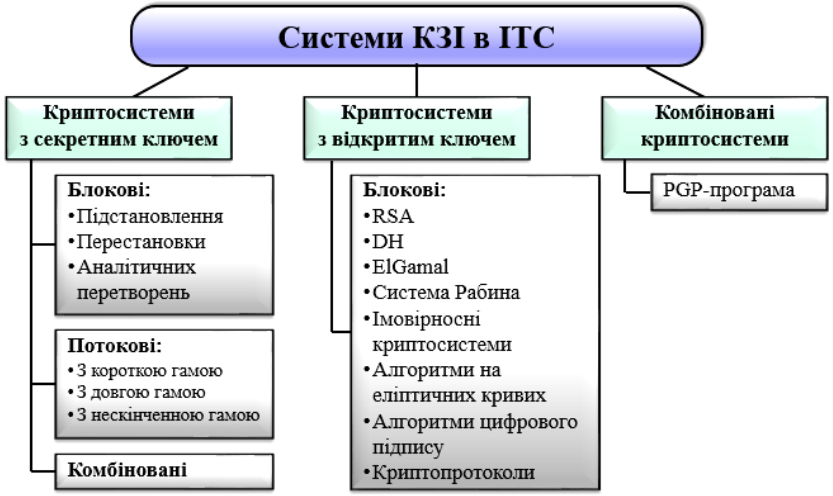
1) шифрування $C = f(M, K) = E_K(M)$;
2) розшифрування $M = f^{-1}(C, K) = D_K(C) = E_{K^{-1}}(M)$.

Криптосистеми, що використовують в процесі шифрування і розшифрування один і той же ключ, а також взаємнооборотні алгоритми, називаються **симетричними криптосистемами**.

рування – *секретний ключ*. Ці ключі різні і не можуть бути отримані один з іншого. Схема обміну інформацією такій криптосистемі наступна:

- одержувач обчислює відкритий і секретний ключі, секретний ключ зберігає в таємниці, відкритий же робить доступним (повідомляє відправника, групу користувачів мережі, публікує);
- відправник, використовуючи відкритий ключ одержувача, зашифровує повідомлення, яке пересилається одержувачеві;
- одержувач отримує повідомлення і розшифровує його, використовуючи свій секретний ключ.

❒ **комбіновані криптосистеми** використовують спільно шифрування з секретними і відкритими ключами.



Сучасна криптографія включає:

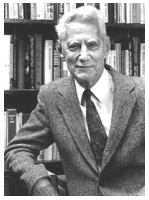
❒ симетричні криптосистеми (чи системи з секретним ключем);

❒ криптографічні системи з відкритим ключем (чи асиметричні криптосистеми);

❒ криптографічні протоколи;

❒ управління ключами.

4.2 Принципи побудови симетричних і асиметричних криптосистем



4.2.1 Криптосистеми з секретним ключем

Великий вплив на розвиток криптографії зробила стаття видатного американського математика **Клода Шеннона (1916-2001) «Теорія зв'язку в секретних системах»** (1945) з грифом «таємно», розсекречена і опублікована в 1949 р. Саме К. Шеннон уперше почав

❒ використовувати міжмережевий екран (апаратний або програмний), контролюючий вихід в мережу Інтернет з персонального комп'ютера на основі політик, які встановлює сам користувач;

❒ використовувати другий комп'ютер (не для роботи) для запуску програм з малонадійних джерел, на якому немає цінної інформації, що представляє інтерес для третіх осіб;

❒ робити резервне копіювання важливої інформації на зовнішні носії і відключати їх від комп'ютера (шкідливе ПЗ може шифрувати або ще якнебудь псувати знайдені ним файли).

3.2.2 Основи роботи антивірусних програм

Ефективним засобом захисту від шкідливого ПЗ є спеціальні програми, здатні розпізнавати і знешкоджувати їх у файлах, листах та інших об'єктах. Такі програми називаються **антивірусами**.

Антивірусна програма (антивірус) – програма для виявлення комп'ютерних вірусів і лікування інфікованих файлів, а також для профілактики – запобігання зараженню файлів або операційної системи шкідливим кодом. Складається з підпрограм (модулів), які намагаються виявити, запобігти розмноженню і видалити комп'ютерні віруси та інше шкідливе програмне забезпечення.

У сучасних антивірусних продуктах використовується два основні підходи до виявлення шкідливих програм: *сигнатурний* і *проактивний/евристичний*.

Сигнатурні методи– точніметоди виявлення вірусів, засновані на порівнянні файлу з відомими зразками вірусів.

Проактивні / евристичні методи– цеприблизні методи виявлення, які дозволяють з певною ймовірністю припустити, що файл заражений.

Сигнатурний аналіз

Сигнатурний аналіз полягає у виявленні характерних ідентифікуючих рис кожноїшкідливої програми і пошуку вірусів шляхом порівняння файлів з виявленими рисами.

Сигнатурою вірусу вважається сукупність рис, що дозволяють однозначно ідентифікувати наявність вірусу у файлі (включаючи випадки, коли файл цілком є вірусом). Всі разом сигнатури відомих вірусів складають **антивірусну базу**.

Ця технологія припускає безперервне відстежування нових екземплярів шкідливих програм, їх опис і включення в базу сигнатур. Задачу виділення сигнатур, як правило, вирішують люди – експертив області комп'ютерної вірусології, здатні виділити код вірусу з коду програми і сформулювати його характерні риси у формі, найбільш зручній для пошуку. У найбільш простих випадках можуть застосовуватися спеціальні автоматизовані засоби виділення сигнатур, наприклад, для нескладних за структурою троянов або хробаків, які не заражають інші програми, а цілком є шкідливими програмами.

Часто для виявлення сімейства схожих вірусів використовується одна сигнатура, і тому кількість сигнатур не завжди дорівнює кількості вірусів, що виявляються.

Важлива додаткова властивість сигнатур –*точнеі гарантоване визначення типу вірусу*. Ця властивість дозволяє занести в базу не лише самі сигнатури, але і способи лікування вірусу.

Головні критерії ефективності сигнатурного методу:

- швидкість реакції на нові загрози;
- частота оновлень;
- максимальне число виявлених загроз.

Головний *недолік* сигнатурного методу – затримкапри реакції на нові загрози. Для отримання сигнатури необхідно мати зразок вірусу. Створити його сигнатуру неможливо, поки вірус не потрапив на аналіз до експертів. Тому сигнатури завжди з'являються тільки через деякий час після появи нового вірусу З моменту появи вірусу в мережі Інтернет до випуску перших сигнатур зазвичай проходить декілька годин, і весь цей час вірус здатний заражати комп'ютери майже безперешкодно. Саме тому традиційний сигнатурний метод непридатний для оперативного захисту від вірусів, що знову з'являються.

Цей недолік традиційного сигнатурного аналізу дозволяє здолати «хмарний» антивірусний захист.

Особливості «хмарної» антивірусної технології

При використанні «хмарного» антивірусного захисту процес обміну інформацією між ПК і сервером виробника антивірусної програми відбувається постійно. Усі ПК підключені до видаленого сервера виробника і утворюють так звану «антивірусну хмару», що є інфраструктурою, яка використовується для обробки сервером інформації, яка поступає від користувачів ПК, про підозрілі шкідливі програми з метою своєчасно розпізнати нові, раніше невідомі загрози.

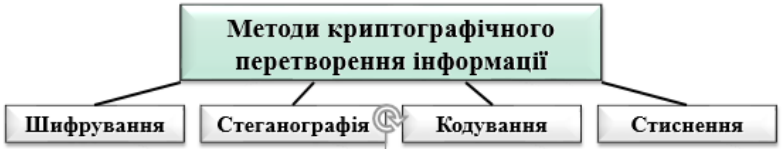
Хмарний антивірус не вимагає від користувача ніяких зайвих дій – користувач ПК просто відправляє запит з приводу підозрілої програми або поси-

Засоби криптографічного захисту інформації – програмні, апаратно-програмні, апаратні та інші засоби, призначені для криптографічного захисту інформації. До них відносяться:

- засоби, що реалізовують криптоалгоритми перетворення інформації;
- засоби, системи і комплекси захисту від нав'язування помилкової інформації, включаючи засоби імітозахисту і електронного підпису;
- засоби, системи і комплекси, призначені для виготовлення і розподілу ключових документів;
- засоби, системи і комплекси, призначені для захисту від НСД.

Криптографічна система (криптосистема) – сукупність засобів КЗІ, необхідних ключовій, нормативній, експлуатаційній та іншій документації, використання яких забезпечує відповідний рівень захищеності оброблюваної інформації, що зберігається або передається.

Система криптографічного захисту інформації – сукупність органів, підрозділів, груп, діяльність яких спрямована на забезпечення криптографічного захисту інформації, і підприємств, установ і організацій, які розробляють, випускають, експлуатують або поширюють криптосистеми і засоби КЗІ.



- **Шифрування** – процес проведення оборотних математичних, логічних, комбінаторних та інших перетворень вихідної інформації, в результаті яких зашифрована інформація є хаотичним набором букв, цифр, інших символів і двійкових кодів.
- **Стеганографія** – процес приховання не лише сенсу інформації, що зберігається або переробляється, але і сам факт зберігання або передачі закритої інформації.
- **Кодування** – процес заміни смислових конструкцій вихідній інформації (слів, речень) кодами (комбінаціями букв і цифр).
- **Стиснення** – процес скорочення об'єму інформації шляхом усунення надмірності.

4.1.2 Класифікація криптосистем

За **принципами побудови** (використання ключів) криптосистеми підрозділяються на:

- **криптосистеми з секретним ключем** (використовують симетричні алгоритми шифрування) засновані на тому, що відправник і одержувач інформації використовують один і той же ключ, який повинен зберігатися в таємниці і передаватися способом, що виключає його перехоплення;
- **криптосистеми з відкритим ключем** (асиметричні криптосистеми) використовують для шифрування інформації *відкритий ключ*, а для розшиф-

ми, то система з меншим числом секретів буде надійніша. Отже, принцип спрямований на те, щоб зробити безпеку алгоритмів і протоколів незалежною від їх секретності, тобто відкритість не повинна впливати на безпеку.

Передбачається, що механізм шифрування відомий порушникові, так само як і сама криптограма. **Загрози**, які можна чекати з боку порушника:

- ❒ розголошування інформації, якщо вона буде розкрита;
- ❒ підміна інформації;
- ❒ імітація повідомлень;
- ❒ модифікація повідомлень;
- ❒ знищення повідомлень та ін.

У будь-якому випадку порушник повинен зробити **атаку на шифр** з метою його розкриття (злому). Очевидним способом такої атаки є *підбір потрібного ключа*, за допомогою якого здійснюється шифрування. Якщо підбір ключа здійснюється методом повного перебору усіх можливих ключів, то така атака називається *лобовою*.

Здатність криптосистеми протистояти атакам на шифр говорить про її **стійкість** або **стійкість вживаного шифру**. Стійкість конкретного шифру оцінюється шляхом всіляких спроб його розкриття і залежить від кваліфікації криптоаналітиків, що атакують шифр.

Основним об'єктом криптографії є розробка таких методів перетворення інформації, що захищається, які не дозволили б порушникові витягнути її з перехоплюваних повідомлень. При цьому виходять з наступних вимог до криптографічного закриття інформації:

- ❒ складність і стійкість шифрування повинні вибиратися залежно від об'єму і ступеня секретності даних;
- ❒ надійність шифрування має бути такою, щоб секретність не порушувалася навіть у тому випадку, коли порушникові стає відомий метод шифрування;
- ❒ метод шифрування, набір використовуваних ключів і механізм їх розподілу не мають бути занадто складними;
- ❒ виконання процедур прямого і зворотного перетворень має бути формальним; ці процедури не повинні залежати від довжини повідомлень;
- ❒ помилки, що виникають в процесі виконання перетворення, не повинні поширюватися системою;
- ❒ надмірність, що вноситься процедурами захисту, має бути мінімальною.

Поява потужних суперкомп'ютерів, інформаційних технологій, що реалізовують мережеві і нейронні обчислення, зробила можливою дискредитацію криптосистем, що нещодавно вважалися практично нерозкривними.

Криптографічний захист інформації (КЗІ) – вид захисту інформації, який реалізується за допомогою перетворення відкритих даних з використанням спеціальних (ключових) даних з метою закриття (чи розкриття) змісту інформації, підтвердження її істинності, цілісності, авторства і т.д.

лання. При підтвердженні небезпеки усі необхідні дії виконуються автоматично. Швидкість виявлення і блокування загроз антивірусною хмарою істотно перевершує традиційний антивірусний аналіз. Якщо традиційне сигнатурне оновлення вимагає декількох годин, то при детектуванні загроз антивірусною хмарою йдеться про хвилини. При цьому, як показує практика, ймовірність помилкового спрацьовування мінімум в 100 разів нижче, ніж при традиційному детектуванні.

Збираючи і обробляючи інформацію, що поступає, антивірусний хмарний захист працює як потужна експертна система, що безперервно аналізує кіберкримінальну активність. Дані, необхідні для блокування атак, миттєво передаються усім учасникам хмари, запобігаючи масштабним вірусним епідеміям.

З використанням хмарної антивірусної технології можуть перевірятися і веб-ресурси, і поштові скриньки на наявність спаму. Тому антивірусні програми, засновані на хмарних обчисленнях, можуть *забезпечити багатовекторний характер захисту*. Багатовекторність може істотно полегшити побудову захисту локальних машин. Звичайно, класичні антивіруси також можуть проводити перевірку електронної пошти на наявність спаму, у них є і веб-контроль для спостереження за утримуваним відвідуваних користувачем сайтів. Але за це відповідають окремі, часто досить масивні модулі. А у разі хмарних антивірусів усе буде набагато легше.

Висока швидкість реакції на нові загрози і низький рівень помилкових спрацьовувань, забезпечувані хмарними антивірусними технологіями, потенційно можуть зробити їх незамінними в антивірусній індустрії. Хмарні технології вже зараз застосовуються в тому або іншому виді в антивірусних продуктах **Symantec**, **Agnitum**, **ESET**, **Panda Security** і ряду інших компаній.

Єдиним недоліком хмарного антивірусного захисту є залежність від стабільності роботи каналу зв'язку.

Проактивні методи виявлення

Проактивні методи виявлення вірусів набувають усього більшого поширення. В принципі, використання цієї технології дозволяє виявляти ще невідомі шкідливі програми. Існує декілька підходів до проактивного захисту.

Розглянемо два найбільш популярних підходу: *евристичні аналізатори* і *поведінкові блокіратори*.

Суть евристичних методів полягає в тому, що вирішення проблеми ґрунтується на деяких правдоподібних припущеннях, а не на строгих виводах з наявних фактів і передумов.

Якщо сигнатурний метод заснований на виділенні характерних ознак вірусу і пошуку цих ознак у файлах, що перевіряються, то **евристичний аналіз** грунтується на припущенні (дуже правдоподібному), що нові віруси часто виявляються схожі на яких-небудь із вже відомих. Таке припущення виправдовується наявністю в антивірусних базах сигнатур для визначення не одного, а

відразу декількох вірусів. Цей евристичний метод часто називають *пошуком вірусів, схожих на відомих*, або *статичним аналізом*.

Евристичний аналізатор (евристик)– цепрограма, яка аналізує програмний код об'єкту, що перевіряється, і за непрямими ознаками визначає, чи є об'єкт шкідливим. Робота евристичного аналізатора, як правило, починається з пошуку в програмному коді підозрілих ознак (команд), характерних для шкідливих програм.

Наприклад, багато шкідливих кодів шукають виконувані програми, відкривають знайдені файли і змінюють їх. Евристичний аналізатор переглядає код застосування і, зустрівши підозрілу команду, збільшує деякий «лічильник підозрілості» для цього застосування. Якщо після перегляду усього коду значення лічильника перевищує задане порогове значення, то об'єкт визнається підозрілим.

Достоїнствами статичного аналізу є простота реалізації, висока швидкість роботи, можливість виявлення нових, невідомих вірусів ще до того, як для них будуть виділені сигнатури.

Проте рівень виявлення нових шкідливих кодів залишається досить низьким, а ймовірність помилкових спрацьовувань – високою. Тому в сучасних антивірусах статичний аналіз використовуються у поєднанні з динамічним.

Ідея такого комбінованого підходу полягає в тому, щоб до того, як застосування буде запущено на комп'ютері користувача, емулювати його запуск в безпечному віртуальному оточенні, яке називається також буфером емуляції, або "пісочницею".

Динамічний евристичний аналізатор читає частину коду застосування в буфер емуляції антивіруса і за допомогою спеціальних прийомів емулює його виконання. Якщо в процесі цього псевдовиконання виявляються якінебудьпідозрілі дії, об'єкт визнається шкідливим і його запуск на комп'ютері користувача блокується.

На відміну від статичного методу, динамічний вимогливіший до ресурсів ПК, оскільки для аналізу доводиться використовувати безпечний віртуальний простір, а запуск застосування на комп'ютері користувача відкладається на час аналізу. Проте і рівень виявлення шкідників у динамічного методу значно вищий, ніж у статичного, а ймовірність помилкових спрацьовувань істотно менша.

Недоліки евристичних аналізаторів:

- ❒ неможливість лікування – черезпотенційні помилкові спрацьовування і можливе неточне визначення типу вірусу спроба лікування може привести до великих втрат інформації, чим із-за самого вірусу, а це неприпустимо;
- ❒ низька ефективність проти принципово нових типів вірусів.

Поведінковий блокіратор – цепрограма, яка аналізує поведінку запущеного застосування і блокує будь-які небезпечні дії.До основних шкідливих дій відносять:

- ❒ видалення файлу;

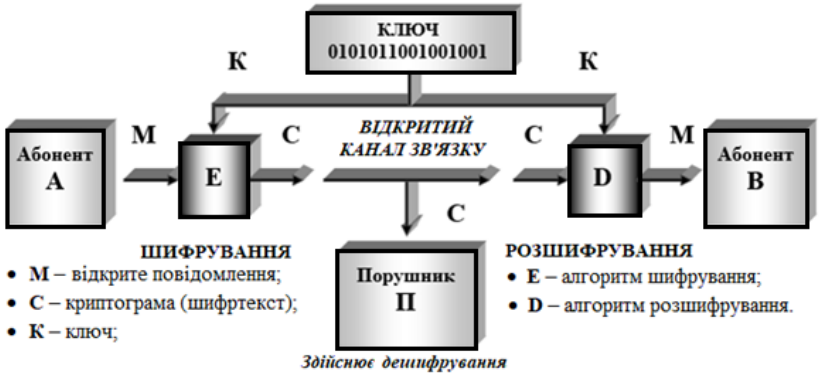
Основне завдання криптографії полягає в наступному:

Відомо:

- є два абоненти (*A* і *B*), які збираються обмінятися між собою конфіденційною інформацією;
- існує третя недружня особа – порушник (*П*), прагнучий оволодіти інформацією, що передається.

Вимагається:

- побудувати криптосистему, яка забезпечить задану гарантовану стійкість шифру.



Вихідне повідомлення є текстом, записаним в деякому **алфавіті** – кінцевій множині використовуваних символів (наприклад, український алфавітZ34$_{укр}$ містить 34 букви і пропуск, розширений код**ASCII** Z256$_{ASCII}$, використовуваний в комп'ютерах, включає 256 символів, двійковий алфавіт Z2 ={0, 1} і т.д.).

Відкритий текст – це впорядкований набір символів якого-небудь алфавіту.

Щоб приховати зміст повідомлення, абонент *A* перетворить відкритий текст ***M*** в криптограму ***C*** за допомогою спеціального алгоритму шифрування ***E*** і секретного ключа ***K***:

$$C=E_K(M).$$

Аналогічно, запис $M= D_K(C)$ означає, що відкритий текст ***M*** виходить з шифртекста ***C*** і ключа ***K***за допомогою алгоритму розшифрування ***D***.

При розробці шифрів керуються принципом, сформульованим у кінці XIX в. видатним нідерландським математиком і криптографом **ОгюстомКеркхоффом (1835-1903)**: «**стійкість шифру визначається тільки стійкістю ключа!**»

Суть принципу полягає в тому, що ніж менше секретів містить система, тим вище її безпека. Таким чином, якщо втрата будь-якого з секретів приведе до руйнування систе-

3. **Використовувати загальнодоступний канал зв'язку, але передавати по ньому потрібну інформацію в такому перетвореному виді, щоб відновити її міг тільки адресат.** Розробкою методів перетворень інформації з метою її захисту від незаконних користувачів займається ***криптографія***.

Криптологія (від греч. *kryptos* – таємний, *logos* – наука) – наука, що вивчає методи побудови і аналізу систем захисту інформації, заснованих на математичних перетвореннях даних з використанням деяких секретних параметрів. Такі системи називаються **криптографічними**. Вона розділяється на два напрями:

- **криптографія** – 1) галузь наукових знань, яка об'єднує принципи, методи і засоби перетворення даних з метою замаскувати зміст інформації, запобігти можливості її перехоплення і спотворення, захистити її від несанкціонованого доступу; 2) наука про методи і способи перетворення (шифрування) інформації з метою її захисту від несанкціонованих користувачів;
- **криптоаналіз** – наука (і практика її застосування) про методи і способи «зламування» шифрів, тобто несанкціонованого доступу аналітичним шляхом до інформації, захищеної криптографічними методами.

Іншими словами, **криптографія** – це *захист*, тобто розробка шифрів, а **криптоаналіз** – це *напад*, тобто атака на шифри.

Основні поняття криптографії:

- **шифр** – криптографічний прийом, пов'язаний із застосуванням деякого алгоритму перетворення символів (букв і цифр) вихідного (відкритого) тексту в зашифрований. В основі шифру лежить застосування ключів;
- **ключ** – послідовність символів, на основі якої виробляється шифрування і розшифрування даних;
- **шифрування** – процес застосування шифру до інформації, що захищається, тобто взаємно-однозначне перетворення повідомлення (*відкритого тексту*) в шифроване повідомлення (*криптограму*, *шифртекст*) за допомогою ключа і певних правил, що містяться в шифрі, з метою приховання змісту вихідного повідомлення від сторонніх осіб;
- **розшифрування** – процес, зворотний шифруванню, тобто перетворення шифрованого повідомлення у відкритий текст за допомогою певних правил, що містяться в шифрі;
- **криптосистема –** сукупність документів, пристроїв, устаткування і відповідних методів, використання яких в сукупності забезпечує засоби шифрування і розшифрування повідомлень;
- **стійкість шифру** – здатність шифру протистояти всіляким атакам на нього;
- **розкриття (злом) шифру** – процес отримання інформації (відкритого тексту), що захищається, з шифрованого повідомлення без знання застосованого шифру. Якщо кому-небудь із сторонніх вдається шляхом аналізу криптограми отримати відкритий текст, то говорять, що криптограма була **дешифрована** (розкрита).

- запис у файл;
- запис в певні області системного реєстру;
- відкриття порту на прослуховування;
- перехоплення даних, що вводяться з клавіатури;
- форматування носіїв;
- розсилка листів та ін.

Виконання такої дії окремо не дає приводу вважати програму шкідливою. Але якщо програма послідовно виконує декілька таких дій, наприклад перехоплює дані, що вводяться з клавіатури, і з певною частотою пересилає їх на якусь адресу в Інтернеті, значить, ця програма щонайменше підозріла.

На відміну від евристичних аналізаторів, де підозрілі дії відстежуються в режимі емуляції, поведінкові блокіратори працюють в реальних умовах.

Сучасні поведінкові блокіратори аналізують не окремі дії, а послідовність операцій, тобто висновок про небезпеку того або іншого застосування виноситься на основі складного аналізу, а це дозволяє значно скоротити кількість запитів до користувача і підвищити надійність детектування.

Поведінкові блокіратори здатні контролювати широкий спектр подій, що відбуваються в системі. Це передусім контроль небезпечної активності (аналіз поведінки усіх процесів, запущених в системі, збереження усіх змін, вироблюваних у файловій системі і реєстрі).

При виконанні деяким застосуванням набору підозрілих дій видається попередження користувачеві про небезпеку цього процесу. Окрім цього блокіратор дозволяє перехопити усі можливі впровадження програмного коду в чужі процеси, здатний виявити *руткити*, тобто програми, які приховують від користувача роботу шкідливого коду з файлами, папками і ключами реєстру, а також ховають запущені програми, системні служби, драйвери і мережеві з'єднання.

Особливо варто виділити таку функціональність поведінкових блокіраторів, як контроль цілісності застосувань і системного реєстру MS Windows. У останньому випадку блокіратор контролює зміни ключів реєстру і дозволяє задавати правила доступу до них для різних застосувань. Це дозволяє здійснити відкат змін після визначення небезпечної активності в системі. Таким чином можна відновлювати систему навіть після шкідливих дій невідомих програм, повернувши її до незараженого стану.

Як приклад ефективного поведінкового блокує нового покоління можна привести модуль проактивного захисту **Proactive Defence Module**, який включає усі перераховані вище можливості і, що особливо важливо, хорошу систему інформування користувача про те, в чому реально полягає небезпека тих або інших підозрілих дій.

Поведінковий блокіратор може запобігти поширенню як відомого, так і невідомого (написаного після створення блокіратора) вірусу, що є незаперечним достоїнством такого підходу до захисту.

Недоліком поведінкових блокіраторів залишається спрацьовування на дії ряду легітимних програм. Для прийнята остаточного рішення про шкідливість застосування вимагається втручання користувача, що припускає наявність у нього достатній кваліфікації.

Для оптимального антивірусного захисту потрібне поєднання проактивних і сигнатурних підходів. Максимального рівня виявлення загроз можна досягти, тільки комбінуючи ці методи. Прикладом успішного поєднання проактивних і сигнатурних методів може служити технологія **ThreatSense** компанії **Eset**.

Додаткові модулі

Практично будь-який антивірус сьогодні використовує усі відомі методи виявлення вірусів. Але одних засобів виявлення мало для успішної роботи антивіруса - дня того щоб чисто антивірусні засоби були ефективними, потрібні додаткові модулі, що виконують допоміжні функції.

Модуль оновлення. Щоб сигнатурний аналіз ефективно справлявся з самими останніми вірусами, антивірусні експерти постійно аналізують зразки нових вірусів і випускають для них сигнатури. Після цього головною проблемою стає доставка сигнатур на комп'ютери усіх користувачів, що використовують відповідну антивірусну програму. Саме це завдання і вирішує модуль оновлення.

Модуль планування. Існує ряд дій, які антивірус повинен виконувати регулярно: перевіряти увесь комп'ютер на наявність вірусів і оновлювати антивірусну базу. Модуль планування дозволяє налаштувати періодичність виконання цих дій.

Модуль управління. Основні властивості цього модуля управління:

- підтримка видаленого управління і налаштування – адміністраторбезпеки може запускати і зупиняти антивірусні модулі, а також міняти їх налаштування по мережі, не встаючи зі свого місця;
- захист налаштувань від змін – модульуправління не дозволяє локальному користувачеві змінювати налаштування або зупиняти антивірус, щоб користувач не міг ослабити антивірусний захист організації.

Карантин.У багатьох антивірусах серед допоміжних засобів є спеціальна технологія – *карантин*, яка захищає від можливої втрати даних в результаті дій антивіруса.

Наприклад, неважко представити ситуацію, при якій файл детектується як можливо заражений евристичним аналізатором і віддаляється згідно з налаштуваннями антивіруса.Проте евристичний аналізатор ніколи не дає 100% гарантії того, що файл дійсно заражений, а значить, з певною ймовірністю антивірус міг видалити незаражений файл. Або ж антивірус виявляє важливий документ, заражений вірусом, і намагається згідно з налаштуваннями виконати лікування, але з якихось причин відбувається збій і разом з вилікуваним вірусом втрачається важлива інформація.

4. КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

Розвиток і повсюдне впровадження сучасних інформаційних технологій значно підвищив уразливість інформації, циркулюючої в інформаційно-телекомунікаційних системах (ІТС). Однією з причин цього є масове використання для обробки інформації засобів обчислювальної техніки з програмним забезпеченням, що дозволяє порівняно легко спотворювати, копіювати або знищувати оброблювану інформацію, а також змінювати штатні алгоритми накопичення, зберігання, обробки і передачі інформації каналами зв'язку.

Принцип комплексного рішення завдань захисту інформації в ІТС припускає застосування разом з традиційними апаратно-програмними засобами і способами, організаційними і нормативно-правовими заходами захисту, також сучасних засобів, зокрема, криптографічних і стеганографічних.

Унікальність цих методів і засобів захисту інформації як безпосередньо в ІТС, так і в зовнішніх каналах зв'язку, полягає в тому, що вони забезпечують найнадійніший шлях захисту, бо охороняють безпосередньо саму інформацію, а не доступ до неї. Криптографічні методи дозволяють, окрім конфіденційності інформації, забезпечити її цілісність і достовірність, організувати процедуру автентифікації абонентів, що обмінюються інформацією.

Криптографічні методи захисту інформації засновані на тонких і не до кінця досліджених властивостях математичних об'єктів. У їх основі лежить ідея використання математичних перетворень. Подібні перетворення називаються криптографічними. Побудова криптографічних перетворень, як і методи їх використання, для захисту інформації не тривіальні.

4.1 Основні поняття криптології. Класифікація криптосистем

4.1.1 Поняття криптографії

Як передати важливу інформацію потрібному адресатові в таємниці від інших або як зберігати таку інформацію в ІТС, щоб до неї мало доступ обмежене коло осіб? Очевидно, є три можливих шляху:

1. **Створити абсолютно надійний, недоступний для інших канал зв'язку між абонентами.** При сучасному рівні розвитку науки і техніки зробити такий канал зв'язку між видаленими абонентами для неодноразової передачі великих об'ємів інформації практично нереально.
2. **Використовувати загальнодоступний канал зв'язку, але приховати сам факт передачі інформації.** Розробкою засобів і методів приховання факту передачі повідомлення займається ***стеганографія***.

16. Визначите склад системи виявлення таргетованих атак.
17. Що є агресивність програмних засобів?
18. Як здійснюється оцінка агресивности програмних засобів?
19. Як здійснюється виявлення факта інформаційного втручання?

Література

1. Антивирусная защита компьютерных систем. [Електронний ресурс]. – Режим доступу: http://www.intuit.ru/studies/courses/2259/155/info
2. Вирусы и средства борьбы с ними. [Електронний ресурс]. – Режим доступу: http://www.intuit.ru/studies/courses/1042/154/info
3. Гордон Я. Компьютерные вирусы без секретов / Я.Гордон. – М.: Новый издательский дом, 2004. – 320 с.
4. Грайворонський М.В. Безпека інформаційно-комунікаційних систем / М.В.Грайворонський, О.М.Новіков. — К: Видавнича група BHV, 2009. – 608 с
5. Информатика. Базовый курс. 3-е издание / Под ред. С.В.Симоновича. — СПб.: Питер, 2011. — 640 с.
6. Левцов В. Анатомия таргетированной атаки / В.Левцов, Н.Демидов; Журнал «Information Security/ Информационная безопасность», № 2, 2016, с. 36-39 [Електронний ресурс]. – Режим доступу : http://www.itsec.ru/articles2/Oborandteh/anatomiya-targetirovannoy-ataki, №3, 2016, с. 34-38[Електронний ресурс]. – Режим доступу: http://www.itsec.ru/articles2/Oborandteh/anatomiya-targetirovannoy-ataki-chast-2, №4, 2016, с. 40-45 [Електронний ресурс]. – Режим доступу: http://www.itsec.ru/articles2/Oborandteh/anatomiya-targetirovannoy-ataki-chast-3, №6, 2016, с. 18-23 [Електронний ресурс]. – Режим доступу: http://www.itsec.ru/articles2/target/anatomiya-targetirovannoy-ataki-chast-4
7. Фейнштайн К. Защита ПК от спама, вирусов, всплывающих окон и шпионских программ / Кен Фейнштайн; Пер. с англ. О.Б.Версиной. – М.: НТ Пресс, 2005. – 240 с.
8. Ленков С.В. Методы и средства защиты информации. В 2-х томах / С.В. Ленков, Д.А. Перегудов, В.А. Хорошко – К: Арий, 2008.
9. Блаватська Н.М. Програмне забезпечення систем захисту інформації / Н.М. Блаватська, В.Д. Козюра, В.О. Хорошко – К: Вид. ДУІКТ, 2011. – 330 с.

Від таких випадків бажано застрахуватися. Це можна зробити, якщо перед лікуванням або видаленням файлів зберегти їх резервні копії, тоді, якщо виявиться, що файл був видалений помилково або втрачена важлива інформація, завжди можна буде виконати відновлення з резервної копії.

Режими роботи антивірусів

Надійність антивірусного захисту забезпечується не лише здатністю відбивати будь-які вірусні атаки. Не менш важлива властивість захисту – її*безперервність* – антивірусповинен починати роботу по можливості до того, як віруси зможуть заразити тільки що включений комп'ютер, і вимикатися тільки після завершення роботи усіх програм.

Однак, з іншого боку, користувач повинен мати можливість у будь-який момент запитати максимум ресурсів комп'ютера для вирішення свого прикладного завдання і антивірусний захист не повинен йому заважати це зробити. Оптимальний вихід в цій ситуації –використання двох різних режимів роботи антивірусних засобів:

1) безперервна перевірка на наявність вірусів з невеликою функціональністю в режимі реального часу;
2) ретельна перевірка на наявність вірусів по запиту користувача.

Перевірка в режимі реального часу забезпечує безперервність роботи антивірусного захисту. Це реалізується за допомогою обов'язкової перевірки усіх дій, що здійснюються іншими програмами і самим користувачем, на предмет шкідливості незалежно від їх вихідного розташування (жорсткому диску, зовнішніх носіях інформації, інших мережевих ресурсах або в оперативній пам'яті). Також перевірці піддаються усі побічні дії через треті програми. Режим постійної перевірки захисту системи від зараження має бути включений з моменту початку завантаження ОС і вимикатися тільки в останню чергу.

Перевірка на вимогу.В деяких випадках наявність постійно працюючої перевірки в режимі реального часу може бути недостатньо. Допустимо, що на комп'ютер був скопійований заражений файл, виключений з постійної перевірки зважаючи на великі розміри, і, отже, вірус в нім виявлений не був. Якщо цей файл на даному комп'ютері запускатися не буде, то вірус може виявити себе тільки після пересилки його на інший комп'ютер, що може сильно пошкодити репутації відправника – розповсюджувача вірусів. Для виключення подібних випадків використовується другий режим роботи антивіруса - перевірка на вимогу.

Для такого режиму користувач сам вказує, які файли, каталоги або області диска необхідно перевірити, і час, коли треба провести таку перевірку, – увигляді розкладу або разового запуску вручну. Рекомендується перевіряти усі чужі зовнішні носії інформації, такі як дискети, компакт-диски, флеш-накопичувачі, кожного разу перед читанням інформації з них, а також увесь свій жорсткий диск не рідше за один раз в тиждень.

Тестування роботи антивіруса. Після того, як антивірус встановлений і

налагоджений, необхідно переконатися, що усе зроблено правильно і антивірусний захист працює. Як перевірити роботу антивіруса?

Використовувати для тестування справжні віруси украй небезпечно. Якщо користувач неправильно виконав установку або налаштування антивіруса, то в процесі такого тестування він може заразити свій комп'ютер, втративши в результаті дані або ставши джерелом зараження для інших комп'ютерів.

Потрібний такий спосіб тестування антивірусів, який був би безпечним, але давав чітку відповідь на питання, чи коректне працює антивірус.

Враховуючи важливість проблеми, організація **EICAR** за участю антивірусних компаній створила спеціальний тестовий файл, названий **eicar.com**.

Файл не виконує ніяких шкідливих дій, а просто виводить на екран рядок «**EICAR - STANDARD - ANTIVIRUS - TEST - FILE!**».Отримати файл можна на сайті організації за адресою **http://www.eicar.org/anti_virus_ test_ file.htm**.

Файл eicar.com дозволяє протестувати, як антивірус справляється з файловими вірусами і близькими по структурі шкідливими програмами – більшістю троянов, деякими хробаками.

Антивірусні комплекси

Другий спосіб оптимізації роботи антивіруса – цестворення різних його версій для комп'ютерів, що служать різним цілям. Частенько вони відрізняються лише наявністю тих або інших специфічних модулів і відмінністю в інтерфейсі, тоді як безпосередньо антивірусна перевірка здійснюється однією і тією ж підпрограмою, званою *антивірусним ядром*.

Антивірусний комплекс–це набір антивірусів, що використовують однакове антивірусне ядро, який призначений для вирішення практичних проблем із забезпечення антивірусної безпеки комп'ютерних систем.

Всяка локальна мережа, як правило, містить комп'ютери двох типів: робочі станції, за якими безпосередньо сидять люди, і мережеві сервери, використовувані для службових цілей. Відповідно до характеру виконуваних функцій сервери діляться на:

- ❒ *мережеві*, які забезпечують централізоване сховище інформації: *файлові сервери*, сервери застосувань та ін.;
- ❒ *поштові*, на яких працює програма, що служить для передачі електронних повідомлень від одного комп'ютера до іншого;
- ❒ *шлюзи*, що відповідають за передачу інформації з однієї мережі в іншу. Наприклад, шлюз потрібний для з'єднання локальної мережі з Інтернетом.

Відповідно, розрізняють чотири види антивірусних комплексів – длязахисту робочих станцій, файлових серверів, поштових систем і шлюзів.

Робочі станції – цекомп'ютери локальної мережі, за якими безпосередньо працюють користувачі. Головним завданням комплексу для захисту робочих станцій є *забезпечення безпечної роботи на даному комп'ютері*– для цього потрібна перевірка в режимі реального часу, перевірка на вимогу і перевірка локальної електронної пошти.

зараженню файлів або операційної системи шкідливим кодом. Складається з підпрограм, які намагаються виявити, запобігти розмноженню і видалити комп'ютерні віруси і інше шкідливе програмне забезпечення. До антивірусів відносяться сканери, ревізори, монітори і вакцини.

5. **Цільова або таргетованаатака** – це безперервний процес несанкціонованої активності в інфраструктурі системи, що атакується, який видалено управляється вручну в реальному часі. **APT – Advanced Persistent Threat** (*передова постійна загроза*) – це комбінація утиліт, шкідливого ПЗ, механізмів використання уразливостей нульового дня, інших компонентів, спеціально розроблених для реалізації цільової атаки.
6. Цільова атака, як правило, у своєму розвитку проходить через чотири фази (життєвий цикл): підготовка, проникнення, поширення і досягнення цілей. Інструментами атаки є командний центр, інструменти проникнення (експлойт, валідатор, завантажувач, модуль доставки), тело вірусу.
7. Комплексна стратегія протидії цільовим атакам включає чотири важливі елементи системи захисту: *запобігання* – недопущення початку і розвитку атаки; *виявлення* – виявлення слідів атаки, розпізнавання ознак, зв'язку усіх деталей атаки в єдину картину; *реагування* – у разі підтвердження факту атаки визначаються наслідки і кроки по їх *усуненню*; *прогнозування* – реалізація проактивних заходів, що дозволяють істотно утруднити порушникам підготовку і проведення атаки.

Питання для самоконтролю

1. Що є шкідливими програмами? Якої шкоди вони завдають комп'ютерній системі? Хто і з якою метою створює шкідливе ПЗ?
2. Поясните основні способи проникнення шкідливого ПЗ в комп'ютерну систему.
3. Як класифікується шкідливе ПЗ?
4. Що є комп'ютерними вірусами, мережевими хробаками, троянськими програмами і як вони класифікуються?
5. У чому проявляються симптоми зараження комп'ютерної системи шкідливим ПЗ?
6. Як підрозділяються методи захисту від шкідливого ПЗ?
7. Що таке антивірусна програма і які функції вона виконує?
8. Як працюють сканери антивірусних програм?
9. Як виявляється зараження комп'ютерної системи невідомим шкідливим ПЗ?
10. У яких цілях використовуються міжмережеві екрани (файрволи)?
11. Які вимоги пред'являються до сучасної антивірусної програми?
12. Що таке таргетована атака? Які її основні ознаки?
13. Визначите життєвий цикл таргетованої атаки.
14. Який інструментарій використовується в цільовій атаці?
15. Що є комплексна стратегія протидії цільовим атакам?

Замінюється один з підрядків контрольованих даних на випадкову підрядок (вносяться зміни в контрольовані дані).

Порівняння детекторів зі зміненим рядком контрольованих даних.

Якщо будь-який з детекторів фіксує зміну в контрольованих даних, вносяться зміни до звіту про хід експерименту.

4. Виконується порівняння значень ймовірності P_F, отриманих на першому кроці в ході експерименту зі значенням P_F.

Експеримент проводився з використанням тільки одного масиву детекторів. Рядок контрольованих даних генерувався випадковим чином, а зміни стосувалися лише одного підрядка довжиною 32 біта. Даний клас експериментів показав, що експериментальні і теоретичні значення знаходяться в межах допустимого відхилення (числа в дужках в табл. 18.2 - стандартне відхилення), що дозволило підтвердити справедливість зроблених припущень.

Висновки

1. **Шкідлива програма** – будь-яке програмне забезпечення, призначене для діставання несанкціонованого доступу до обчислювальних ресурсів комп'ютера або до інформації, що зберігається в комп'ютері, з метою несанкціонованого використання ресурсів ЕОМ або спричинення шкоди (завдання збитку) власникові інформації, і/або власникові ЕОМ, і/або власникові мережі ЕОМ, шляхом копіювання, спотворення, видалення або підміни інформації. Шкідливі програми призначені *для діставання несанкціонованого доступу* до інформації в обхід існуючих правил розмежування доступу.

2. **Комп'ютерний вірус** – програмний код, що мимоволі поширює свої копії по ресурсах локального комп'ютера з метою подальшого запуску свого коду при яких-небудь діях користувача і подальшого впровадження в інші ресурси комп'ютера. **Мережеві хробаки**– програми, що поширюють свої копії по локальних і/або глобальних мережах з метою проникнення на видалені комп'ютери, запуску своєї копії на видаленому комп'ютері, подальшого поширення на інші комп'ютери мережі, транспортування іншого шкідливого ПЗ. **Троянські програми**– програми, що здійснюють різні несанкціоновані користувачем дії: збір інформації і її передачу порушникові, її руйнування або зловмисну модифікацію, порушення працездатності комп'ютера, використання ресурсів комп'ютера в непристойних цілях. **Шкідливі утиліти**(Malicious tools) - програми, розроблені для автоматизації створення вірусів, черв'яків або троянських програм, організації DoS -атак на видалені сервера, злому інших комп'ютерів.

3. Для захисту від шкідливого ПЗ застосовуються юридичні (поліцейські), освітні і технічні методи (програмно-апаратні антивірусні засоби).

4. **Антивірусна програма** – програма для виявлення комп'ютерних вірусів і лікування інфікованих файлів, а також для профілактики - запобігання

Мережеві сервери– цекомп'ютери, спеціально виділені для зберігання або обробки інформації. Вони зазвичай не використовуються для безпосередньої роботи за ними, і тому, на відміну від робочих станцій, перевірка електронної пошти на наявність вірусів тут не потрібна. Отже, антивірусний комплекс для файлових серверів повинен проводити перевірку в режимі реального часу і перевірку на вимогу.

Антивірусний комплекс для захисту **поштових систем** призначений для перевірки усіх електронних листів, що проходять, на наявність в них вірусів. Тобто перевіряти інші файли, розміщені на цьому комп'ютері, він не зобов'язаний (для цього існує комплекс захисту мережевих серверів). Тому до нього пред'являються вимоги по наявності програми для перевірки усієї поштової кореспонденції, що приймається і відправляється, в режимі реального часу і додатково механізму перевірки на вимогу поштових баз даних.

Аналогічно, відповідно до свого призначення, антивірусний комплекс для **шлюзу** здійснює тільки перевірку даних, що проходять через шлюз.

Оскільки усі вище перелічені комплекси використовують сигнатурний аналіз, то в обов'язковому порядку в них повинен входити засіб для підтримки антивірусних баз в актуальному стані, тобто механізм їх оновлення. Додатково часто виявляється корисним модуль для видаленого централізованого управління, який дозволяє системному адміністраторові зі свого робочого місця налаштовувати параметри роботи антивіруса, запускати перевірку на вимогу і оновлення антивірусних баз.

Додаткові засоби захисту

Можливості антивірусів розширюють додаткові засоби захисту від шкідливих програм і небажаної кореспонденції.Такими засобами захисту є:

- ❒ оновлення ПЗ, що усувають уразливості в операційній системі та інших застосуваннях, через які можуть проникати віруси;
- ❒ брандмауери (міжмережеві екрани, файрволи) – програми, що захищають від атак по мережі;
- ❒ засоби боротьби із спамом.

Оновлення ПЗ. Шкідливі програми нерідко проникають на комп'ютери через уразливості в ОС або встановлених програмах. Причому найчастіше шкідливі програми використовують уразливості операційної системи **MS Windows**, пакету застосувань **MS Office**, браузеру **Internet Explorer** і поштової програми **Outlook Express**.

Щоб не дати вірусам можливості використовувати уразливість, ОС і ПЗ треба оновлювати. Виробники, як правило, раніше вірусописьменників дізнаються про «діри» у своїх програмах і завчасно випускають для них виправлення.

Для завантаження і установки оновлень в більшості програм і систем є вбудовані засоби.

Брандмауери. Для того, щоб видалено скористатися уразливістю в про-

грамному забезпеченні або операційній системі, треба встановити з'єднання і передати спеціально сформований пакет даних. Від таких спроб проникнення і зараження можна захиститися шляхом заборони певних з'єднань. Задачу контролю з'єднань успішно вирішують програми-брандмауери (міжмережеві екрани).

Брандмауер – цепрограма, яка стежить за мережевими з'єднаннями і приймає рішення про дозвіл або заборону нових з'єднань на підставі заданого набору правил.Правило брандмауера задається декількома атрибутами:

- *застосування* – визначаєпрограму, до якої відноситься правило, так що одні і ті ж дії можуть бути дозволені одним програмам і заборонені іншим. Наприклад, отримувати і відправляти пошту розумно дозволити тільки поштовому клієнтові;
- *протокол* – визначаєпротокол, використовуваний для передачі даних. Зазвичай можна вибрати між двома протоколами – **TCP** і **UDP**;
- *адреси* – визначає, для з'єднань з яких адрес або на які адреси діятиме правило;
- *порт* – задаєномери портів, на які поширюється правило;
- *напрям* – дозволяєокремо контролювати з'єднання, що входять і витікають;
- *дія* – визначаєреакцію на виявлення з'єднання, відповідного іншим параметрам. Реакція може бути *наступною*:*дозволити*, *заборонити* або *запитати* у користувача.

Необов'язково задавати конкретні значення усім атрибутам правила. Можна створити правило, яке заборонятиме з'єднання, що входять, на TCP – порт 111 для усіх застосувань або дозволяти будь-які витікаючи з'єднання для програми Internet Explorer.

Для боротьби з вірусами брандмауери можуть застосовуватися таким чином:

По-перше, брандмауер можна успішно використовувати для захисту від шкідливих програм, які поширюються безпосередньо по мережі, використовуючи уразливості в ОС. Наприклад, хробак **Sasser** атакує службу **Windows LSASS** через TCP-порт 445. Для захисту від хробака досить створити в брандмауері правило, що забороняє з'єднання, що входять, на цей порт. Брандмауер можна використовувати і для захисту від атак невідомих вірусів.

Другий спосіб застосування брандмауерів для захисту від шкідливих програм полягає в контролі витікаючих з'єднань. Багато троянських програм, та і хробаки після виконання шкідливої функції прагнуть подати сигнал авторові вірусу. Наприклад, троянська програма, що краде паролі, намагатиметься переслати усі знайдені паролі на певний сайт або поштову адресу. Для того, щоб перешкодити цьому, можна налаштувати брандмауер на блокування усіх невідомих з'єднань: дозволити тільки з'єднання від довірених програм, таких як використовуваний браузер, поштовий клієнт, програма миттєвого обміну повідомлень, а усі інші з'єднання заборонити.

Експериментальна перевірка алгоритму

У табл. 3.1 наведені для порівняння експериментальні та теоретичні дані залежності ймовірності не виявлення випадкової зміни в рядку (P_F)від кількості детекторів (N_R).У цій таблиці також наведені експериментальні та теоретичні дані про початкову потужність сукупності детекторів (N_{R_0}),що дозволяє оцінити обчислювальну складність алгоритму.

Таблиця3.1. Теоретичні та експериментальні значення N_{R_0} і P_F

N_S	Експериментальне значення N_{R_0}	Теоретичне значення N_{R_0}	Експериментальне значення P_F
8	69 (6,06)	69	0,085 (0,009)
16	105 (11,99)	105	0,104 (0,010)
24	156 (20,09)	158	0,110 (0,011)
32	240 (32,49)	239	0,099 (0,099)
40	360 (53,49)	361	0,107 (0,010)
48	549 (82,98)	545	0,133 (0,011)
56	829 (133,06)	823	0,109 (0,010)
64	1253 (218,77)	1243	0,124 (0,010)
72	1876 (318,55)	1876	0,112 (0,010)
80	2872 (495,61)	2833	0,116 (0,010)
88	4327 (781,78)	4277	0,130 (0,011)
96	6618 (1343,56)	6458	0,130 (0,011)
104	9903 (2082,86)	9750	0,135 (0,011)
112	15074 (3140,29)	14722	0,124 (0,010)
120	22878 (5283,80)	22228	0,154 (0,011)
128	34915 (8513,26)	33561	0,157 (0,012)

В експериментах приймалося кількість детекторів (N_R) рівним 46 (кожен з яких є 32-бітовим рядком). Імовірність невиявлення випадкової зміни в рядку (P_F) вибрана рівною 0,1.

Алгоритм проведення експерименту наступний:

1. Фіксується ймовірність P_F дорівнює 0,1.
2. Обчислюється ймовірність P_m, використовуючи дані $m = 2$, $l = 32$, $r = 8$. Обмеження $r = 8$ дозволяє досліджувати алгоритм при випадкових змінах одного байта.
3. Обчислюється значення N_R на базі P_F і P_m.

4. Далі в циклі (близько 1000 повторів) виконуються наступні дії.

Генерується N_S випадкових рядків (кожна довжиною 32 біта).

Визначається N_{R_0} за допомогою генерації випадкових рядків до тих пір, поки не буде знайдено N_R різних детекторів.

Виконується тестування детекторів.

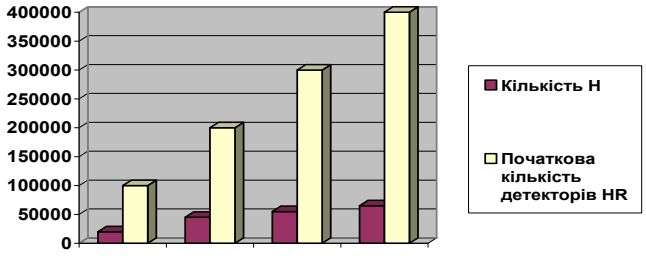


Рис. 3.5. Залежність потужності початкової сукупності від кількості контрольованих підрядків

Основні властивості алгоритму

Наведені дослідження дозволяють визначити основні властивості описаного алгоритму. Визначити необхідно ймовірність виявлення факту інформаційного впливу P_f, можна обчислити необхідну кількість детекторів як функції від числа N_S. Тим часом, підвищення ймовірності виявлення пов'язано зі збільшенням обчислювальної складності алгоритму.

N_R не залежитьвід N_S для заданої ймовірності P_m і P_f. Це дає можливість не змінювати кількість детекторів зі зростанням кількості контрольованих підрядків, що, в свою чергу, дозволяє організувати контроль за досить великими масивами даних.

Якщо N_R, P_f и P_m – фіксовані, то N_{R_0} зростає експоненціально зі зростанням N_S, що значно збільшує обчислювальну складність алгоритму.

Ймовірність виявлення факту інформаційного впливу зростає експоненціально зі зростанням числа функціонуючих незалежних копій алгоритму контролю. Якщо N_f – кількість функціонуючих копій, то ймовірність того, що всі алгоритми не виявлять вплив $P_{all} = (P_f)^{N_t}$. Це є ключовою властивістю алгоритму. Експериментально встановлено, що одна копія алгоритму з 46 детекторами визначає факт інформаційного впливу з ймовірністю 90,6%. При тих же характеристиках виявлення інформаційного впливу, але при задіяних десяти копіях алгоритму контролю ($N_t = 10$)достатньо всього чотирьох детекторів на кожну копію алгоритму.

Процес детектування є симетричним. Якщо зміни виявлені, то вони можуть належати як до сукупності детекторів (при незмінній сукупності підрядків контрольованих даних), так і власне до контрольованих даними (при незмінності сукупності детекторів).

Деякі шкідливі програми пасивно чекають з'єднання на якомусь з портів. Якщо з'єднання, що входять, дозволені, то автор шкідливої програми зможе через деякий час звернутися на цей порт і забрати потрібну йому інформацію або ж передати шкідливій програмі нові команди. Щоб цього не сталося, брандмауер має бути налаштований на заборону з'єднань, що входять, на усі порти, окрім фіксованого переліку портів, використовуваних відомими програмами або ОС.

Останнім часом широко поширені універсальні захисні програми, що об'єднують можливості брандмауера н антивіруса, наприклад **Norton Internet Security**, **McAfee Internet Security**та ін.

Засоби захисту від небажаної кореспонденції. Для вирішення проблеми захисту від *спаму* використовуються спеціальні антиспамові фільтри. Для фільтрації небажаної пошти в антиспамових фільтрах застосовується декілька методів:

- ❒ *чорні і білі списки адрес*. Чорний список - це список тих адрес, листи з яких фільтр відбраковує відразу, не застосовуючи інших методів. У цей список треба заносити адреси, якщо з них постійно приходять непотрібні або заражені листи. Білий список – цесписок адрес добре відомих користувачеві людей або організацій, які передають тільки корисну інформацію. Антиспамовий фільтр можна налаштувати так, що прийматимуться тільки листи від адресатів з білого списку;
- ❒ *бази даних зразків спаму*. Як і антивірус, антиспамовий фільтр може використовувати базу даних зразків небажаних листів для видалення листів, відповідних цим зразкам;
- ❒ *аналіз службових заголовків*. У листі у відносно прихованій формі зберігається службова інформація про те, з якого сервера було доставлено лист, який адресат є реальним відправником та ін. Використовуючи цю інформацію, антиспамовий фільтр може вирішувати, є лист спамом або ні. Наприклад, деякі поштові сервери, часто використовувані для розсилки спаму, заносяться в спеціальні загальнодоступні чорні списки, і якщо лист був доставлений з такого сервера, цілком імовірно, що це спам. Інший варіант перевірки – запитати у поштового сервера, чи дійсно існує адресат, вказаний в листі як відправник. Якщо такого адресата не існує, значить, лист, швидше за все, є небажаним;
- ❒ *самонавчання*. Антиспамові фільтри можна навчати, вказуючи вручну, які листи є нормальними, а які небажаними. Через деякий час антиспамовий фільтр починає з великою достовірністю самостійно визначати небажані листи по їх схожості на попередній спам і несхожість на попередні нормальні листи.

Використання антиспамових фільтрів допомагає захиститися і від деяких поштових хробаків. Найочевидніше застосування – цепри отриманні першого зараженого листа (у відсутність антивіруса це можна визначити за непрямими ознаками) відмітити його як небажане – інадалі усі інші заражені листи будуть заблоковані фільтром.

Більше того, поштові черв'яки відомі тим, що мають велику кількість модифікацій, що трохи відрізняються один від одного. Тому антиспамовий фільтр може допомогти і в боротьбі з новими модифікаціями відомих вірусів з самого початку епідемії. У цьому сенсі антиспамовий фільтр навіть ефективніше за антивірус, оскільки необхідно дочекатися оновлення антивірусних баз, щоб антивірус зміг виявити нову модифікацію.

Основні вимоги до антивіруса

1. Надійність роботи антивіруса і простота використання.

2.Якість захисту.Антивірус даремний, якщо він не в змозі забезпечувати достатній рівень захисту від шкідливих програм.

Якість захисту складається з наступних характеристик:

- рівень детектування шкідливих програм,
- частота і регулярність виходу оновлень,
- можливість коректного видалення вірусного коду з системи,
- ресурсомісткість,
- уміння захищати не лише від вже відомих, але і від нових вірусів і троянських програм.

3. Комплексність захисту.Під постійним контролем повинні знаходитися усі області комп'ютера, усі типи файлів, усі елементи мережі, які можуть стати об'єктом вірусної атаки. При цьому потрібне уміння виявляти шкідливий код і в усіх каналах його можливого проникнення (пошта, WWW, FTP і т.д.).

Антивірус для комплексного захисту комп’ютерної системи від вірусів та інших типів шкідливих програм, а також від хакерських атак і спаму повинен будуватися за модульним принципом і містити наступні компоненти:

- **Файловий Антивірус**контролює файлову систему комп'ютера: перевіряє усі файли, що відкриваються, запускаються і зберігаються, на комп’ютері та на усіх приєднаних дисках. Кожне звернення до файлу перехоплюється антивірусом і файл перевіряється на присутність відомих вірусів.

P_m - ймовірність збігу випадкового рядка з будь-яким з N_S рядків і дорівнює $(1-P_m)^{N_S}$;

P_f - ймовірність того, що N_r детекторів не виявлять факт інформаційного впливу.

Якщо P_m достатньо мала, а N_S велика, то;

$$f \approx e^{-P_m/N_S} \tag{3.1}$$

і

$$N_R = N_{R_0} \cdot f; \tag{3.2}$$

$$P_f = (1-P_m)^{N_R}.$$

Якщо P_m достатньо мала, а N_R велика, то;

$$P_f \approx e^{-P_m N_R},$$

звідси,

$$N_R = N_{R_0} \cdot f = \frac{-\ln P_f}{P_m}.$$

Розв’яючи рівняння (18.10) і (18.11) відносно N_{R_0} отримаємо:

$$N_{R_0} = \frac{-\ln P_f}{P_m \cdot (1-P_m)^{N_S}}.$$

Дана формула дозволяє визначити необхідну початкову кількість рядків для формування сукупності детекторів (N_{R_0})як функцію від ймовірності виявлення факту інформаційного впливу $(1-P_f)$, кількості контрольованих підрядків N_S і правила обчислення P_m. N_{R_0} буде мінімальному тому випадку, якщо для обчислення P_m вибрано наступне правило:

$$P_m = \frac{1}{N_S}.$$

Наведені викладки дозволяють зробити приблизні висновки щодо обчислювальної складності запропонованого алгоритму.

Як уже зазначалося, алгоритм складається з двох основних етапів:

1. формування сукупності детекторів;
2. тестування контрольованих даних.

Будемо вважати, що кожен з кроків займає фіксований час виконання. У цьому випадку складність (3.1) буде пропорційна кількості рядків в R_0 (N_{R_0}) і в S (N_S).Складність (3.2) пропорційна кількості детекторів в R (N_R)і кількості підрядків в S (N_S). Залежність N_S від N_{R_0} представлена на рис. 3.5.

На другому кроці алгоритму проводиться тестування контрольованих даних шляхом порівняння підрядка із сукупності S з підрядком сукупності R. Для цього вибирається один підрядок з S і один підрядок з R. Порядок доступу до рядків може бути як детермінованим, так і випадковим. У нашому прикладі для рядків-детекторів і S була обрана стратегія FIFO («першим прийшов - першим обслуговується»).

Ключовою ланкою алгоритму є оператор порівняння рядків. Визначимо правило функціонування оператора порівняння. Початковими параметрами алгоритму є: r – кількість співпадаючих позицій в рядках. Оператори для рядків x і y матимуть справжнє значення в тому випадку, якщо в рядках x і y співпадає менше ніж r послідовних символів (рис. 3.5).

X: ABADCBAB
Y: CACDCBBA

Рис. 3.5. Приклад правила порівняння для рядків x і y заданих на алфавіті (A, B, C, D). Оператор порівняння буде мати справжнє значення для $r = 3$ і менше

Імовірність P_m того, що два випадкові рядки співпадуть менш ніж в r позиціях, якщо m – кількість символів базового алфавіту і l – довжина рядка, дорівнюватиме:

$$P_m \approx m^{-r}\left[(l-r)(m-1)/m+1\right].$$

Таблиця 3.1 ілюструє результати обчислення P_m для різних значень r, l, m.

Перші чотири рядки таблиці відображають бітові зміни в межах одного байта. Видно, що ймовірність P_m збільшується зі збільшенням довжини контрольованого рядка. Рядки 1 і 5 таблиці відображають експоненціальне зростання P_m зі збільшенням r. Останні вісім рядків таблиці демонструють стрімке зменшення P_m при збільшенні базового алфавіту.

Обчислення ймовірності виявлення змін в контрольованих даних

Раніше зазначалося, що визначення змін в контрольованих даних носить характер імовірності. Використовуючи запропонований алгоритм, необхідно визначити достатню кількість детекторів і їх довжину з тим, щоб домогтися прийнятних значень P_m.

Приймемо наступні позначення:

N_{R_0} - початкова кількість згенерованих випадковим чином рядків;

N_R - кількість детекторів;

N_S - кількість підрядків, на які розбивається рядок контрольованих даних;

❒ **Поштовий Антивірус**перевіряє поштові повідомлення комп'ютера (ті що входять і витікають): аналізує електронні листи на присутність шкідливих програм. Крім того, аналізує поштові повідомлення на предмет фішинг-шахрайства.

❒ **Веб-антивірус**перехоплює і блокує виконання скрипта, розташованого на веб-сайті, якщо він представляє загрозу. Строгому контролю піддається увесь HTTP-трафік. Крім того, аналізує веб-сторінки на предмет фішинг-шахрайства.

❒ **ІМ -антивірус**забезпечує безпеку роботи з Інтернет-пейджерами та іншими програмами, призначеними для швидкого обміну повідомленнями.

❒ **Контроль програм**реєструє дії, що здійснюються програмами в системі, і регулює діяльність програм, виходячи з того, до якої групи компонент відносить цю програму. Для кожної групи програм заданий набір правил. Ці правила регламентують доступ програм до різних ресурсів.

❒ **Мережевий екран**забезпечує безпеку роботи в локальних мережах і Інтернеті. Проводить фільтрацію усієї мережевої активності згідно з правилами двох типів: правилам для програм і пакетним правилам.

❒ **Проактивний захист**дозволяє виявити нову шкідливу програму ще до того, як вона встигне завдати шкоди. Компонент заснований на контролі і аналізі поведінки усіх програм, встановлених на комп'ютері. На підставі виконуваних дій приймається рішення про те, є програма потенційно небезпечною або ні.

❒ **Захист від мережевих атак**запускається при старті ОС і відстежує в трафіку, що входить, активність, характерну для мережевих атак. Виявивши спробу атаки на комп'ютер, блокує будь-яку мережеву активність атакуючого комп'ютера.

❒ **Моніторинг мережі**– компонент, призначений для перегляду інформації про мережеву активність в реальному часі.

❒ **Анти-спам** вбудовується у встановлений на ПК поштовий клієнт і контролює усі поштові повідомлення, що поступають, на предмет спаму. Усі листи, що містять спам, позначаються спеціальним заголовком. Передбачена також можливість налаштування Анти-спаму на обробку спаму (автоматичне видалення, приміщення в спеціальну папку і т.д.). Компонент аналізує поштові повідомлення на предмет фішинг-шахрайства.

❒ **Анти-фішинг**– компонент, вбудований у веб-антивірус, анти-спам і ІМ-антивірус, який дозволяє перевіряти веб-адреси на приналежність до списків фішингових і підозрілих веб-адрес.

❒ **Анти-банер** блокує рекламну інформацію, розміщену на банерах, вбудованих в інтерфейс різних програм, встановлених на ПК, і що знаходяться в Інтернеті.

❒ **Батьківський контроль**компонент, що виконує функції контролю доступу користувачів комп'ютера до веб-ресурсів. Основним завданням є обмеження доступу до веб-сайтів, призначеним для дорослої аудиторії, зачіпа-

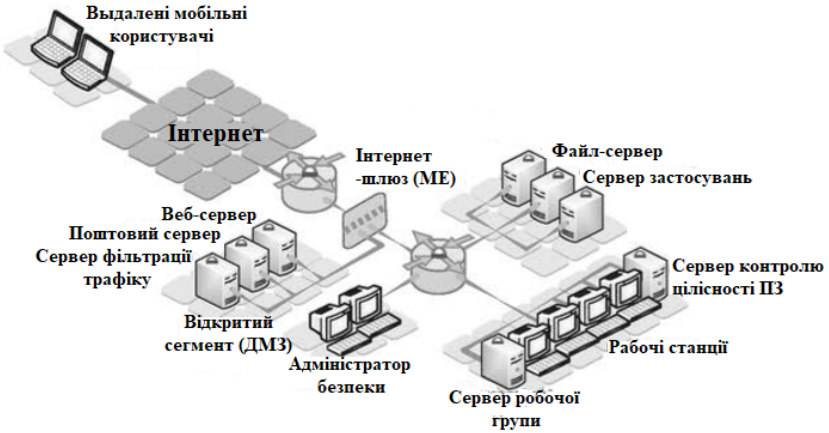
ють теми порнографії, зброї, наркотиків, що провокують жорстокість, насильство тощо, а також до веб-сайтів, які є потенційною причиною втрати часу (чати, ігрові ресурси) або грошей (інтернет-магазини, аукціони).

❒ **Безпечне середовище**дозволяє запускати програму у безпечному для системи оточенні (т.з. пісочниця). Це може бути використано для запуску потенційно небезпечного ПЗ або для анонімного веб-серфінгу.

❒ **Перевірка посилань**– доповненнядля браузерів, що перевіряє посилання на веб-сторінці, що переглядається, на предмет зараження веб-сторінок, на які вони посилаються.

Багаторівневий захист корпоративної інформації від шкідливих програм

Одна з головних переваг цього рішення – розглядпідсистеми захисту корпоративної інформації від шкідливих програм як багаторівневої системи.



Перший рівень включає засоби захисту від шкідливих програм, що встановлюються на стику з глобальними мережами (інтернет-шлюз і/або міжмережевий екран, публічні сервери (веб-, SMTP-, FTP-), що розміщуються в демілітаризованій зоні (ДМЗ)) і що здійснюють фільтрацію основних видів трафіку (HTTP, SMTP, FTP і т.д.). Антивірусні засоби, що встановлюються на міжмережевому екрані (МЭ), сумісні з **Check Point FireWall-1** і **Cisco PIX**, які входять до числа найпоширеніших міжмережевих екранів.

Другий рівень– засобизахисту, що встановлюються на внутрішніх корпоративних серверах і серверах робочих груп (файлових сховищах, серверах застосувань і т.д. далі).

Третій рівень– засобизахисту від шкідливих програм, що встановлюються на робочих станціях користувачів, у тому числі видалених і мобільних.

Переваги цього рішення полягають:

❒ у використанні продуктів світових лідерів;

❒ у централізованому управлінні усією підсистемою захисту від шкідливих програм;

Залежність ймовірності P_m від m, r, l:

m	r	l	P_m
2	8	32	0,0502023
2	8	64	0,108697
2	8	128	0,2151
2	8	256	0,391316
2	16	32	0,000137329
2	16	64	0,000381437
2	16	128	0,000869474
2	16	256	0,00184483
128	8	32	$3{,}33067 \cdot 10^{-16}$
128	8	64	$7{,}77156 \cdot 10^{-16}$
128	8	128	$1{,}66533 \cdot 10^{-15}$
128	8	256	$3{,}44169 \cdot 10^{-15}$
128	16	32	$\approx 0{,}0$
128	16	64	$\approx 0{,}0$
128	16	128	$\approx 0{,}0$
128	16	256	$\approx 0{,}0$

Наступний крок алгоритму полягає в генерації сукупності випадкових рядків (R_0)(в нашому випадку це 4-бітові рядки), а потім порівняння елементів сформованої сукупності R_0 з елементами сукупності S.

Рядки сукупності R_0, які співпадають з рядками S, вилучаються, а рядки, що не збігаються з будь-яким з елементів S, переміщаються в сукупність детекторів R.

Згідно з нашим прикладом (для S 1011011100110000) (рис. 3.4), R_0 містить чотири згенеровані випадковим чином 4-бітові рядки. В цьому випадку сукупність R включатиме рядок 1101, рядки ж 1000 і 1100 будуть виключені, оскільки вони збігаються з підрядками в сукупності S.

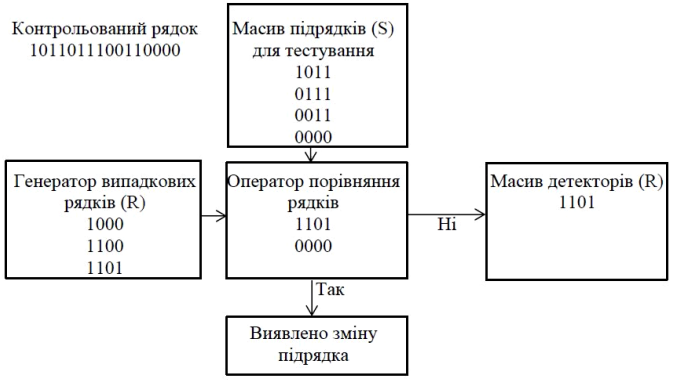


Рис. 3.4. Формування сукупності детекторів (для $r = 2$)

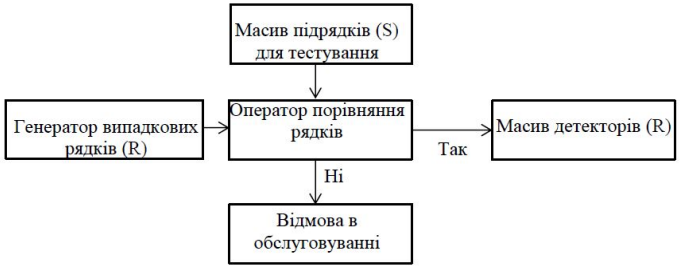


Рис. 3.2. Генерація масиву детекторів

Другий крок алгоритму полягає власне у контролі даних на базі створеної великої кількості детекторів. Цей крок представлений на рис. 3.3.

Рис. 3.3. Контроль даних з використанням масиву детекторів

Суть алгоритму полягає в наступному. Якщо уявити всі контрольовані дані як безліч рядків, що задані на кінцевому алфавіті, а зміни контрольованих даних представити у вигляді рядка, що не збігається з вихідними рядками, то можливо створити безліч рядків-детекторів для всіх рядків, що не входять у вихідні рядки контрольованих даних. Як буде показано далі, досить мала кількість рядків-детекторів можуть з високою ймовірністю сигналізувати про факт зміни контрольованих даних. Такий алгоритм має високий ступінь паралелізму, при цьому кожна копія алгоритму контролю повинна генерувати власну підмножину рядків-детекторів.

Під рядками-детекторами і рядками контрольованих даних в нашому випадку розуміються бітові послідовності, але це можуть бути і команди асемблера, шістнадцяткове представлення даних та ін..

Розглянемо весь обсяг контрольованих даних (файли на дисках, вміст оперативної пам'яті та ін.) як невпорядковану сукупність підрядків (термін «сукупність» використаний тому, що не виключається наявність елементів, які дублюються, об'єднаних в єдиний рядок даних). Перед виконанням операції формування сукупності детекторів розіб'ємо (логічно) єдиний рядок даних на ряд підрядків однакової довжини, наприклад, виконаємо для 32-бітового рядка (розбивається на 4-бітові підрядки). Це буде виглядати наступним чином:

0010 1000 1001 0000 0100 0010 1001 0011.

Сукупність підрядків (в нашому випадку це 4-бітові послідовності) S є неконтрольованими даними (S містить всі підрядки).

- ❒ у автоматичному оновленні антивірусних баз;
- ❒ у тісній взаємодії антивірусних засобів усіх рівнів підсистеми і т.д.

Усі ці переваги забезпечують високу ймовірність виявлення шкідливих програм.

3.3 Таргетовані атаки і захист від них

3.3.1 Поняття цільової атаки

Останніми роками застосування шкідливого ПЗ стало носити виборчий, цільовий характер.

Цільова або таргетована (від англ. *target* - ціль) **атака**– цебезперервний процес несанкціонованої активності в інфраструктурі системи, що атакується, який видалено управляється вручну в реальному часі.

Особливості цільової атаки:

1) це саме *процес*– діяльністьв часі, деяка спеціальна операція, а не просто разова технічна дія;

2) цей *процес спрямований для роботи в умовах конкретної інфраструктури*, покликаний здолати конкретні механізми безпеки, певні продукти, залучити у взаємодію конкретних співробітників;

3) ця *операція керується організованою групою професіоналів*, що має на озброєнні витончений технічний інструментарій.

Основні цілі таргетованих атак:

6. Державний сектор:
   - ❒ шпигунство;
   - ❒ маніпуляція інформацією;
   - ❒ порушення доступності online сервісів;
   - ❒ розкрадання персональних даних.
7. Телекомунікації:
   - ❒ атаки на корпоративних клієнтів;
   - ❒ маніпуляція поштовим сервером з метою соціальної інженерії;
   - ❒ маніпуляція розкрученими веб-ресурсами в цілях фішинга;
   - ❒ контроль білінга.
8. Фінанси:
   - ❒ розкрадання коштів;
   - ❒ розкрадання персональних даних.
9. Бізнес:
   - ❒ розкрадання коштів;
   - ❒ маніпуляція бізнес-процесом;
   - ❒ послаблення в конкурентній боротьбі;
   - ❒ шантаж, здирство;
   - ❒ тероризм;
   - ❒ розкрадання персональних даних.
10. Медицина:

- ❒ розкрадання інформації пацієнтів;
- ❒ атаки на телекероване медичне устаткування.

У ситуації цільової атаки не комп'ютерні системи «б'ються» один з одним, а люди – однінападають, інші – відбиваютьдобре підготовлений напад, що враховує слабкі сторони і особливості систем протидії.

З'явився термін **APT – Advanced Persistent Threat** (*передова постійна загроза*) – це комбінація утиліт, шкідливого ПЗ, механізмів використання уразливостей нульового дня, інших компонентів, спеціально розроблених для реалізації цільової атаки. Іншими словами, цільова або таргетована атака – це *процес*, *діяльність*, а APT – *технічний засіб*, що дозволяє реалізувати атаку.

Активне поширення цільових атак обумовлене у тому числі й сильним скороченням вартості та трудовитрат в реалізації самої атаки. Велика кількість раніше розроблених інструментів доступна хакерським угрупуванням, іноді відсутня гостра необхідність створювати екзотичні шкідливі програми з нуля. В більшості своїй сучасні цільові атаки побудовані на раніше створених експлойтах і шкідливому ПЗ, лише мала частина використовує абсолютно нову техніку, яка переважно відноситься до загроз класу APT. Іноді у рамках атаки використовуються і абсолютно легальні, створені для «мирних» цілей утиліти.

3.3.2 Фази цільової атаки

Цільова атака, як правило, у своєму розвитку проходить через чотири фази (життєвий цикл).

1. Перша фаза – підготовка – основне завдання першої фази знайти ціль, зібрати про неї досить детальній приватній інформації, спираючись на яку, виявити слабкі місця в інфраструктурі. Збудувати стратегію атаки, підібрати раніше створені інструменти, доступні на ринку, або розробити необхідні самостійно. Зазвичай плановані кроки проникнення будуть ретельно протестовані, у тому числі на невиявлення стандартними засобами захисту інформації.

стовувати в критичних системах. Однак оцінку агресивності необхідно проводити після сигнатурного аналізу, за допомогою якого ідентифікуються вже відомі віруси.

Використання описаного підходу до програмної реалізації оцінки агресивності програмних засобів дозволяє підвищити ймовірність виявлення руйнуючих програмних засобів і запобігти нанесенню шкоди системі.

3.5 Виявлення факта інформаційного втручання

Метод виявлення факту інформаційного вторгнення

В даний час більшість систем, призначених для вирішення завдань виявлення вторгнень, використовують методи визначення факту інформаційного впливу одного з трьох типів (можливе і комбіноване використання кількох методів): **сигнатурні методи** контролю цілісності даних (за основу методів взято пошук унікальних і незмінних фрагментів коду, властивих певному класу засобів впливу, наприклад, вірусам); **методи статистичного аналізу трафіку і класичні методи захистувід несанкціонованого доступу** (дискреційні і мандатні). Розглянутий метод і алгоритм відноситься до класу сигнатурних методів.

Метод грунтується на порівнянні великої кількості рядків-детекторів, створених випадковим чином (унікальних для кожного контрольованого рядка) і власне такого рядка.

Метод заснований на трьох базових принципах.

1. Кожна копія алгоритму детектування унікальна. Для захисту різних рядків необхідно кілька різних алгоритмів. Ідея полягає в тому, щоб забезпечувати контроль за кожним рядком за допомогою великої кількості унікальних детекторів.
2. Процес виявлення змін носить характер імовірності. Умова використання декількох різних наборів детекторів обумовлюється і характером імовірності методу. Так як поодинокі зміни в ланцюжках (рядках) можуть бути ефективно виявлені тільки з використанням великої кількості детекторів.
3. Стійка система повинна із заданою імовірністю забезпечувати будь-яку несанкціоновану активність по зміні інформаційних ланцюжків.

Алгоритм складається з двох кроків. На першому кроці проводиться генерація великої кількості детекторів. Кожен детектор являє собою рядок, що не збігається з контрольованим рядком даних. Даний крок алгоритму представлений на рис. 3.2.

Дизасемблери умовно можна розділити на «дурні» (dumb) і «розумні» (smart). «Дурні» дизасемблери навіть не намагаються вирішити наведені проблеми, тому результат роботи дизасемблера нагадує асемблерний текст досить віддалено. У той же час «дурні» дизасемблери перетворять машинний код дуже швидко (майже миттєво). До «дурних» дизассемблерів відносяться вбудовані дизасемблери налагоджувачів і антивірусних утиліт, які призначені для інтерактивного перегляду невеликих ділянок машинного коду.

«Розумні» дизасемблери перетворять машинний код в асемблерний настільки точно, що в окремих випадках повторне ассемблерування призводить до того ж самому машинного коду. «Розумні» дизасемблери розрізняють команди і дані, практично завжди правильно визначають межі машинних команд, виділяють в машинному коді окремі функції, відстежують перехресні посилання. Однак за такі можливості доводиться платити. Час дизассемблерування програми «розумним» дизассемблером може становити десять хвилин.

Наступний крок - квантова декомпозиція лістингу програми, в результаті лістинг перетворюється в квантову структуру програми. Квантова декомпозиція була запропонована М.М. Безруковим як метод моделювання програм, написаних мовою високого рівня. Стосовно програми на мові Асемблер введено поняття «квант 0-рівня» - це ділянка машинного коду, у якого відсутній базисний блок і є тільки оператор передачі управління. При квантовій декомпозиції ассемблерної програми виходить квантова структура, яка за своєю складністю значно простіша, ніж структура програми на мові високого рівня.

На основі бази небезпечних операторів проводиться аналіз зараженості квантів і розраховується показник зараженості програми.

Далі розраховується показник захоплення ресурсів системи. При цьому аналізується місце, зайняте на вінчестері. За допомогою програмної емуляції роботи системи оцінюється обсяг оперативної пам'яті, який потрібен досліджуваній програмі для її нормального функціонування, і аналізується завантаження процесора. На основі отриманих даних робиться розрахунок показника захоплення ресурсів системи.

У блоці аналізу функцій саморепродукції відбувається статистичний аналіз можливої саморепродуктивної діяльності досліджуваної програми, тобто періодично фіксується будь-яка спроба додавання свого машинного коду в інші програмні засоби системи досліджуваної програмою. Дослідження проводиться в ізольованій системі. За результатами контролю робиться висновок про наявність в алгоритмі досліджуваної програми функцій саморепродукції.

На основі розрахованих показників агресивності експертом оголошується заключний висновок про агресивність програми.

Запропонований підхід до програмної реалізації підтримки експертної системи дозволяє оцінити одну зі складових інформаційної безпеки - агресивність програмних засобів. При цьому її застосування є доцільним для сертифікації деяких видів програмного забезпечення, які планується викори-

❒ *Виявлення цілі.*Ціллю для атаки може стати будь-яка організація. А починається усе із замовлення, або загальної розвідки (моніторингу), в ході якої хакерські групи використовують загальнодоступні інструменти, такі як RSS-розсилки, офіційні Twitter-акаунти компаній, профільні форуми, де обмінюються інформацією різні співробітники. Це допомагає визначити жертву і завдання атаки, після чого ресурси групи переходжуватимуть до етапу активної розвідки.

❒ *Збір інформації.* . Основне завдання розвідки – збір цільової приватної інформації про жертву. Тут важливі усі дрібниці, які допоможуть виявити потенційні слабкі місця. У роботі можуть бути використані самі нетривіальні підходи для отримання закритих первинних даних, наприклад, соціальна інженерія. Розглянемо декілька техніки соціальної інженерії та інших механізмів розвідки, вживаних на практиці.

Способи проведення розвідки:

- *Інсайд*. Існує підхід з пошуком нещодавно звільнених співробітників компанії-цілі. Колишній співробітник компанії отримує запрошення на звичайну співбесіду на дуже принадну позицію. Досвідчений психолог-рекрут в змозі розговорити майже будь-якого співробітника, який бореться за позицію. Від таких людей отримують досить великий об'єм інформації для підготовки і вибору вектору атаки: від топології мережі й використовуваних засобах захисту до інформації про приватне життя інших співробітників.
  Кіберзлочинці можуть удаватися до підкупу потрібних ним людей в компанії, що володіють інформацією, або входити в круг довіри шляхом дружнього спілкування в громадських місцях.
- *Відкриті джерела*. Хакери можуть використовувати несумлінне відношення компаній до паперових носіїв інформації, які викидають в кошик без правильного знищення, серед сміття можуть бути знайдені звіти і внутрішня інформація або, наприклад, сайти компанії, які містять реальні імена співробітників в загальному доступі. Отримані дані можна буде комбінувати з іншою технікою соціальної інженерії.
  В результаті цієї роботи організатори атаки можуть мати досить повну інформацію про жертву, включаючи:
  - імена співробітників, e - mail, телефон;
  - графік роботи підрозділів компанії;
  - внутрішню інформацію про процеси в компанії;
  - інформацію про бізнес-партнерів.

  Державні портали закупівель також є хорошим джерелом отримання інформації про рішення, які впроваджені у замовника, у тому числі про системи захисту інформації.
- *Соціальна інженерія*. Використовуючи соціальну інженерію, можна добитися значного успіху в отриманні закритої інформації

компанії: наприклад у разі телефонного дзвінка порушник може представитися від імені працівника інформаційної служби, поставити правильні питання або попросити виконати потрібну команду на комп'ютері. Соціальні мережі добре допомагають визначити круг друзів і інтереси потрібної людини, така інформація може допомогти кіберпорушникам виробити правильну стратегію спілкування з майбутньою жертвою.

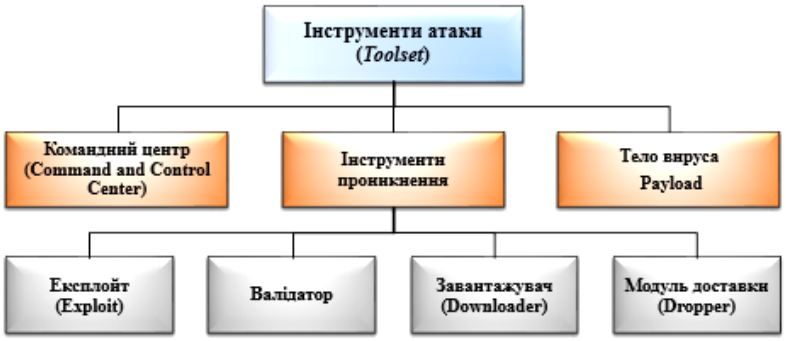
❒ *Розробка стратегії*. Стратегія є обов'язкової в реалізації успішної цільової атаки, вона враховує увесь план дій на всіх стадіях атаки:

- опис етапів атаки: проникнення, розвиток, досягнення цілей;
- методи соціальної інженерії, використовувані уразливості, обхід стандартних засобів безпеки;
- етапи розвитку атаки з урахуванням можливих позаштатних ситуацій;
- закріплення усередині, підвищення привілеїв, контроль над ключовими ресурсами;
- витягання даних, видалення слідів, деструктивні дії.

❒ *Створення стенду*. Спираючись на зібрану інформацію, група порушників приступає до створення стенду с ідентичними версіями експлуатованого ПЗ. Полігон, що дає можливість випробувати етапи проникнення вже на діючій моделі. Відпрацювати різну техніку прихованого впровадження і обходу стандартних засобів захисту інформації. По суті, стенд служить головним мостом між пасивною і активною фазами проникнення в інфраструктуру жертви. Створення подібного стенду обходиться недешево для хакерів. Витрати на виконання успішної цільової атаки зростають з кожним етапом.

❒ *Розробка набору інструментів*. Перед порушниками встає непростий вибір: їм важливо визначитися між фінансовими витратами на купівлю вже готових інструментів на ринку і трудовитратами і часом для створення власних. Тіньовий ринок пропонує досить широкий вибір різних інструментів, що значно скорочує час, за винятком унікальних випадків.

❒



Набір інструментів (*Toolset*) складається з трьох основних компонентів:

оцінки агресивності призначених для користувача програм, схема якої наведена на рис. 3.1.

Оцінка агресивності полягає в підрахунку показників агресивності на базі статистичного методу аналізу. При цьому послідовно розраховуються зараженість програми, захоплення ресурсів системою і наявність функцій саморепродукції.

Зараженість програми оцінюється на основі бази небезпечних операторів, яка формується експериментом на попередньому етапі і поповнюється в процесі функціонування експертної системи. Формування бази здійснюється шляхом статистичного аналізу сукупності руйнуючих програмних засобів (РПЗ) і сукупності потенційно надійних програмних засобів (тобто програмних засобів, що не порушують безпеки і цілісності обчислювальної системи).

Програма, яку аналізують, може бути представлена або у вигляді виконуваного файлу (* .exe) або у вигляді лістингу програми (* .asm, * .txt). При аналізі виконуваного файлу використовується «розумний» дизассемблер.

При практичній реалізації алгоритмів дизасемблерування виникають такі проблеми:

відновлення символічних імен. При компіляції програми всі імена замінюються фізичними адресами, і в скомпільованому машинному коді не залишається інформації про символічні імена;

відмінності команд і даних. У скомпільованій програмі машинний код і дані відрізняються один від одного тільки в контексті використання;

визначення кордону машинної команди. Команди машинного коду слідують одна за одною поспіль, без будь-яких розмежувань. Якщо дизасемблер неправильно визначає межу команди, він неправильно відновлює цю команду і кілька команд, що виконуються далі.

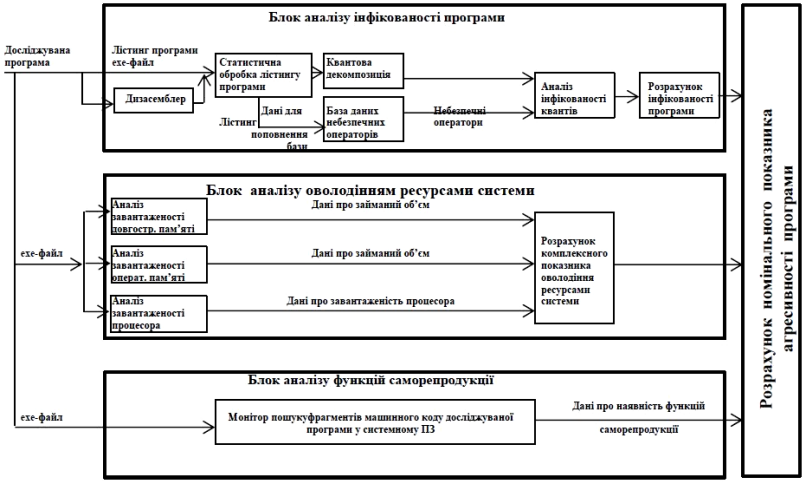


Рис. 3.1. Схема програмної реалізації підтримки експертної системи оцінки агресивності програмних засобів

- динамічний метод;
- статистичний метод.

Метод експериментів. Цей метод полягає в проведенні з досліджуваної програмою багаторазових експериментів і порівняльному аналізі отриманих результатів. Досліджувана програма розглядається як «чорний ящик», для якого відомі вхідні і вихідні дані, але невідомий внутрішній устрій. Завдання аналітиків полягає в тому, щоб, підбираючи вхідні дані, відновити алгоритми функціонування «чорного ящика».

Ефективність методу експериментів слабо залежить від програмної реалізації системи захисту (на неї, зокрема, не впливає застосування засобів захисту ПЗ від випромінювання) і визначається в першу чергу складністю аналізованих алгоритмів. Даний метод ефективний при аналізі програм, що реалізують відносно прості алгоритми.

Метод експериментів рідко застосовується в чистому вигляді. Найчастіше він служить доповненням до динамічного або статистичного методу. Аналіз програми динамічним методом розбивається на наступні етапи:

пошук шляху до досліджуваних функцій програми;

пошук досліджуваних функцій програми на обраному шляху;

аналіз знайдених функцій програми.

Після виконання цих трьох етапів висувається гіпотеза про суть досліджуваних алгоритмів. Цю гіпотезу потрібно обов'язково перевірити. Для цього можна, наприклад, запрограмувати досліджуваний алгоритм на мові високого рівня, вставити в тестову програму, запустити і подивитися, чи збігається її поведінка в частині аналізованих алгоритмів з поведінкою досліджуваної програми.

Як правило, на етапі перевірки отриманих результатів виявляються помилки в розумінні аналізованого алгоритму. В середньому, для повного відновлення алгоритму роботи програми потрібно три-чотири ітерації аналізу/перевірки, що вимагає значних затрат часу і інтелектуальних витрат.

Статистичний метод полягає у відновленні алгоритму за допомогою аналізу програмного забезпечення, наявного в розпорядженні. Файли програми для виконання зазвичай складаються із заголовка, в якому міститься необхідна для роботи допоміжна інформація, і послідовності виконуваних команд, записаних в машинних кодах команд процесора. У файлах програмного забезпечення збережені всі дані, необхідні для відновлення алгоритму. Завдання полягає лише в тому, щоб знайти відповідні ділянки програми і перевести їх на мову, зрозумілу аналітику. В даний час є цілий ряд програм аналізу, які полегшують переклад. Основою такого перекладу є програми дизасемблерування (дизасемблери), які перетворюють послідовність машинних кодів в лістинг, близький до початкового тексту програми на мові Асемблер. Ефективні програми перекладу на більш зрозумілі мови високого рівня, на жаль, відсутні.

З огляду на вищесказане, слід зробити висновок, що для автоматизації аналізу агресивності призначених для користувача програм метод експериментів і динамічний метод не підходять, тому за основу необхідно брати статистичний метод.

Одним з можливих способів реалізації статистичного методу є запропонована (як приклад) програмна реалізація підтримки експертної системи

1. Командний центр (*Command and Control Center*, *C&C*) забезпечує передачу команд підконтрольним шкідливим модулям, з яких вони збирають результати роботи. Центром атаки є люди, що проводять атаку. Найчастіше центри розташовуються в Інтернеті у провайдерів, що надають послуги хостингу, колокації і оренди віртуальних машин.

2. Інструменти проникнення, що вирішують задачу «відкриття дверей» видаленого хоста, що атакується:

- *експлойт* (*Exploit*) – шкідливийкод, що використовує уразливості в ПЗ;
- *валідатор*– шкідливийкод, який застосовується у випадках первинного інфікування, здатний зібрати інформацію про хост, передати її C&C для подальшого прийняття рішення про розвиток атаки або повну її відміну на конкретній машині;
- *завантажувач* (*Downloader*) – модульдоставки Dropper; завантажувач украй часто використовується в атаках, побудованих на методах соціальної інженерії, відправляється вкладенням в поштових повідомленнях;
- *модуль доставки Dropper*– шкідливапрограма (як правило, троян), завданням якої є доставка основного вірусу **Payload** на заражену машину жертви, призначена для:
  - закріплення усередині зараженої машини, прихованого автозавантаження, інжектування процесів після перезавантаження машини;
  - інжектування в легітимний процес для закачування і активації вірусу **Payload** по шифрованому каналу або витягання і запуску зашифрованої копії вірусу Payload з диска.

Виконання коду протікає в інжектованому легітимному процесі з системними правами, така активність украй складно детектується стандартними засобами безпеки.

3. Тіло вірусу Payload. Основний шкідливий модуль в цільовій атаці, що завантажується на інфікований хост Dropper'ом, може складатися з декількох функціональних додаткових модулів, кожен з яких виконуватиме свою функцію:

- клавіатурний шпигун;
- запис екрану;
- видалений доступ;
- модуль поширення усередині інфраструктури;
- взаємодія з C&C і оновлення;
- шифрування;
- очищення слідів активності, самознищення;
- читання локальної пошти;
- пошук інформації на диску.

2. Друга фаза – проникнення– активна фаза цільової атаки, що використовує різну техніку соціальної інженерії і уразливостей нульового дня для первинного інфікування мети і проведення внутрішньої розвідки. Після закінчення розвідки і визначенні приналежності інфікованого хоста (сервер/робоча

станція) по команді порушника через центр управління може завантажуватися додатковий шкідливий код.

За сукупністю вживаної техніки ця фаза є для порушників однієї з найскладніших у виконанні і реалізації.

Приклади проникнення з реальних атак.

❒ **Атака Carbanak** відома як велике банківське пограбування. Сумарний збиток від неї склав 1 млрд. доларів.Перше зараження проводилося за допомогою фішингових листів, в яких знаходилися вкладення з іменами файлів «Відповідність ФЗ-115 від 24.06.2014г.doc», «Запит.doc», «Анкета.doc». Усередині кожного файлу був вбудований експлойт з набором наступних уразливостей, кожна з яких дозволяла виконувати довільний код:

- **CVE-2014-1761**– уразливість в Microsoft Office;
- **CVE-2013-3906**– уразливість в Microsoft Office компонента Microsoft Graphics;
- **CVE-2012-0158**– уразливість ActiveX.

❒ **Duqu 2.0** - компанія кібершпигунства, створена для збору конфіденційних даних і держтаємниці різних держав.Кіберзлочинці також комбінували e- mail розсилки з фішингом через розміщену в листах URL, яка вела на інфікований сайт. Для проникнення в мережу використовувалася уразливість нульового дня з інжектуванням **svchost** процесу і ще одна уразливість нульового дня в **Kerberos** для отримання прав доступу адміністратора домена.

Основні технічні засоби проникнення:

❒ **Експлойт** (*Exploit*) – шкідливий код, що використовує уразливості в ПЗ. Це основний інструмент проникнення, засобами доставки якого є електронна пошта, компрометовані веб-сайти і USB-пристрої.

Проникнувши завдяки уразливості на цільовий корпоративний комп'ютер, експлойт запускає засіб доставки, тип якого залежить від дизайну атаки: це можуть бути валідатор, завантажувач або Dropper.

❒ **Валідатор** – збирач інформації із зараженого хоста, виконує фільтрацію інформації про облікові записи користувачів, встановлене ПЗ, активні процеси і засоби захисту. Передає шифровані дані в центр управління. Залежно від отриманої інформації хакери приймають рішення про подальший розвиток атаки, вибравши відповідну команду:

- *завантаження Dropper*– приступитидо виконання цільової атаки;
- *самознищення*– увипадках, коли комп'ютер і дані на нім не представляють цінності для цільової атаки;
- *очікування*– рішеннявідкладається, режим «сну».

Маючи мінімальний розмір і функціонал, валідатор не несе в собі унікальної інформації про цільову атаку та її організаторів. У випадку якщо він перехоплюється засобами захисту, це не створює для порушників загрози витоку методів і засобів, що плануються до застосування. Завдяки таким якостям може застосовуватися у випадках, коли:

❒ наявність потенційних уразливостей;

❒ наявність діючого шкідливого ПЗ.

Таким чином сучасна система виявлення таргетированих атак повинна уміти:

❒ збирати інформацію про події на різних рівнях інфраструктури в режимі 24/7;

❒ швидко виявляти сліди цільової атаки;

❒ повідомляти про інциденти інформаційної безпеки;

❒ детально протоколювати виявлені інциденти;

❒ передавати події про інциденти в систему кореляції подій SIEM та інші рішення;

❒ отримувати статистику загроз через хмарну інфраструктуру;

❒ накопичувати історичну інформацію про інциденти.

3.4 Оцінка агресивності програмних засобів

Підхід до програмної реалізації оцінки агресивності програмних засобів

У міру розширення областей застосування обчислювальних систем, поряд із проблемами надійності та стійкості їх функціонування, великого значення набуває проблема безпеки циркулюючої в них інформації при впливі зовнішніх і внутрішніх руйнівних чинників.

Концепція безпеки програмного засобу в даний час включає перелік п'яти основних властивостей:

❒ надійності;

❒ захищеності;

❒ достовірності;

❒ уразливості;

❒ агресивності.

Агресивність характеризує ступінь потенційних руйнівних дій програмного засобу на об'єкти обчислювальної системи.

Основні показники агресивності:

❒ показник «зараженості» - показник, що характеризує потужність кінцевої безлічі агресивних програмних елементів (ділянок програми, що містять руйнуючі функції) в програмному засобі;

❒ показник саморепродукції - показник, що характеризує здатність програмного засобу до виробництва інших програмних засобів - своїх змінених копій (наприклад, програмні закладки або віруси);

❒ показник «захоплення» - показник ступеня захоплення системних ресурсів обчислювальної системи, в якій функціонує програмний засіб.

Для оцінки всіх перерахованих вище показників агресивності досліджуваної програми необхідно відновити алгоритм її функціонування.

В даний час сформувалися три підходи до відновлення алгоритмів, що реалізуються програмою:

❒ метод експериментів;

в майбутньому. Це дозволить накопичувати знання про можливі ланцюжки атак і необхідні реакції на них. Цей процес повинен виконуватися на постійній основі і використовувати згадані вище методи інтелектуальної обробки даних, безперервно розширюючи можливості системи. Найбільш технологічно просунуті рішення в області захисту від таргетованих атак неодмінно повинні мати такий функціонал, причому надавати механізми поповнення таких формалізованих знань із зовнішніх джерел.

Прогнозування

Усунення наслідків цільової атаки є набагато складнішим завданням, ніж своєчасне застосування застережливих заходів. Важливо ідентифікувати уразливі сегменти корпоративної мережі і можливі вектори загроз для оперативного усунення «проломів» в системі безпеки. Етап прогнозування допомагає впоратися з цим завданням і включає певний набір послуг.

1. Тест на проникнення (*Penetration Test*), в ході якого моделюється друга фаза таргетованої атаки «Проникнення», в якій застосовуються комбінована техніка обходу і методи соціальної інженерії. Завданням тесту є виявлення найуразливіших елементів інфраструктури компанії і виробітку рекомендацій по їх усуненню.

2. Оцінка рівня захищеності (*Security Assessment*)відбувається без застосування засобів експлуатації уразливостей. Аналітики безпеки дістають доступ до усієї інфраструктури в цілому і проводять аудит зсередини, не удаючись до моделювання атаки ззовні. Сервіс дозволяє виявити критичні місця в інфраструктурі, вказавши на можливі вектори загроз.

3. Своєчасна оцінка уразливостей (*Vulnerability Assessment*). Автоматизований процес сканування і кваліфікації уразливостей дозволяє оперативно вказати на знайдені критичні точки в програмному забезпеченні. Має вбудований механізм (*Patch Management*), що забезпечує своєчасне оновлення програмних продуктів.

Щоб отримати точну оцінку загроз для виконання кожного з цих тестів, рекомендується притягати висококваліфікованих аналітиків з інформаційній безпеці.

4. Аналітичний звіт про загрози інформаційної безпеки (*Threat Intelligence Report*). Окремо важливо позначити наявність спеціалізованого інструменту оцінки загального стану захищеності компанії. Таким інструментом є сервіс, що базується на використанні пасивних і напівпасивних методів розвідки на основі відкритих джерел (OSINT), які не викликають яких-небудь підозр. Робота здійснюється з боку, без взаємодії з внутрішньою інфраструктурою компанії.

Тривалість надання сервісу дорівнює одному кварталу, що дозволяє визначити:

- ❒ актуальні загрози;
- ❒ факти компрометації інформаційних систем;

- ризик виявлення стандартними засобами захисту великий, для виключення можливого витоку вживаної техніки в цільовій атаці;
- цільова поштова розсилка.

Засоби доставки: електронна пошта, скомпрометовані веб-сайти, в окремих випадках USB-пристрої.

❒ **Завантажувач** (*Downloader*) – інструмент, використовуваний в цілях швидкого зараження із застосуванням техніки фішинга через вкладення в листах або з фішингових веб-сайтів. При запуску викачує основний модуль **Payload** або **Dropper** залежно від цілей і планів порушників.

❒ **Dropper** – це троянська програма, яка здійснює доставку основного шкідливого модуля **Payload** на цільову машину з наступним закріпленням усередині операційної системи, як правило, це приховане автозавантаження:

- визначає активні процеси в операційній системі, вибираючи найбільш вигідний з точки зору привілеїв таінжектує власний код в код активного процесу безпосередньо в оперативній пам'яті, що дозволяє йому отримати усі рівні доступу до ресурсів операційної системи, не викликаючи підозр у засобів захисту;
- завантажує тіло основного модуля Payload;
- виконує часткове дешифрування і запуск основного модуля.

Засобами доставки можуть бути електронна пошта, скомпрометовані веб-сайти, в окремих випадках USB-пристрої, а також експлойт, валідатор.

❒ **Основний модуль Payload** – основниймодуль в цільовій атаці може мати саме різне озброєння, яке залежить від поставленої мети і завдання.

Тіло модуля містить *багаторівневе шифрування*, покликане захистити розробки і технології порушників і детектування атаки. При першому запуску Dropper дешифрує тільки ту частину кода, яка містить техніки перевірки, покликані забезпечити гарантований запуск модуля у відповідному для нього середовищі і не допустити запуску, якщо середовище не задовольняє вимогам. Серед непідходящих для модуля умов можна назвати:

- поведінковий аналіз в пісочниці (Sandbox);
- емуляторні техніка у випадках, коли антиемуляторні механізми не спрацьовують;
- наявність відладчика і будь-якого іншого засобу роботи вірусного аналітика;
- наявність засобів моніторингу системи і мережевого трафіку;
- наявність невідомого антивіруса, що не потрапляє під використовувану техніку обходу.

У разі гіпотетичної атаки на компанію «А» хакери використовуватимуть конкретний набір інструментів. Він може складатися з одного Dropper з основним модулем Payload. Увесь перелік інструментів, приведений вище, лише демонструє різноманітність технологій в арсеналі кіберзлочинців.

Обхід стандартних засобів захисту

На сьогодні стандартні рішення інформаційної безпеки мають велику кількість функцій, що забезпечують високий рівень по контролю і фільтрації даних. Цей факт сильно ускладнює роботу кіберзлочинців і змушує їх винаходити і використовувати різну техніку, що дозволяє обдурити або обійти захисні механізми:

- **обфускація коду** - заплутування коду на рівні алгоритму за допомогою спеціальних компіляторів для ускладнення його аналізу антивірусом;
- **багаторівневе шифрування** застосовується для приховання частини коду від детектувальних механізмів;
- **інжектування процесу** – техніка по динамічному впровадженню власного коду у чужій процес. Дозволяє використовувати усі привілеї легітимного процесу у своїх цілях, не звертаючи на себе увагу встановлених засобів захисту. Цей метод дозволяє обійти різні системи контролю безпеки, у тому числі контролю застосувань. Інжектування застосовується на рівні Windows API:
  - визначення дескриптора потрібного процесу;
  - створення нового потоку у віртуальному просторі процесу;
- **mimikatz** – інструмент перехоплення паролів відкритих сесій в Windows, реалізовуючий функціонал Windows Credential Editor. Здатний витягати автентифікаційні дані користувача Кожна цільова атака будується на підвищенні прав доступу і реалізується під правами суперкористувача;
- **руткит** – цей засіб використовується для обходу захисту, закріплення в зламаній системі і приховання слідів присутності. Для Unix-середовища пакет утиліт (який включає також сканер, сніфер, кейлогер) містить і троянські програми, які замінюють собою основні утиліти Unix. Для Windows пакет перехоплює і модифікує низькорівневі API-функції, дозволяючи маскувати свою присутність в системі (приховуючи процеси, файли на диску, ключі в реєстрі). Руткити встановлюють всистеми свої драйвери і служби (вони також є «невидимими»). Руткит також може бути використаний як засіб доставки, здатний вивантажити усе необхідне хакерові після зараження машини;
- **обхід емулятора** – антивірусний емулятор перевіряє виконуваний файл в ізольованому середовищі, аналізуючи логіку його роботи. Виявлення шкідливого коду відбувається сигнатурним або евристичним методом. Хакери використовують різні практики по зміні алгоритму коду, не дозволяючи емулятору визначити логіку виконання шкідливої програми;
- **обхід поведінкового аналізу** – метод детектування застосовує пісочницю в цілях виявлення загроз нульового дня. Оскільки час перевірки пісочницею обмежений її функціональними можливос-

- ❒ аналіз витікаючих мережевих з'єднань для виявлення можливих з'єднань з командними центрами кіберзлочинців;
- ❒ використання відкритих джерел (**OSINT**, *Open source intelligence* – розвідка на основі відкритих джерел);
- ❒ збір доказів;
- ❒ аналіз доказів і реконструкція інциденту (хронологія і логіка);
- ❒ аналіз шкідливого ПЗ, використаного в атаці (у разі його виявлення);
- ❒ виявлення імовірної компрометації інших систем в оточенні;
- ❒ надання рекомендацій по подальших кроках виправлення.

Сервіс надається великими виробниками інформаційної безпеки і консалтинговими компаніями. Дозволяє зняти або підтвердити підозри про наявність слідів активних елементів таргетованої атаки.

Реагування

Основна мета реагування полягає в реакції на інцидент ІБ, наслідуючи набір прийнятих процедур, спрямованих на мінімізацію збитку і усуненню наслідків.

Етапи реагування включають:

- ❒ ідентифікацію;
- ❒ заборона;
- ❒ лікування;
- ❒ відновлення;
- ❒ висновки і профілактика.

Через особливості і відмінності цілеспрямованих атак від звичайних загроз процес реагування у разі APT має свою специфіку. Треба прийняти факт, що якщо компанія є метою для атаки, то той, що рано чи пізно атакує проб'ється в інфраструктуру тим або іншим чином. Виходячи з цього, необхідно усвідомлювати, що якщо система безпеки не може гарантувати 100% ефективність превентивних заходів, то необхідно забезпечити 100% детектування атаки, причому на самому ранньому етапі розвитку. Виявлення і реагування на загрозу на ранньому етапі дозволить мінімізувати збиток.

Треба не лише виявити атаку, необхідно своєчасно і адекватно зреагувати на неї. У разі неправильних дій на етапі реагування можна зруйнувати цифрові докази, тим самим утруднивши подальший аналіз або навіть зробивши його неможливим. Також не можна дозволити порушнику, що атакує, виявити протидію завчасно - запідозривши це, він може також вичистити сліди атаки.

Після блокування атаки також украй важливо провести аналіз дій порушника, з'ясувати вектори проникнення і поширення. На цьому етапі важливо розуміти, що якщо в процесі аналізу будуть виявлені не усі компоненти і сліди компрометації, є ризик, що порушник зможе залишитися в системі, згодом змінити свою поведінку і ще довго продовжувати свою діяльність.

Тому важливо формалізувати усі отримані в результаті аналізу знання і закласти їх в систему у вигляді правил/політик для автоматичного реагування

1. Динамічний аналіз об'єктів (пісочниця) – технологія виявлення підозрілої поведінки об'єкту у віртуальному ізольованому середовищі. *Пісочниця*– це, по суті, набір актуальних віртуальних середовищ, контрольованих технологіями аналізу виконання.

Порушники навчилися обходити цю технологію, застосовуючи різну техніку обходу (*Sandbox Evasion*).

Приклад **Sandbox Evasion-технік**, коли об'єкт не виявить свою активність:

- ❒ розділення екрану не більше ніж 800×600. Звичайний користувач не використовуватиме такерозділення в роботі;
- ❒ наявність встановлених програм (певний інвентар). Наприклад, будь-які встановлені засоби, які свідчать про наявність віртуального середовища;
- ❒ відсутність дій при демонстрації діалогового вікна. Немає реакції, отже, за машиною ніхто не працює.

Тому важливою особливістю сучасної пісочниці є її здатність протистояти техніці обходу (*Anti-Evasion*). Приклад **Anti-Evasion-технік**:

- ❒ емуляція роботи користувача (руху мишею, робота на клавіатурі);
- ❒ розпізнавання діалогових вікон, автоматична дія;
- ❒ виконання прокрутки документу на другу сторінку (*Scroll*);
- ❒ налаштування оточення, максимально схожого на реального користувача.

2. Аналіз аномалій. Технологія грунтується на статистичному аналізі інформації, що враховує частоту подій та їх послідовність.

Каналами збору інформації є мережеві сенсори і сенсори робочих станцій. В процесі роботи технології формується поведінкова модель, відхилення від якої може бути ознакою шкідливої активності.

Приклад роботи аналізатора аномалій:

- ❒ нетипове сканування мережі, здійснене з робочої станції секретаря;
- ❒ використання програм видаленого доступу в неробочийчас або в певний час, що повторюється.
- ❒ завантаження в мережу великого об'єму даних.

Впровадження спеціалізованих систем виявлення таргетованої атаки дозволить бачити симптоматику атаки, а не набір розрізненої інформації, що підлягає тривалому аналізу інженерами безпеки.

3. Сервіс з виявлення таргетованої атаки. У разі підозри аномальної активності усередині інфраструктури може виявитися ефективним спеціалізований сервіс, мета якого – виявлення слідів таргетованої атаки і виробітку рекомендацій по їх усуненню.

Перелік робіт у рамках такого сервісу:

- ❒ аналіз ландшафту загроз застосовано до конкретної компанії;
- ❒ використання спеціалізованих засобів для виявлення слідів компрометації;

тями, хакери використовують уповільнювач, і виконуваний код «засинає» на деякий час, щоб запобігти виявленню.

Експлуатація уразливостей

Важливо відмітити факт присутності уразливостей в різному ПЗ. Таке трапляється внаслідок прорахунків проектування або допущених помилок розробниками ПЗ, адже програми пишуть люди. Уразливість експлуатується через впровадження коду у вже запущену ОС або програму – такимчином змінюється штатна логіка роботи ПЗ, що дозволяє порушникам виконувати функції, що не декларують, частенько з правами адміністратора.

Уразливості програмного забезпечення можна розділити на два типи:

- ❒ **відомі**–що мають стандартно класифікований **CVE** (*Common Vulnerabilities and Exposures*) опис і готові виправлення в оновленнях розробника (CVE – відкритабаза відомих уразливостей);
- ❒ **невідомі(уразливості нульового дня)** – неусунені і ще не виявлені розробниками і дослідниками загрози.

Приклади експлуатації уразливості в процесі проникнення в інфраструктуру:

- ❒ **переповнювання буфера** (*buffer overflow*) – можевикликати аварійне завершення або зависання програми (відмова в обслуговуванні). Окремі види переповнювань дозволяють порушникові завантажити і виконати довільний машинний код від імені програми і з правами облікового запису, під яким ця програма запущена. Наприклад, користувач з правами адміністратора на своєму комп'ютері може отримати лист з вкладеним PDF-документом, в який буде вшитий експлойт. При відкритті чи попередньомуперегляді прикладеного документу запуститься процес переповнювання буфера, що дозволить порушникові отримати права локального адміністратора на цьому комп'ютері;
- ❒ **USB-пристрої**у поєднанні з уразливістю і методом соціальної інженерії. Приклад зараження через підключені USB-пристрої надзвичайно простий в реалізації. Наприклад, відомі випадки, коли в офісі компанії (на парковці, біля входу, в ліфті) були розкидані інфіковані USB-флешки з документом під принадним для простого співробітника найменуванням (річний звіт, фінансовий план), в який був вшитий експлойт.Другий приклад ще простіший: порушник приходить на співбесіду в цільову компанію і просить секретаря роздрукувати резюме, яке він нібито забув, з його флеш-карти.
- ❒ **цільовий фішинг** (*Spear phising*) у поєднанні з соціальною інженерією. Абсолютно усі сучасні компанії використовують спеціалізовані засоби контролю електронної пошти. Наявність таких сервісів, як антиспам і антивірус, є обов'язковою умовою для роботи корпоративної пошти. Багато користувачів звикли довіряти кореспонденції, що входить, пройшла, як вони припускають, професійну перевірку спеціалізованими засобами контролю безпеки. На жаль, це не відноситься до соціальної інженерії, коли хакери цілеспрямовано готують листи для обходу фільтрів і антивіруса.

Так, секретар може отримати електронний лист з вкладенням нібито від свого керівництва або партнера (кіберзлочинці можуть підробити адресу відправника або просто зробити його схожим на потрібний e-mail, видозмінивши один символ). При відкритті або попередньому перегляді документу станеться інфікування будь-яким з описаних вище інструментів з витікаючими наслідками розвитку атаки.

Різновиди використання пошти як точки входу:

- підробка/імітація адреси відправника;
- небезпечне вкладення (різні документи і зображення з вшитим експлойтом);
- посилання на HTML-сторінкуіз заздалегідь розміщеним інструментом проникнення.

Комбінована техніка

Організатори цільової атаки використовують *комбінований підхід*, що включає різні технічні засоби для реалізації проникнення. Очевидний приклад – коли хакер робив розсилку по електронних адресах співробітників компанії, заздалегідь увійшовши до контакту з ними в соціальній мережі під вигаданим ім'ям і зібравши дані про них. Це комбінація методів соціальної інженерії з фішингом.

Експлойт може містити великий набір уразливостей і застосовувати їх послідовно. При цьомучастина інструментів по проникненню можуть поєднуватися з легальним ПЗ, що дозволяє мінімізувати виявлення атаки, оскільки довірені програми внесені до «білих списків» систем безпеки, наприклад:

- програми видаленого адміністрування;
- програми перемикання мов клавіатури, що мають можливість легітимно використовувати логирование клавіатури;
- мережеві сканери і т.д.

Інвентаризація мережі

Після виконання автоматичних ухилень від виявлення і системних тестів на відповідність операційному середовищу **Payload** активує основні функціональні модулі, встановлюючи закрите шифроване з'єднання з командними центрами і сигналізуючи про свій активний статус. На цій стадії порушники приступають до консольної роботи, використовуючи термінал підконтрольної машини. Їм дуже важливо швидко зорієнтуватися усередині, щоб зберегти свою присутність і закріпитися в мережі. Першочерговим по важливості є підняття рівня доступу до привілейованого, після чого відразу ж починається вивчення топології мережі. Для виконання цього завдання зазвичай застосовують вільно поширюване ПЗ, наприклад **Netscan**.

3. Третя фаза – поширення– фазазакріплення усередині інфраструктури переважно на ключові машини жертви. З урахуванням зібраних даних по топології мережі відбувається ручний відбір ключових робочих станцій і серверів. Вибрані машини порушники беруть під свій контроль і використовують під нові завдання. На цьому кроці хакери вже мають адміністративні права і

Результати аналізу потрапляють в звіт, який містить інформацію:

- ❒ тип ботнета – класифікація;
- ❒ IP -адреса командних центрів;
- ❒ географічний розподіл зразків ПЗ;
- ❒ тип атаки – інформаціяпро цілі і використовуваному ПЗ;
- ❒ зведення про використані алгоритми атаки (ін'єкції Web-коду);
- ❒ MD5 – хешишкідливого ПЗ.

Послуга з аналізу активності ботнет-мереж є механізмом *раннього виявлення*. Використовуючи її, компанія забезпечить себе експертною інформацією по плановані атаці на свої ресурси, що значно зміцнить захист в цілому.

Впровадження спеціалізованих систем виявлення таргетованих атак

Таргетована атака характеризується високою мірою індивідуальності і стійкості до традиційних засобів захисту. Для виявлення ознак зараження необхідно застосовувати рішення, що мають засоби збору інформації про події на різних рівнях інфраструктури: як на зовнішньому контурі (Web, E-mail, основний Gateway), так і на внутрішньому (кінцеві вузли, внутрішні комутаційні вузли і т.д.). Такі рішення відрізняються комплексним застосуванням різних технологій *динамічного детектування* і здатністю забезпечувати аналітичне зіставлення потоків інформації з різних джерел. Цей підхід робить можливим виявлення складних, іноді розтягнутих в часі і добре замаскованих шкідливих дій. Модульність забезпечує розподілений контроль над усіма можливими каналами вступу і поширення елементів цільових атак.

Розглянемо детальніше вживані технології в системах виявлення цільової атаки.

Звіт служить інструментом для фахівця безпеки, надаючи глибоке розуміння загрози і можливість для проактивних дій, спрямованих на попередження конкретної загрози.

Аналіз активності ботнет-мереж. В останні роки спостерігається експоненціальне зростання кількості ботнет-мереж. Окрім стандартних шляхів поширення **Malware**, драйвером бурхливого зростання виступили рішення класу «Інтернет речей» (*Internet of Things*, *IoT*), виробники яких практично не приділяють уваги безпеки. Для кіберзлочинців IoT є ласим шматком, оскільки в більшості своїй ці рішення мають стабільне високошвидкісне підключення до Інтернету і мають достатній запас продуктивності.

Чому варто звертати увагу на активність ботнет-мереж в розрізі профілактики таргетованої атаки?

Річ у тому, що атаки, організовані ботнет-мережами, часто застосовують для цілеспрямованого фішинга (розсилки підробних поштових листів з прикріпленим завантажувачем).

Також через ботнет-мережи часто здійснюють DDoS-атаки, які служать засобом відвернення уваги від чогось більш значущого, у тому числі розвитку таргетованої атаки. Такі атаки дістали назву *Smokescreen*–**димовазавіса**.

Приклади таргетованих атак з використанням DDoS для відвернення уваги:

❒ в 2011 р. кінокомпанія **Sony Pictures Entertainment** піддалася цілеспрямованій атаці, в ході якої було паралізовано більше 80% кінцевих вузлів внутрішньої мережі компанії і зашифровані великі масиви інформації. Операція супроводжувалася масштабною DDoS-атакой, якій відводилася відволікаюча роль. Порушивши статистику мережевої активності, вона дозволила кіберзлочинцям непомітно вивантажити більше 100 Тбайт закритої інформації;

❒ в 2015 р. таргетована атака на енергоресурси України привела до віялового відключення енергопідстанцій. Перед початком активної фази кіберзлочинці застосували масштабну DDoS-атаку на зовнішні Web-ресурси компанії, тим самим відвернули увагу інженерів безпеці, змусивши їх в терміновому порядку вирішувати нав'язану проблему.

Моніторинг ботнет-мереж. Виробники рішень ІБ пропонують послугу, що надає експертну інформацію по планованих атаках, доступну в двох варіантах:

1) у складі потоку даних (Data Feed), надаючи комплексну інформацію по усіх відомим ботнет-мережам зі всього світу;

2) детальний звіт, що інформує компанію перед початком спроби реалізації атаки на корпоративні ресурси.

Яким чином здійснюється моніторинг активності ботнет-мереж?

Для виконання цього завдання застосовуються *безпечні контейнери* (ізольовані системи), які піддаються інфікуванню в реальному часі. Після чого процеси роботи троянов детально розбираються і аналізуються, дозволяючи побачити усю карту комунікацій трояна з командними центрами і сусідніми зомбохостами.

усі їх дії по відношенню до систем безпеки абсолютно легальні. Використовуючи стандартні засоби видаленого доступу, вони вибирають найбільш зручні з точки зору їх завдань сервери і робочі станції.

❒ *Крок 1. Закріплення усередині інфраструктури*, тобто реалізація комплексу заходів, спрямованих на організацію гарантованого доступу в інфраструктуру жертви.

Наприклад, в компанії Duqu 2.0 порушники використовували підписаний компанією Foxconn сертифікат. Хакери створили клон бібліотеки DLL, яка була присутня на комп'ютері жертви і після модифікації стала виконувати роль завантажувача Payload. Це дозволяло завантажувати шкідливий модуль при включенні комп'ютера і вивантажувати його при виключенні. Тим самим порушники не залишали слідів на жорстких дисках заражених машин, але при цьому зберігали свою присутність усередині, поширюючи такий метод усередині інфраструктури. Для основної точки входу використовувався головний контролер домена компанії.

❒ *Крок 2. Поширення.* Значущим аспектом є наявність постійних активних точок входу, зазвичай для цього використовуються сервери з малим часом простою, добре відповідні для виконання одного з правил цільової атаки Persistent. На такому рівні для зараження досить підключитися у вибраній машині видаленим RDP-клієнтом і запустити шкідливий модуль, заздалегідь скопіювавши його одним кликом миші.

❒ *Крок 3. Оновлення.* Трапляється, коли певна функція відсутня в арсеналі вже задіяного в атаці основного модуля, наприклад, такою функцією може бути запис звуку із зовнішнього мікрофону. Можливість відновити модуль заздалегідь передбачена розробником атаки і може бути активована при необхідності.

❒ *Крок 4. Пошук ключової інформації і методів досягнення мети*. Виконання етапу може сильно варіюватися за часом, адже інформація може бути різною. Якщо метою порушників є, наприклад, фінансова інформація, сконцентрована в одній системі, то це сильно спрощує їм завдання. Але якщо метою є шпигунство і довгостроковий збір розрізнених даних, то і кількість пристроїв, що зберігають потрібну хакерам інформацію, істотно зростає, що впливає на терміни виявлення цілей і на тривалість етапу.

4. Четверта фаза – досягнення цілей – ключова фаза цільової атаки.

❒ *Крок 1. Виконання шкідливих дій*. На цьому етапі порушники вже можуть виконати будь-яку дію, спрямовану проти компанії, що атакується. Перерахуємо основні типи загроз:

- **розкрадання ключової інформації** – у комерції це цілий бізнес, заснований на конкуренції і великих грошах; у державних структурах це шпигунство, рідше отримання інформації, що містить конфіденційні дані, для наступного перепродажу;у фінансовому секторі це інформація про платіжні і білінгові системи, рахунках

великих клієнтів та інша фінансова інформація для проведення незаконних транзакцій.

Само розкрадання відбувається максимально непомітно для систем моніторингу компанії, маскуючи мережеву активність під роботу відомого інтернет-сервісу з найменуванням домена, що сильно нагадує реальний. Зазвичай це виглядає як активна шифрована сесія, де веб-адреса часто схожа на популярні мережеві ресурси (наприклад, поштові сервіси, пошукові системи або новинні сайти);

- **зміна даних** – наприклад в цільової атаки **Metel**, від якої постраждали сотні фінансових організацій, кіберзлочинці, використовуючи контроль над платіжною системою, змінювали доступний кредит на балансі кредитної карти, тим самим дозволяючи спільникові кілька разів обналичивать засоби з однієї і тієї ж карти.

  А у разі кіберпограблення **Carbanak** хакери, вивчивши роботу операціоністів, діяли від імені співробітників, використовуючи онлайн-банкинг для переказу коштів на підконтрольні кіберзлочинцям рахунки;
- **маніпуляції з бізнес-процесами і шантаж**. Наочний випадок стався з компанією Sony Pictures, яка піддалася таргетованій атаці в 2014 році. В результаті були викрадені тисячі файлів і документів, фінансових даних, а також до кіберзлочинців в руки потрапили фільми, підготовлювані до прокату. У компанії розповіли, що більшість її комп'ютерів вийшли з ладу, а на екранах робочих станцій відображувалася фраза «ми оволоділи вашими секретами». Усі дані на жорстких дисках робочих комп'ютерів були стерті, кіберзлочинці погрожували опублікувати інформацію, якщо компанія не підкорятиметься їх вимогам;
- **знищення даних** – інший приклад розвитку цільової атаки. В серпні 2012 р. близько 30 тис. персональних комп'ютерів, що належать найбільшої у світі нафтовидобувної компанії Saudi Aramco, було виведено з ладу. Кіберзлочинці переслідували дві цілі: перша – розкрадання закритої інформації, друга – повна зупинка бізнес-процесів компанії. В результаті атаки компанія була вимушена майже на місяць припинити свою операційну діяльність, відключивши філії від мережіІнтернет.

❒ *Крок 2. Приховання слідів*. Упродовж усієї цільової атаки порушники прагнуть маскувати свою присутність під легітимний процес, в крайніх випадках, коли це неможливо, хакери вручну очищають журнали подій. Як правило, велика частина активності протікає під адміністративним доступом, не викликаючи підозри.

❒ *Крок 3. Точка повернення*. На фінальному етапі атаки багато порушників прагнуть залишити усередині засіб, що дозволяє їм у разі потреби повето

тою передачі пакетів раз в 10 хвилин. І ці дані відразу ж застосовуються в наявних засобах захисту, наприклад SIEM.

Потік даних є невід'ємною частиною будь-якого **SOC** (*Security Operational Center* – ситуаційнийцентр управління безпекою).

Потік даних містити в собі:

❒ набор URL-адрес, відповідних найбільш шкідливим посиланням і Web-сайтам;

❒ IP-репутація – градаціяIP-адрес по рівню безпеки;

❒ набір файлових хешей, що охоплює шкідливі програми;

❒ активність ботнет-мереж (шкідливі об'єкти, командні центри).

Комбінація з потоку даних і SIEM дозволяє істотно підвищити якість роботи рішення в частині детектування, фактично наділяючи корелятор експертними знаннями про самі останні атаки і загрози, що поширюються по світу. Тим самим досягається зниження кількості помилкових спрацьовувань і надається щонайпотужніший імпульс детектувальним здібностям системи.

Приклад: на базі виявленого командного центру ботнет-мережі нижче ви бачите рядки, що містяться в отриманому JSON-пакеті:

❒ унікальний ідентифікатор запису – **«id»: «143348»**;

❒ посилання на домен центру управління **«mask»: «botnetccurl.com»**;

❒ тип запису (застосовується для зіставлення правил в кореляторі) **«type»: «1»**;

❒ перший раз, коли був помічений **«first_seen»: «08.04.2014 16:45»**;

❒ останній раз, коли був помічений **«last_seen»: «12.02.2015 13:56»**;

❒ популярність (наскільки поширений, найвища популярність 5) **«popularity»: «5»**;

❒ найменування загрози **«threat»: «CnC.Win32.Zbot»**.

Після того, як інформація з отриманого JSON-пакету за допомогою парсера буде додана в SIEM, будь-яке звернення на домен **«botnetccurl.com»** з корпоративної мережі розцінюватиметься як інцидент і спроба зв'язку з командним центром. Що тим самим дозволить виявити можливі інфіковані машини усередині компанії.

Звіти про APT.Окрім потоків даних, на ринку існує можливість отримувати деталізовані звіти. Деталізовані звіти в основному надають ті ж компанії, які пропонують підписку на потоки даних. Це виробники рішень ІБ і цілий ряд консалтингових компаній з власним SOC і командою аналітиків.

APT-звітмістить, як правило, детальний опис таргетованих атак, виявлених експертною групою. Як і у випадку з потоком даних, звіти надаються зазвичай у вигляді підписки, є глибоко деталізованими і складаються з декількох типів файлів:

❒ **IoC** (*Indicator of Compromise*)– індикаторкомпрометації, опис шкідливих об'єктів;

❒ **YARA**– набірдетектувальних правил;

❒ детальний звіт, що містить аналіз (фази розвитку) атаки.

❒ формування бази знань з інцидентів;

❒ надання звітності для аналітиків.

Приклади детектування атаки усередині інфраструктури за допомогою SIEM:

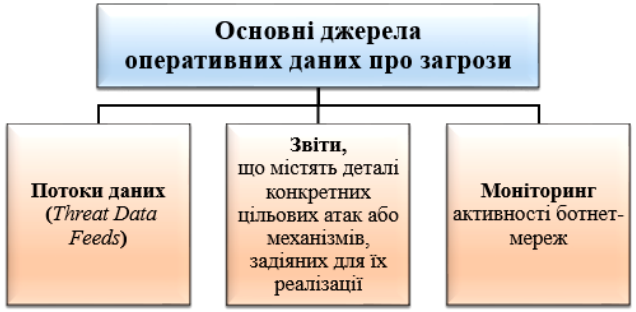
❒ обліковий запис співробітника «І» був використаний на його робочій машині в робочий час; в той же час пропускна система не мала записів про прихід співробітника на роботу (картка не активована), отже, корелятор, зіставивши обидві події, створить інцидент ІБ;

❒ три неправильні спроби входу в день з певного хоста упродовж певного періоду;

❒ множинні RDP-з'єднання з робочої машини, де раніше видалений доступ не використовувався (*RDP – Remote Desktop Protocol* – протокол видаленого робочого столу);

❒ сплеск мережевого трафіку між різними вузлами в неробочі години.

Threat Intelligence – даніпро актуальні загрози ІБ.Розуміння поточного ландшафту загроз і оперативні дані про актуальні атаки і шкідливі активності дозволяють компанії зміцнити захист корпоративних інформаційних систем.



Потоки даних (*Threat Data Feeds*) створюються великими компаніями, що працюють у сфері ІБ і мають у своєму арсеналі цілодобову службу з виявлення і аналізазагроз. Такі служби, як правило, складаються з аналітиків безпеки і автоматизованих комплексів детектування, що включають безліч новітніх технологій.

Зважаючи на безперервність процесу опису нових загроз кращим способом поширення експертної інформації є підписка, яка оформляється у вигляді сервісу. Вона дозволяє компанії оперативно отримувати нові порції інформації про загрози у форматі **JSON** (*JavaScript Object Notation*– простийформат обміну даними). В процесі доставки JSON-пакети даних легко трансформуються в будь-який необхідний вид під конкретне рішення завдяки використанню *парсера* (програми для прочитування і обробки текстових даних).

Фактично потік даних (*Data Feeds*), що складається з JSON-файлів, поповнює локальну експертну базу компанії з високою ефективністю: з часто-

рнутися назад в інфраструктуру. Таким засобом зазвичай є керований завантажувач, здатний по команді захитати виконуваний модуль.

3.3.3 Протидія таргетованим атакам

Комплексна стратегія протидії цільовим атакам включає чотири важливі елементи системи захисту:

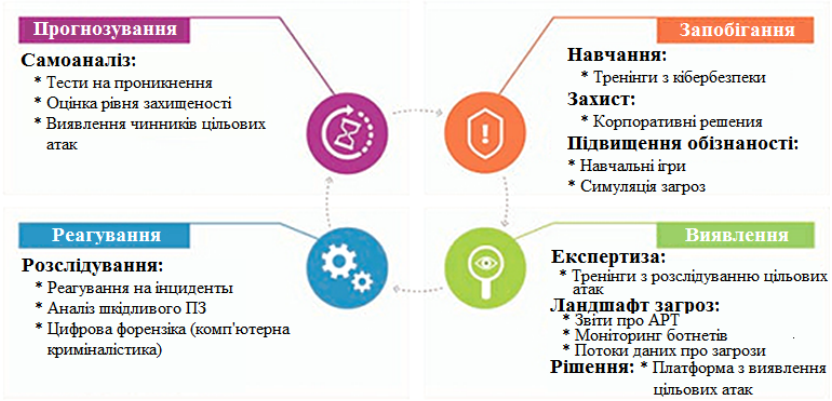
❒ *запобігання* – недопущенняпочатку і розвитку атаки;

❒ *виявлення* – виявлення слідів атаки, розпізнавання ознак, зв'язку усіх деталей атаки в єдину картину;

❒ *реагування* – у разі підтвердження факту атаки визначаються наслідки і кроки по їх *усуненню*;

❒ *прогнозування* – реалізаціяпроактивних заходів, що дозволяють істотно утруднити порушникам підготовку і проведення атаки.

АДАПТИВНА СТРАТЕГІЯ ЗАХИСТУ ВІД ЦІЛЬОВИХ АТАК



Запобігання таргетованій атаці

Головна мета– недопустити запуск якихось неконтрольованих процесів в корпоративній мережі.

Основні класи заходів:

1) використання набору технічних рішень, здатних перервати мережеву комунікацію або запуск якогось процесу в інфраструктурі;

2) навчання технічного персоналу і користувачів.

Технічні засоби.Йдеться про такі класичні засоби, як:

❒ захист кінцевих точок, включаючи антивірусні компоненти і контроль застосувань;

❒ міжмережеві екрани;

❒ системи запобігання вторгненням.

Основними технологіями детектування для рішень, що відносяться до превентивної групи, *єсигнатурний аналіз*, виконання правил для мережевих з'єднань, чорні та білі списки застосувань (**Black & Whitelisting**).

Добре збалансована система захисту, що використовує рішення від різних виробників, регулярно оновлювана, така, що проходить перевірку хоч би раз на рік – залишається важливою ділянкою оборони, що у ряді випадків дозволяє зупинити саму спробу атаки.

Наприклад, таргетована атака **Carbanak** була спрямована на фінансові установи, кумулятивний приблизний збиток від якої склало 1 млрд. доларів. Багато хто з уражених банків не мав сегментації внутрішньої мережі, мережа управління банкоматами була доступна з корпоративної мережі, чим скористалися кіберзлочинці. Використання мережевих екранів для сегментації мережі та міжсегментного контролю взаємодії дозволить забезпечити профілактику поширення цільової атаки усередині інфраструктури компанії.

Таргетована атака **Hellsing**, спрямована на шпигунство в урядових структурах, використовувала у своєму розвитку цілеспрямований фішинг – поширення листів з вкладенням, в якому знаходився архів, закритий паролем. Це дозволяло надійно обходити традиційні засоби захисту, засновані на перевірці вкладень. Усередині архіву знаходилися PDF-файли з вкладеним бекдором.

Наявність проактивного антиспам-фільтру у поєднанні з файловим антивірусом дозволить визначити спробу обходу стандартних засобів контролю (антивірус), тим самим ускладнивши організацію цільової шкідливої розсилки.

В організації ефективного захисту від цільової атаки необхідно застосовувати *технології динамічного аналізу*. Антивірус, хоч і відноситься до превентивної групи, але має ряд подібних технологій, що відносяться до категорії **Machine Learning** (*машинне навчання*).

Навчання в цілях підвищення грамотності в області інформаційної безпеки. Людський чинник і уразливості в ПЗ є головними складовими успіху в реалізації таргетованої атаки. Звичайний персонал компанії є ключем доступу порушників для входу в інфраструктуру. Мінімізація пропусків в знаннях з інформаційній безпеці дозволить співробітникам розпізнавати застосовувані до них методи соціальної інженерії і правильно реагувати на них. Персонал зобов'язаний знати, як соціальна інженерія управляє діями людини без застосування технічних засобів і для чого використовується цілеспрямований обман або інші дії, здатні ввести співробітника в оману.

Для того, щоб приносити реальну користь у відбитті атак, таке навчання повинне охоплювати найважливіші галузі знань, уявлення про яких повинен мати навіть рядовий співробітник:

- ❒ призначення антивіруса і контроль застосувань;
- ❒ повідомлення систем безпеки - як на них реагувати;
- ❒ розуміння проблеми витоку даних;
- ❒ риски при використанні мобільних пристроїв;
- ❒ загрози при роботі з електронною поштою і Інтернетом;
- ❒ соціальні мережі і риски роботи в них;
- ❒ методи соціальної інженерії;

- ❒ розвиток пильності;
- ❒ політика ІБ і як її можна порушити.

Отже, ефективне поєднання класичних превентивних рішень захисту разом з навченим основам кібербезпеки персоналом ускладнюють завдання тому, що атакує.

Детектування. Наступним найважливішим елементом системи захисту стає *детектування атаки*. Виявлення окремих ознак атаки або її компонентів ймовірніше при дотриманні наступних умов:

- ❒ тренінги та інші способи підвищення експертизи. Фахівці ІБ мають досить глибокі уявлення про природу і особливості таргетованих атак, постійно підвищують рівень своїх знань – в чому організація їх всіляко повинна підтримувати;
- ❒ SIEM (*Security Information and Event Management*–управління інформаційною безпекою і подіями безпеки) як автоматизація обробки подій безпеки;
- ❒ зовнішні джерела інформації про загрози (*Threat Intelligence*). Постачання актуальної інформації про погрози у вигляді фідов (потоків даних), списків IoC, звітів;
- ❒ системи з динамічним аналізом виконання – спеціалізовані засоби виявлення ознак таргетованої атаки;
- ❒ сервіси з виявлення атак з проведенням регулярних перевірок.

Забезпечення експертизи.Першою умовою є професійне навчання розслідуванню цільових атак – спеціалізований курс, що дозволяє фахівцям безпеки компанії ефективно виконувати подібні завдання, використовувати потрібні інструменти, виставляти пріоритети і збирати докази, проводити аналітичну роботу.

Курс зачіпає наступні аспекти:

- ❒ що таке інциденти інформаційної безпеки і їх категоризація;
- ❒ як проводити збір свідчень інциденту;
- ❒ вивчення прикладної науки –«криміналістика комп'ютерних злочинів (*Digital Forensics*)»;
- ❒ як правильно застосовувати спеціалізовані інструменти.

Автоматизація обробки подій безпеки.Розвиток ІБ підштовхнув компанії до автоматизації процесу збору, нормалізації, зберігання і обробки подій, що отримуються з журналів різних IT-систем. Це, у свою чергу, вплинуло на появу нового класу систем з консолідації і зберігання журналів подій – менеджментподій. Дані з логів прямують в єдину систему SIEM –систему управління подіями інформаційної безпеки, яка покликана вирішувати наступні завдання:

- ❒ збір, об'єднання, зберігання подій, що отримуються від різних джерел;
- ❒ оперативне виявлення порушень політик ІБ;
- ❒ автоматичне оповіщення і управління інцидентами;