**ЛЕКЦІЯ 3. Тема.** Поняття інформаційної загрози. Основні загрози та канали витоку інформації.

**План викладу матеріалу теми**

1. Поняття інформаційної загрози.
2. Основні загрози та канали витоку інформації.

## 1. Поняття інформаційної загрози.

*Несприятливий вплив* – вплив, що призводить до зменшення цінності інформаційних ресурсів.

*Загроза* – будь-які обставини чи події, що можуть спричинити порушення політики безпеки інформації та (або) нанесення збитку ІКС. Тобто загроза – це будь-який потенційно можливий несприятливий вплив.

*Атака* – це спроба реалізації загрози. Якщо атака є успішною (здійснено подолання засобів захисту), це називають *проникненням*. Наслідком успішної атаки є порушення безпеки інформації в системі, яке називають *компрометацією*.

Слід звернути увагу на те, що за комплексного підходу до захисту інформації ми маємо розглядати не лише впливи, спрямовані на інформаційні ресурси, але й будь-які впливи, що можуть завдати шкоди ІКС. Ми вже узагальнили це твердженням про необхідність захисту не самої інформації, а насамперед технології її оброблення.

*Уразливість системи* – нездатність системи протистояти реалізації певної загрози або ж сукупності загроз.

*Вади захисту* – сукупність причин, умов і обставин, наявність яких може призвести до порушення нормального функціонування системи або політики безпеки інформації. Здебільшого під вадами захисту розуміють особливості побудови програмних (а іноді й апаратних) засобів захисту, що за певних обставин спричиняють їхню нездатність протистояти загрозам і виконувати свої функції. Тобто вади захисту є окремим випадком уразливості системи.

*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).

*Модель [політики] безпеки* – абстрактний формалізований чи неформалізований опис політики безпеки. Модель безпеки використовують під час проектування системи для визначення механізмів і алгоритмів захисту, а також під час аналізу захищеності системи для перевірки й доведення коректності та достатності реалізованих механізмів.

*Модель загроз* – абстрактний формалізований чи неформалізований опис методів і засобів здійснення загроз.

*Модель порушника* – абстрактний формалізований чи неформалізований опис порушника. Моделі загроз і порушника є вихідною інформацією для

розроблення політики безпеки і проектування будь-яких систем захисту.

*Захищена комп'ютерна система* – комп'ютерна система, що здатна забезпечувати захист оброблюваної інформації від визначених загроз. Цей термін частіше вживають до обчислювальних систем або їхніх складових (програмних продуктів, окремих програмно-апаратних пристроїв). Іноді його застосовують до ІКС, але слід розуміти, що будь-яка сучасна ІКС має бути захищеною (навіть домашній комп'ютер із одним користувачем). Інакше її використання дуже швидко призведе до втрат інформації.

*Спостережність* – властивість ІКС, що дає змогу фіксувати діяльність користувачів і процесів, використання пасивних об'єктів, а також однозначно встановлювати ідентифікатори причетних до певних подій користувачів і процесів із метою запобігання порушенню політики безпеки і (або) забезпеченню відповідальності за певні дії. Це дуже важлива властивість обчислювальних систем та ІКС, яка досягається реалізацією засобів *реєстрації,* або *аудита*.

## 2. Основні загрози та канали витоку інформації.

***Основні загрози та канали витоку інформації:***

- Незаконне використання привілеїв.
- Атаки "салямі".
- Приховані канали.
- "Маскарад".
- "Збір сміття".
- "Зламування системи".
- Шкідливе програмне забезпечення.
- Утиліти схованого адміністрування.
- Intended-віруси.
- Конструктори вірусів.
- Поліморфні генератори.

***Несанкціонований доступ – НСД*** Це найпоширеніший вид комп'ютерних порушень. Він полягає в отримані користувачем доступу до об'єкта, на який у нього немає дозволу. За характером впливу НСД є активним впливом, йому може підлягати будь-який об'єкт ІКС. НСД може бути здійснений як стандартними, так і спеціально розробленими програмними засобами до об'єктів у будь-якому стані (при зберіганні, при передачі, при обробці інформації).

Методика реалізації НСД залежить від організації обробки інформації в даній ІКС, від організації політики безпеки, від можливостей встановлених засобів захисту. НСД стає можливим через непродуманий вибір засобів захисту, їх некоректне встановлення і налагодження, контроль роботи, недбале ставлення до захисту своїх власних даних.

Для реалізації НСД використовують два способи:

- можна подолати систему захисту, тобто шляхом різних впливів на неї припинити її дії стосовно себе або своїх програм (це складно, трудомістко, не завжди можливо, але ефектно);

- можна спостерігати за тим, що “погано лежить”, тобто які дані, що становлять інтерес для зловмисника, відкриті через недогляд або навмисно адміністратором. Такий НСД легко здійснити, але від нього і легше захиститись.

***2.1 Незаконне використання привілеїв***

Для даного способу атаки здебільшого використовується штатне програмне забезпечення (системне і прикладне), але яке функціонує в нештатному режимі. Майже будь-яка захищена система вміщує засоби, які можуть працювати з порушенням існуючої політики безпеки. У більшості випадків небезпечні засоби, які не повинні бути доступні звичайному користувачу, використовуються адміністраторами, системними програмістами та тими користувачами, які виконують деякі спеціалізовані функції. Для того, щоб зменшити ризик від застосування таких засобів, більшість систем захисту реалізує ці функції за допомогою набору привілеїв. Кожний користувач отримує свій набір привілеїв. Набори привілеїв кожного користувача є його атрибутами і зберігаються системою захисту. Отже, заповітна мрія зловмисника в такому випадку – оволодіти розширеним набором привілеїв.

Незаконне захоплення привілеїв можливе або за наявності помилок у самій системі захисту, або у випадку недобросовісного керування ІКС взагалі і системою привілеїв зокрема.

***2.2 Атаки “салямі”***

Атаки такого виду можливі в системах, де, наприклад, обробляються грошові рахунки для банків. Принцип атак “салямі” побудований на тому факті, що при обробці рахунків використовуються цілі одиниці (гривні, рублі, центи, копійки), а при нарахуванні процентів майже завжди виходять дробові числа.

Наприклад, 6.5% річних від суми 102.87 коп. за 31 день становлять 0.5495726 грн. Будь-яка банківська система заокруглить цю суму до 0.55 грн. Якщо робітник банку має доступ до банківських рахунків або до програм їх обробки, то він може округлити цю суму в інший бік – 0.54 грн., а різницю в 1 коп. скидати на свій рахунок. Власник рахунку ніколи не помітить цієї помилки або спише її на похибки обробки інформації. Таким чином, зловмисник при обробці за день 10000 рахунків матиме 100 грн., або більше 30000 грн. за рік.

Назва ж “салямі” пішла від ковбаси з такою назвою, яка виготовляється з різних сортів м’яса. Таким саме чином рахунок зловмисника поповнюється за рахунок різних вкладників.

Отже, атаки даного типу переважають у великих банках та інших фінансових організаціях. Причинами цих атак є:

- похибки обчислень, які дозволяють по-різному інтерпретувати правила округлення чисел;
- великі обсяги обчислень, які необхідно виконувати при обробці рахунків.

Атаки “салямі” досить важко розпізнаються (якщо тільки на рахунку зловмисника не накопичується величезна сума, яка може привернути увагу).

Запобігти цим атакам можна лише забезпечивши цілісність і коректність прикладних програм, розмежуванням доступу користувачів, постійним контролем рахунків на предмет їх змінювання.

### *2.3 Приховані канали*

Приховані канали – це шляхи передачі інформації між процесами системи, які порушують політику безпеки. Користувач може не мати дозволу на обробку даних, які його цікавлять, але він шукає обхідні шляхи. Оскільки будь-яка дія в системі викликає зміни стану інших складових системи, то за умови спостережливості і знання цих зв'язків можна відновити першопричину події хоча б частково.

Реалізовані "приховані канали" можуть бути різними шляхами, наприклад, за допомогою закладання "троянських коней".

Наприклад, програміст банку не завжди має доступ до імен і балансів депозитних рахунків, а програміст системи не має доступу до пропозицій про купівлю і продаж. Але при створенні таких систем він може передбачити спосіб отримання таких відомостей. В цьому разі програма встановлює таємно канал зв'язку з цим програмістом і повідомляє йому необхідні дані.

Прикладом активізації "прихованих каналів" може бути кінцевий звіт, в якому замість одного слова буде використовуватись інше. "Прихованим каналом" може стати число пропусків між двома словами, або значення третьої цифри після коми в якомусь виразі, на який ніхто не зверне увагу. "Прихованим каналом" може стати і інформація про присутність або відсутність якогось набору даних, його розміру, дати створення і модифікації тощо. Отож, існує багато способів організації зв'язку між двома процесами. Більше того, багато ОС мають у своєму розпорядженні такі засоби, оскільки вони полегшують роботу користувачів і програмістів. Тут важливо розрізняти недозволені і дозволені "приховані канали".

Атаки з використанням "прихованих каналів" у всіх випадках приводять до порушення конфіденційності інформації. За характером впливу вони є пасивними, для їх організації може бути використане як спеціальне, так і стандартне програмне забезпечення. Атаки здебільшого здійснюються програмним методом у пакетному режимі.

Характерними особливостями "прихованих каналів" є їх мала пропускна спроможність, оскільки по них можна передавати невелику кількість інформації, а також великі труднощі в їх організації і малі, як правило, збитки, яких вони завдають. Найчастіше ці збитки взагалі бувають непомітними, тому спеціальний захист проти "прихованих каналів" здійснюється дуже рідко.

### *2.4 "Маскарад"*

Під "маскарадом" (симуляція, моделювання), розуміється виконання яких-небудь дій одним користувачем ІКС від імені іншого, тобто права і привілеї одного користувача ІКС присвоюються іншим з метою доступу до наборів даних першого і використання його привілеїв.

Приклади "маскараду":

а) вхід в систему під ім'ям і паролем іншого користувача (при цьому система захисту не зможе розпізнати це порушення). В цьому випадку "маскараду" передує зламування системи або перехоплення паролю;

б) привласнення імені іншого користувача в процесі роботи за допомогою засобів ОС (деякі ОС дозволяють змінювати ідентифікатор користувача в процесі роботи) або за допомогою спеціальної програми, яка у визначеному місці змінює певні дані, в результаті чого користувач одержить інше ім'я;

в) передача повідомлень у мережі від імені іншого користувача. Особливо небезпечно, якщо це стосується керуючих повідомлень, які можуть змінити конфігурацію мережі, або повідомлень, які пов'язані з виконанням привілейованих операцій.

Дуже небезпечний "маскарад" у банківських системах електронних платежів, де невірна ідентифікація клієнта може призвести до великих втрат. Особливо це стосується платежів за допомогою електронних карток. Сам по собі метод ідентифікації за допомогою персонального ідентифікатора (Personal Identification Number (PIN)) досить надійний, а порушення можуть виникнути внаслідок неправильного його використання.

Для запобігання "маскараду" необхідно:

- використовувати надійні методи ідентифікації і аутентифікації;
- блокування спроб зламу системи;
- контроль входу в систему;
- фіксація всіх подій, які можуть свідчити про "маскарад" з метою їх подальшого аналізу;
- відмовлятись від програмних продуктів, які містять помилки і можуть призвести до "маскараду".

***2.5 "Збір сміття"***

Після закінчення роботи по обробці інформації частина даних може залишитися в оперативній пам'яті, на дисках, магнітних стрічках та інших носіях і зберігатися там до перезаписування або знищення. Прочитати прямим звичайним способом їх важко, але при використанні спеціальних програм і обладнання це все ж таки можна зробити. Такий процес і називається "збором сміття".

Для захисту від "збору сміття" використовуються спеціальні механізми, які можуть бути реалізовані в операційній системі і/або апаратурі комп'ютера чи в додаткових програмних (апаратних) засобах. Прикладами таких механізмів є стираючий зразок і мітка повноти.

*Стираючий зразок* – це послідовність бітів, яка записується на певне місце та стирає дані. Адміністратор може автоматично активізувати запис цієї послідовності при кожному звільнені ділянки пам'яті. При цьому стерті дані знищуються і ніхто не зможе вже їх відновити або прочитати (без спеціальної апаратури).

*Мітка повноти* – робить неможливим читання ділянок пам'яті, відведених для процесу записування, але не використаних ним. Верхня межа пам'яті, яка використ овується, і є міткою повноти. Цей спосіб використовується для захисту послідовних файлів виняткового доступу (результуючі файли редакторів, компіляторів, компоновщиків). Для індексних файлів і послідовних розділюваних файлів цей метод носить назву "стирання при розміщенні", тобто пам'ять очищується при видаленні її процесу.

### *2.6 “Зламування системи”*

Це навмисне проникнення в ІКС з несанкціонованими параметрами входу (іменем користувача і його паролем).

Причини можливостей зламування: помилки в керуванні системою захисту; помилки в проектуванні систем захисту; помилки в кодуванні алгоритмів захисту. “Зламування”, як правило, здійснюється в інтерактивному режимі.

Оскільки ім’я користувача зазвичай не є таємницею, то об’єктом полювання для зламщиків в більшості випадків є пароль. Способи розкриття паролю можуть бути різними: перебір можливих паролів, “маскарад” з використанням іншого користувача, захоплення привілеїв.

Основне навантаження на захист ІКС від зламування несе програма входу. Алгоритм вводу імені і пароля, їх шифрування, правила зберігання і заміни паролів не повинні мати помилок.

### *2.7 Шкідливе програмне забезпечення*

Цілком очевидно, що найбільш витончені загрози для ІКС являють собою програми, що досліджують їх вразливі місця.

Шкідливі програми – це програми, які призначені для того, щоб чинити шкоду і використовувати ресурси комп’ютера, вибраного в якості мішені. Вони часто маскуються в легальних програмах або імітуються під них. В деяких випадках вони розповсюджуються самі по собі, переходячи по електронній пошті від одного комп’ютера до іншого, або через заражені файли і диски.

Першу групу складають ті програми, що вимагають програм-носіїв. До них, в основному, відносяться фрагменти програм, що не можуть існувати незалежно від програм-носіїв, в ролі яких можуть виступати деякі програмні додатки, утиліти, системні програми. В цю групу входять:

- люки;
- логічні бомби;
- троянські коні;
- віруси.

У другу групу входять програми, що є незалежними. До них відносяться окремі незалежні програми, які можуть плануватися і запускатися операційною системою. До цієї групи належать:

- черв’яки,
- зомбі;
- утиліти прихованого адміністрування;
- програми-крадії паролів;
- “intended”-віруси;
- конструктори вірусів;
- поліморфік-генератори.

Крім того, небезпечні програми поділяються на такі, що:

- не відновлюють себе (не розмножуються). До них відносяться фрагменти програм, які повинні активізуватися під час певних дій головної програми;

- розмножуються – або фрагменти програм (віруси), або незалежні програми (черв'яки), що здатні під час запуску створювати одну або декілька копій самих себе. Ці копії пізніше також активізуються в цій самій або іншій операційній системі.

***Люк***

Люк – це прихована, недокументована точка входу в програмний модуль, яка дозволяє кожному, хто про неї знає, отримати доступ до програми в обхід звичайних процедур, призначених для забезпечення безпеки ІКС. Люк вставляється в програму в більшості випадків на етапі налагодження для полегшення роботи – даний модуль можна буде викликати в різних місцях, що дозволяє налагоджувати окремі його частини незалежно одна від одної. Крім того, люк може вставлятися на етапі розробки для подальшого зв'язку даного модуля з іншими модулями системи, але потім, внаслідок змінених умов, дана точка входу виявиться непотрібною.

Як правило, програміст розробляє програмний додаток, в який входить процедура реєстрації, або який треба довго настроювати, вводячи під час запуску багато значень. Можливо, розробник хоче надати програмі особливі привілеї або мати можливість запобігати процесу налаштування і аутентифікації, або програмісту треба мати в своєму розпорядженні надійний метод, що дозволяє активізувати програму в разі можливих збоїв.

Наявність люка дозволяє викликати програму нестандартним способом, що може серйозно відбитися на стані системи захисту (невідомо, як у такому випадку програма буде сприймати дані, середовище системи, тощо). Крім того, не завжди можна прогнозувати її поведінку.

Люки можуть з'явитися в програмах з таких причин:

- їх забули усунути (необміркований промах);
- для використання при подальшому налагодженні;
- для забезпечення підтримки готової програми;
- для реалізації таємного контролю доступу до даної програми після її встановлення (перший крок до навмисного проникнення в ІКС з використанням даної програми).

Запобігти люкам можна, провівши аналіз початкових текстів програм, міри безпеки повинні прийматися в основному ще на етапі розробки і оновлення програм.

***Логічні бомби***

Це один із самих ранішніх видів програм-загроз. Вони є попередниками вірусів і черв'яків.

Логічна бомба – це код, що поміщається в деяку легальну програму. Він влаштований таким чином, що при певних умовах "вибухає". Умовою для включення логічної бомби може бути наявність або відсутність деяких файлів, певний день тижня або певна дата, а також запуск додатку певним користувачем.

Ось приклад логічної бомби. В одному випадку логічна бомба перевіряла ідентифікаційний номер співробітника компанії, який був автором цієї бомби, і включалась, якщо цей ідентифікатор не фігурував у двох останніх нарахуваннях

заробітної плати. “Вибухаюча”, бомба могла змінити або видалити дані або файли, стати причиною зупинки машини або щось інше.

***Троянські коні***

До даної групи шкідливих програм відносять:

- програми-вандали,
- «дроппери» вірусів,
- «злі жарти»,
- деякі види програм-люків;
- деякі логічні бомби,
- програми вгадування паролів;
- програми прихованого адміністрування.

Останні чотири групи програм можуть і не існувати у вигляді «троянів», а бути цілком самостійними програмними продуктами, що також породжують шкідливі дії в операційній системі.

***Троянський кінь*** – різновид шкідницького програмного забезпечення, яке не здатне поширюватися самостійно (відтворювати себе) на відміну від вірусів та хробаків, тому розповсюджується людьми. Троянський кінь – це корисна, або така, що здається корисною, програма або процедура, в якій приховано код, здатний в разі спрацьовування виконати деяку небажану або шкідливу функцію.

Троянські коні можуть використовуватись для виконання тих функцій, які несанкціонований користувач не може виконати безпосередньо.

За характером троянський кінь належить до активних загроз, які реалізуються програмними засобами і працюють у пакетному режимі. Троянський кінь є загрозою для будь-якого об'єкта комп'ютерної системи, причому ця загроза може виражатися будь-яким із способів: безпосередній вплив на об'єкт атаки, вплив на систему дозволів, опосередкований вплив. Найнебезпечнішим є опосередкований вплив, за якого троянський кінь діє в рамках повноважень одного користувача, але в інтересах іншого користувача, особу якого встановити майже неможливо.

Небезпека троянського коня полягає в додатковому блоці команд, встановленому тим чи іншим способом у початкову нешкідливу програму, яка потім пропонується (подарунок, продаж, заміна) користувачам комп'ютерної системи. Цей блок команд може спрацювати при виконанні деякої умови (дати, часу і т. д., або по команді ззовні). Той, хто запускає таку програму, створює небезпеку як для себе і своїх файлів, так і для всієї комп'ютерної системи в цілому. Отже, у деяких випадках логічні бомби також можна віднести до троянських програм.

Найбільш небезпечні дії троянський кінь може виконувати, якщо користувач, який його запустив, має розширений набір привілеїв. У цьому випадку зловмисник, який склав і впровадив троянського коня, а сам цих привілеїв не має, може виконати несанкціоновані привілейовані функції чужими руками. Або, наприклад, зловмисника дуже цікавлять набори даних користувача, який запустив таку програму. Останній може навіть не мати розширеного набору привілеїв, – це не буде перешкодою для виконання несанкціонованих дій.

Наприклад, деякий користувач-зловмисник хоче отримати доступ до файлів іншого користувача. Він пише програму, яка під час запуску змінює права доступу до файлів користувача, який її викликав, таким чином, щоб ці файли могли прочитати інші користувачі. Далі, помістивши цю програму в загальний каталог і присвоївши їй ім'я, схоже на ім'я якоїсь корисної утиліти, автор програми якимось чином досягає того, щоб потрібний користувач запустив її. Прикладом такої програми може бути програма, яка нібито виводить лістинг файлів користувача в потрібному форматі.

Прикладом троянського коня, який важко виявити, може бути компілятор, змінений таким чином, щоб при компіляції вставляти в певні програми (наприклад, програми реєстрації в системі) додатковий код. За допомогою такого коду в програмі реєстрації можна створити люк, що дозволяє автору входити в систему за допомогою спеціального пароля. Такого тро-янського коня неможливо виявити в початковому тексті програми-реєстрації. Таким чином, і люки можна віднести до програм-троянів.

***Троянський кінь*** – одна з найнебезпечніших загроз безпеці операційних систем. Радикальним способом захисту від цієї загрози є створення замкнутого середовища виконання програм. Бажано також, щоб привілейовані і непривілейовані користувачі працювали з різними екземплярами прикладних програм, які мають зберігатися і захищатися індивідуально. При виконанні цих заходів імовірність впровадження подібних програм буде досить низькою.

У порівнянні з вірусами троянські коні не одержують широкого поширення по досить простих причинах – вони або знищують себе разом з іншими даними на диску, або демаскують свою присутність і знищуються постраждалим користувачем.

До категорії програм-троянів відносять також *програми-вандали*. Ці програми, як правило, імітують виконання якої-небудь корисної функції або маскуються під нову версію відомого програмного продукту. При цьому в якості побічного ефекту вони знищують файли, псують каталоги, форматують диски або виконують деякі інші деструктивні дії.

До троянських коней також можна віднести "дроппери" вірусів – заражені файли, код яких підправлений таким чином, що відомі версії антивірусів не визначають віруса у файлі. Наприклад, файл шифрується яким-небудь спеціальним чином чи упаковується рідко використовуваним архіватором, що не дозволяє антивірусу встановити факт зараження.

Слід зазначити також "злі жарти" (hoax). До них відносяться програми, що не заподіюють комп'ютеру якоїсь прямої шкоди, однак виводять повідомлення про те, що така шкода вже заподіяна, або буде заподіяна за певних умов, або попереджають користувача про неіснуючу небезпеку. До "злих жартів" відносяться, наприклад, програми, що "лякають" користувача повідомленнями про форматування диска (хоча самого форматування насправді не відбувається), детектують віруси в незаражених файлах (так робить відома програма ANTITIME), виводять дивні вірусоподібні повідомлення і т.д. - у залежності від почуття гумору автора.

До такої ж категорії "злих жартів" можна віднести також свідомо помилкові повідомлення про нові супер-віруси. Такі повідомлення періодично з'являються в електронних конференціях і звичайно викликають паніку серед користувачів.

***Віруси.*** Вірус – це програма. яка може заражати інші програми, змінюючи їх (копіює програму-вірус в програму, яка, в свою чергу, може заразити інші програми).

Біологічно віруси являють собою маленькі уламки генетичного коду (ДНК або РНК), які можуть переймати структуру живих клітинок і хитрістю залучити їх до виробництва тисяч точних копій початкового вірусу. Подібно цьому комп'ютерний вірус містить в собі рецепт того, як точно відтворити самого себе. Попавши в середовище комп'ютера, типовий вірус тимчасово бере на себе керування ОС і потім, при контакті зараженого комп'ютера з незараженими програмами, вірус упроваджує в ці програми свою копію. А далі він розповсюджується таким чином через магнітні носії, через мережу.

Вірус може робити те, що робить звичайна програма. Єдина відмінність полягає в тому. що він прикріплюється до іншої програми і приховано виконується під час роботи програми-хазяїна.

За час свого існування типовий вірус проходить 4 стадії:

- фаза спокою. Вірус не діє, а чекає події, яка його активізує. Такою подією може бути настання певної дати, наявність іншого файла або перевищення певного об'єму диска. Але не всі віруси притримуються цієї стратегії;
- фаза розмноження. Вірус розміщує свою копію в інші програми або в певні системні області на диску. Потім кожна заражена програмам містить клон вірусу, який також коли-небудь почне розмножуватись;
- фаза запуску. Вірус активізується для отримання можливості виконувати функції, для яких його створено. Як і вихід з фази спокою, перехід в фазу запуску може бути спровокований різними системними подіями (у тому числі – перевищення деякої припустимої кількості нових копій вірусу);
- фаза виконання. Вірус виконує свої функції. Ці функції можуть бути безпечними (виведення на екран повідомлення) або заподіювати шкоду (видаляти файли з програмами і даними).

Більшість вірусів робить свою справу, пристосовуючись до ОС, в деяких випадках – до певної апаратної платформи., тобто використовують особливості і слабкості операційних систем.

***Черв'яки.*** Черв'як – це програма, яка розповсюджується через мережу і не залишає своєї копії на магнітному носії. Черв'як використовує механізм підтримки мережі для визначення вузла, який може бути заражений. Потім за допомогою тих самих механізмів передає своє тіло на цей вузол й або активізується, або чекає для цього певних сприятливих умов.

Мережні програми-черв'яки використовують мережні з'єднання, щоб переходити з однієї системи в іншу. Одноразово активізувавшись в системі,

черв’як може вести себе як комп’ютерний вірус, породжувати троянських коней, виконувати інші руйнівні або деструктивні дії.

Для свого самовідтворення черв’як використовує деякий транспортний засіб:

- електронну пошту – черв’як розсилає свою копію іншим системам;
- можливості віддаленого запуску програм – черв’як запускає свою копію на іншій системі;
- можливості віддаленої реєстрації – черв’як входить у віддалену систему під виглядом користувача, а потім за допомогою стандартних команд копіює себе із однієї системи в іншу.

Перед тим, як копіювати себе на якусь систему, мережний черв’як може спробувати визначити, чи інфікована ця система. Крім того, в багатозадачній системі він може маскуватися, присвоюючи собі імена системних процесів або якісь інші, які важко помітити системному адміністратору.

Найбільш відомим представником цього класу є вірус Morpica (або, вірніше, "черв'як Моріса”), який вразив мережу Internet у 1988 році. Найсприятливішим середовищем для розповсюдження черв'яка є мережа, всі користувачі якої вважаються товаришами і довіряють один одному. Відсутність захисних механізмів якнайкраще сприяє вразливості мережі.

Найкращий спосіб захисту від черв'яка – вжиття заходів запобігання несанкціонованому доступу до мережі.

***Зомбі.*** Зомбі – це програма, яка приховано під’єднується до інших підключених в Інтернет комп’ютерів, а потім використовує цей комп’ютер для запуску атак, що ускладнює відстеження шляхів до розробника програми-зомбі.

Зомбі використовують при атаках з відмовою в обслуговуванні, які зазвичай направляють проти Web-вузлів. Зомбі розповсюджуються на сотні комп’ютерів, що належать не підозрюючим нічого третім особам, а потім використовуються для ураження вибраного в якості мішені Web-вузла за допомогою сильно збільшеного мережного трафіка.

***"Жадібні" програми*** . "Жадібні" програми – це програми, що намагаються монополізувати який-небудь ресурс, не даючи іншим програмам можливості використовувати його. Доступ таких програм до ресурсів системи призводить до порушення її доступності для інших програм. Безумовно, така атака буде активним втручанням у роботу системи. Безпосередній атаці в більшості випадків піддаються об'єкти системи: процесор, оперативна пам'ять, пристрої введення-виведення.

Багато комп'ютерів, особливо в дослідницьких центрах, мають фонові програми, які виконуються з низьким пріоритетом. Вони проводять великий обсяг обчислень, а результати їхньої роботи потрібні не так вже часто. Але при підвищенні пріоритету така програма може блокувати решту програм. Ось чому вона є "жадібною".

"Тупикова" ситуація виникає тоді, коли "жадібна" програма нескінченна (наприклад, виконує явно нескінченний цикл). Але в багатьох операційних системах існує можливість обмеження часу процесора, який використовується конкретною задачею. Це не стосується операцій, які виконуються залежно від

інших програм, наприклад операцій введення-виведення, що закінчуються асинхронно до основної програми, оскільки час їх виконання не входить у час роботи програми. Перехоплюючи асинхронне повідомлення про закінчення операції введення-виведення і посилаючи знову запит на нове введення-виведення, можна досягти нескінченності програми. Такі атаки називають також асинхронними.

Другий приклад "жадібної" програми – програма, яка захоплює дуже велику ділянку оперативної пам'яті. В оперативній пам'яті послідовно розміщуються, наприклад, дані, які над-ходять із зовнішнього носія. Врешті-решт пам'ять може бути сконцентрована в одній програмі, і виконання інших стане неможливим.

***Захоплювачі паролів***. Захоплювачі паролів – це спеціально при-значені програми для крадіжки паролів. Вони виводять на екран терміналу (один за одним): порожній екран, екран, який з'являється після катастрофи системи або сигналізує про закінчення сеансу роботи. При спробі входу імітується введення імені і пароля, які пересила-ються власнику програми-захоплювача, після чого виводиться повідомлення про помилку введення і управління повертається операційній системі. Користувач думає, що зробив поми-лку при наборі пароля, повторює вхід і отримує доступ до системи. Отже, в результаті таких дій його ім'я і пароль стають відомими власнику програми-захоплювача.

Перехоплення пароля може здійснюватися й іншим способом – за допомогою впливу на програму, яка керує входом користувачів у систему, та її наборів даних.

Захоплення пароля є активним, безпосереднім впливом на комп'ютерну систему в цілому. Для запобігання цій загрозі перед входом в систему необхідно впевнитися, що вводиться ім'я і пароль саме системної програми входу, а не якої-небудь іншої. Крім того, необхідно суворо дотримуватися правил використання паролів і роботи з операційною системою. Слід зауважити, що більшість порушень здійснюється не через хитромудрі атаки, а через елементарну необережність.

Не слід вимикати комп'ютер, доки не будуть закриті всі робочі програми. Необхідно постійно перевіряти повідомлення про дату і час останнього входу і кількість помилкових входів. Ці прості дії допоможуть уникнути захоплення пароля.

Крім описаних вище, існують і інші можливості компрометації паролів. Отже, слід дотримуватись правил, які рекомендуються для створення і використання паролів.

Не слід записувати команди, які містять пароль, у командні процедури, слід намагатись уникати явного повідомлення пароля при запитуванні доступу по мережі, оскільки ці ситуації можна простежити і захопити таким чином пароль. Не слід використовувати один і той самий пароль для доступу до різних вузлів. Рекомендується частіше змінювати пароль.

Дотримання правил використання паролів - необхідна умова надійного захисту.

### *2.8. Утиліти схованого адміністрування*

Цей вид шкідливого програмного забезпечення у деяких випадках також можна віднести до групи троянських коней. Вони по своїй суті є досить могутніми утилітами віддаленого адміністрування комп'ютерів у мережі.

За своєю функціональністю вони багато в чому нагадують різні системи адміністрування, розроблені і розповсюджені різними фірмами-розробниками програмних продуктів. Єдина особливість цих програм змушує класифікувати їх як шкідливі троянські програми – відсутність попередження про інсталяцію і запуск. Під час запуску троянська програма встановлює себе в системі і потім стежить за нею, при цьому користувачу не видається ніяких повідомлень про дії такого трояна в системі. Більш того, посилання на трояна може бути відсутнім у списку активних додатків. У результаті користувач цієї троянської програми може і не знати про її присутність у системі, у той час як його комп'ютер відкритий для віддаленого керування.

Будучи встановленими на комп'ютер, утиліти прихованого адміністрування дозволяють робити з комп'ютером усе, що в них заклав їх автор: приймати і відсилати файли, запускати і знищувати їх, виводити повідомлення, стирати інформацію, перевантажувати комп'ютер і т.д. У результаті ці трояни можуть бути використані для виявлення і передачі конфіденційної інформації, для запуску вірусів, знищення даних і т.п. У цьому випадку уражені комп'ютери виявляються відкритими для злочинних дій хакерів.

### *2.9. Intended-віруси*

До intended-вірусів (intended – навмисний) відносяться програми, які на перший погляд є стовідсотковими вірусами, але не здатні розмножуватися через помилки. Наприклад, вірус, що при зараженні "забуває" помістити в початок файлів команду передачі керування на код вірусу, або записує в неї невірну адресу свого коду, або неправильно встановлює адресу перехоплюваного переривання (що в переважній більшості випадків “завішує” комп'ютер).

До категорії intended-вірусів також відносяться віруси, що за приведеними вище причинами розмножуються тільки один раз – з авторської копії. Заразивши будь-який файл, вони втрачають здатність до подальшого розмноження.

З'являються intended-віруси найчастіше при невдалій перекомпіляції якого-небудь вже існуючого вірусу, через недостатнє знання мови програмування, через незнання технічних тонкощів операційної системи.

### *2.10. Конструктори вірусів*

Конструктор вірусів – це утиліта, призначена для виготовлення нових комп'ютерних вірусів. Відомі конструктори вірусів для DOS, Windows і макровірусів. Вони дозволяють генерувати вихідні тексти вірусів (ASM-файли), об'єктні модулі, і/чи безпосередньо заражені файли.

Деякі конструктори (VLC, NRLG) оздоблені стандартним віконним інтерфейсом, де за допомогою системи меню можна вибрати тип вірусу, об'єкти для зараження (COM і/чи EXE), наявність або відсутність самошифрування, протидії налагоджувачу, внутрішні текстові рядки, вибрати ефекти, що супроводжують роботу вірусу і т.п.

### *2.11. Поліморфні генератори*

Поліморфні генератори (або поліморфік-генератори), як і конструктори вірусів, не є вірусами в буквальному значенні цього слова, оскільки в їх алгоритми не закладаються функції розмноження, тобто відкриття, закриття і записування у файли, читання і записування секторів і т.д. Головною функцією подібного роду програм є шифрування тіла вірусу і генерація відповідного розшифровувача.

Звичайно поліморфні генератори поширюються авторами без обмежень у виді файла-архіву. Основним файлом в архіві будь-якого генератора є об'єктний модуль, що містить цей генератор. В усіх генераторах, що зустрічалися до цих пір, такий модуль містить зовнішню (external) функцію – виклик програми-генератора. У такий спосіб автору вірусу, якщо він бажає створити дійсний поліморфік-вирус, не приходиться "длубатися" над кодами власного за/розшифровувача. При бажанні він може підключити до свого вірусу будь-який відомий поліморфік-генератор і викликати його з кодів вірусу. Фізично це досягається в такий спосіб: об'єктний файл вірусу лінкується з об'єктним файлом генератора, а у вихідний текст вірусу перед командами його запису у файл вставляється виклик поліморфік-генератора, що створює коди розшифровувача і шифрує тіло вірусу.

## Контрольні запитання

1. Поняття інформаційної загрози. Дайте визначення наступним поняттям: «*Несприятливий вплив»,* «*Загроза»,* «*Атака»* , «*Уразливість системи»*, «*Вади захисту»* .
2. Чим на Вашу думку відрізняється «*Порушник*» від «*Зловмисника*»?
3. Модель політики безпеки це?
4. Чим на Вашу думку відрізняється «*Модель загроз» від* «*Модель порушника*»?
5. Захищена комп'ютерна система це?
6. Поясніть, що на Вашу думку є несанкціонованим доступом?
7. Основними загрозами та каналами витоку інформації є?
8. Шкідливі програми це?
9. Перелічіть ті шкідливі програми, що вимагають програм-носіїв.
10. Перелічіть ті шкідливі програми, що не вимагають програм-носіїв і є незалежними.
11. Троянський кінь це?
12. Дайте тлумачення терміну «комп'ютерний вірус». За час свого існування типовий вірус проходить які стадії?