**ВІННИЦЬКИЙ ДЕРЖАВНИЙ ПЕДАГОГІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ МИХАЙЛА КОЦЮБИНСЬКОГО**

**ІНСТИТУТ МАТЕМАТИКИ, ФІЗИКИ І ТЕХНОЛОГІЧНОЇ ОСВІТИ**

**КАФЕДРА ІННОВАЦІЙНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В ОСВІТІ**

**Н.М. Кириленко**

# ІНФОРМАЦІЙНА БЕЗПЕКА

**Навчально-методичний посібник**

**ВІННИЦЯ**

**ГЛОБУС-ПРЕС**

**2011**

ББК 32.973.018.2
УДК 004.9

Рецензенти:
В.М. Михалевич – доктор технічних наук, професор
(Вінницький національний технічний університет)
М.Ю.Кадемія – канд.пед.наук, доцент
(Вінницький державний педагогічний університет імені Михайла Коцюбинського)

**Кириленко Н.М.**
Інформаційна безпека. Навчально-методичний посібник. – Вінниця: – ГЛОБУС-ПРЕС, 2011. - 218с.

Рекомендовано до друку
Вченою радою Інституту математики, фізики і технологічної освіти (протокол № від 14 грудня 2011 р.)

Навчально-методичний посібник складається з шести лекцій та чотирьох практичних робіт. Для вирішення основних питань курсу розглядаються заходи законодавчого, адміністративного, процедурного і програмно-технічного рівнів. Матеріали навчально-методичного посібника можуть бути використані для успішного освоєння сучасного інформаційно-технічного базису, загальної структури інформаційної безпеки.

Рекомендовано для студентів Інституту математики, фізики і технологічної освіти.



**ISBN** 966-8300-18-1

## Передмова

У сучасному суспільстві у зв'язку з дедалі більшими потребами людини виникають проблеми інформаційного забезпечення усіх сфер її діяльності. Проблема інформатизації є найважливішою для сучасного суспільства. Однак вона породжує цілий ряд серйозних супутніх проблем. Однією з таких проблем є надійний захист інформації, який забезпечував би попередження перекручування або знищення інформації, унеможливлював би її зловмисне отримання та використання. Особливої гостроти ця проблема набуває у зв'язку з масовою комп'ютеризацією інформаційних процесів і, насамперед, у зв'язку з об'єднанням комп'ютерів в інформаційно-обчислювальні мережі.

Однією з найважливіших проблем забезпечення інформаційної безпеки безперечно слід вважати підготовку кваліфікованих фахівців з інформаційної безпеки. Нагальна потреба у підготовці фахівців із захисту інформації вимагає введення відповідних курсів у навчальних закладах України.

У навчальний план підготовки фахівців Інституту математики, фізики і технологічної освіти, діяльність яких пов'язана з інформаційними процесами, включається дисципліна «Інформаційна безпека». Це дисципліна, що узагальнює підготовку студентів з інформаційної безпеки обчислювальних систем та комп'ютерних мереж.

Навчально-методичний посібник складається із шести лекцій та чотирьох практичних робіт. Для вирішення основних питань курсу розглядаються заходи законодавчого, адміністративного, процедурного і програмно-технічного рівнів, розглядаються проблеми інформаційної безпеки. Крім цього, матеріали навчально-методичного посібника можуть бути використані для успішного освоєння сучасного інформаційно-технічного базису, загальної структури інформаційної безпеки.

Мета курсу – закласти методично правильні основи знань, необхідні майбутнім фахівцям-практикам.

## Лекція 1

# ПОНЯТТЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

### План лекції:

1.Поняття інформаційної безпеки.
2.Основні складові інформаційної безпеки.
3.Визначення та загальні властивості інформації.
4.Цінність та класифікація інформації.
5.Інформація як об’єкт власності.
6.Інформація як комерційна таємниця.

### 1. Поняття інформаційної безпеки

Перш ніж говорити про інформаційну безпеку необхідно з’ясувати, що таке інформація.

Поняття «інформація» сьогодні застосовується дуже широко і різнобічно. Важко знайти таку область знань, де б воно не використовувалося. Величезні інформаційні потоки буквально захлинають. Обсяг наукових знань за оцінкою фахівців, подвоюється кожні п'ять років.

**Інформація –** дані про людей, предмети, факти, події, явища і процеси незалежно від форми їхнього представлення.

Відомо, що інформація може мати різну форму, зокрема, дані, закладені в комп'ютерах, листи, пам'ятні записи, досьє, формули, креслення, діаграми, моделі продукції і прототипи, дисертації, судові документи тощо.

Як і всякий продукт, інформація має споживачів, що потребують її, і тому має певні споживчі якості, а також має і своїх власників або виробників.

Відповідно до різноманітності інформації, словосполучення "інформаційна безпека" в різних контекстах може мати різний сенс.

**Інформаційна безпека** – стан захищеності життєво важливих інтересів людини, суспільства і держави, при якому запобігається нанесення шкоди через: неповноту, невчасність та невірогідність інформації, що використовується; негативний

інформаційний вплив; негативні наслідки застосування інформаційних технологій; несанкціоноване розповсюдження, використання і порушення цілісності, конфіденційності та доступності інформації.

Спеціальне законодавство в області безпеки інформаційної діяльності представлено низкою законів. У їхньому складі особливе місце належить базовому Закону «Про інформацію, інформатизацію і захист інформації», що закладає основи правового означення всіх найважливіших компонентів інформаційної діяльності:

- інформації й інформаційних систем;
- суб'єктів - учасників інформаційних процесів;
- правовідносин виробників - споживачів інформаційної продукції;
- власників (власників, джерел) інформації - оброблювачів і споживачів на основі відносин власності при забезпеченні гарантій інтересів громадян і держави.

Під інформаційною безпекою (ІБ) ми розумітимемо захищеність інформації та інфраструктури, що її підтримує, від випадкових або навмисних дій природного або штучного характеру, які можуть завдати неприйнятного збитку суб'єктам інформаційних відносин, зокрема власникам і користувачам інформації та інфраструктури, що її підтримує.

Захист інформації – це комплекс заходів, направлених на забезпечення інформаційної безпеки.

Таким чином, правильний з методологічної точки зору підхід до проблем інформаційної безпеки починається з виявлення суб'єктів інформаційних відносин та інтересів цих суб'єктів, пов'язаних з використанням інформаційних систем (ІС). Загрози інформаційній безпеці – це зворотна сторона використання інформаційних технологій.

Тут необхідно зауважити, що трактування проблем, пов'язаних з інформаційною безпекою, для різних категорій суб'єктів може істотно різнитися. Для ілюстрації досить зіставити режимні державні організації і навчальні інститути. У першому випадку "хай краще все зламається, ніж ворог

дізнається хоч один секретний біт", у другому – "немає у нас жодних секретів, аби все працювало". Отже, інформаційна безпека не зводиться виключно до захисту від несанкціонованого доступу до інформації. Це принципово ширше поняття. Суб'єкт інформаційних відносин може постраждати (зазнати збитки та/або одержати моральний збиток) не тільки від несанкціонованого доступу, але й від поломки системи, що викликала перерву в роботі. Більш того, для багатьох відкритих організацій (наприклад, навчальних) захист від несанкціонованого доступу до інформації стоїть за важливістю не на першому місці.

Повертаючись до питань термінології, відзначимо, що термін "комп'ютерна безпека" (як еквівалент або замінник ІБ) представляється нам дуже вузьким. Комп'ютери – тільки одна із складових інформаційних систем. Хоча наша увага буде зосереджена в першу чергу на інформації, яка зберігається, обробляється і передається за допомогою комп'ютерів, її безпека визначається всією сукупністю складових. Найслабкішою ланкою, в переважній більшості випадків, виявляється людина.

Згідно визначення інформаційної безпеки, вона залежить не тільки від комп'ютерів, але і від інфраструктури, що її підтримує. Це системи електро-, водо- і теплопостачання, кондиціонери, засоби комунікацій і, звичайно, обслуговуючий персонал. Ця інфраструктура має самостійну цінність, але нас цікавитиме лише те, як вона впливає на виконання інформаційною системою своїх функцій.

Звернемо увагу, що у визначенні ІБ перед іменником "втрати" стоїть прикметник "неприйнятний". Очевидно, застрахуватися від всіх видів втрат неможливо, тим більше неможливо зробити це економічно доцільним способом, коли вартість захисних засобів і заходів не перевищує розмір очікуваних втрат. Отже, з чимось доводиться миритися і захищатися слід тільки від того, з чим змиритися ніяк не можна. Іноді таким неприпустимими витратами є нанесення шкоди здоров'ю людей або стану навколишнього середовища. Частіше за все, поріг неприйнятності має матеріальний (грошовий) вираз,

а метою захисту інформації стає зменшення розмірів втрат до допустимих значень.

## 2. Основні складові інформаційної безпеки

Інформаційна безпека – багатогранна, можна навіть сказати, багатовимірна область діяльності, в якій успіх може принести тільки систематичний, комплексний підхід.

Спектр інтересів суб'єктів, пов'язаних з використанням інформаційних систем, можна розділити на наступні категорії, що показані на рис.1: забезпечення доступності, цілісності і конфіденційності інформаційних ресурсів та інфраструктури, що її підтримує.

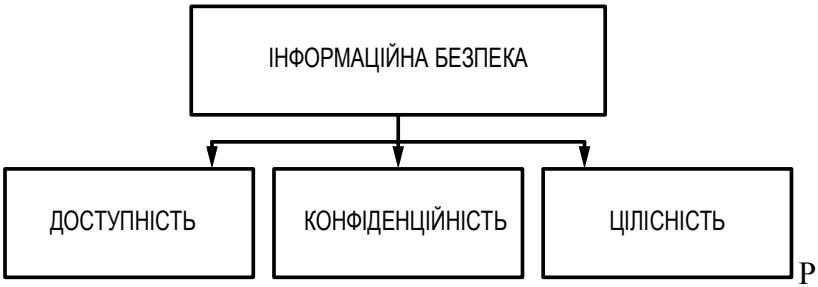



Рис.1. Основні складові інформаційної безпеки

Іноді в число основних складових ІБ включають захист від несанкціонованого копіювання інформації. На наш погляд, це дуже специфічний аспект з сумнівними шансами на успіх, тому ми не будемо його виділяти.

Пояснимо поняття доступності, цілісності і конфіденційності.

**Доступність** - це можливість за прийнятний час одержати необхідну інформаційну послугу. Інформаційні системи створюються для отримання певних інформаційних послуг. Якщо за тими або іншими причинам надати ці послуги користувачам стає неможливо, то це завдасть збитку всім суб'єктам інформаційних відносин. Тому, не протиставляючи доступність решті аспектів, ми виділяємо її як найважливіший елемент інформаційної безпеки.

Особливо яскраво основна роль доступності виявляється в

різного роду системах управління виробництвом, транспортом тощо. Зовні менш драматичні, але також вельми неприємні наслідки і матеріальні, і моральні - може мати тривала недоступність інформаційних послуг, якими користується велика кількість людей (продаж залізничних та авіаквитків, банківські послуги тощо).

**Під цілісністю** мається на увазі актуальність і несуперечність інформації, її захищеність від руйнування і несанкціонованої зміни.

Цілісність можна поділити на статичну (тобто незмінність інформаційних об'єктів) і динамічну (що відноситься до коректного виконання складних дій (транзакцій)). Засоби контролю динамічної цілісності застосовуються, зокрема, при аналізі потоку фінансових повідомлень з метою виявлення крадіжки, переупорядковування або дублювання окремих повідомлень.

Цілісність виявляється найважливішим аспектом ІБ в тих випадках, коли інформація служить "керівництвом до дії". Рецептура ліків, назначені медичні процедури, набір і характеристики комплектуючих виробів, хід технологічного процесу - все це приклади інформації, порушення цілісності якої може опинитися в буквальному розумінні смертельною. Неприємно і спотворення офіційної інформації, будь то текст закону або сторінка Web-сервера якої-небудь урядової організації.

**Конфіденційність** – це захист від несанкціонованого доступу до інформації.

Конфіденційність – найбільш опрацьований у нас в країні аспект інформаційної безпеки. На жаль, практична реалізація заходів по забезпеченню конфіденційності сучасних інформаційних систем натрапляє на серйозні труднощі. По-перше, відомості про технічні канали просочування інформації є закритими, так що більшість користувачів позбавлене можливості скласти уявлення про потенційні ризики. По-друге, на шляху призначеної для користувача криптографії як основного засобу забезпечення конфіденційності стоять

численні законодавчі перепони і технічні проблеми.

Якщо повернутися до аналізу інтересів різних категорій суб'єктів інформаційних відносин, то майже для всіх, хто реально використовує ІС, на першому місці стоїть доступність. Практично не поступається їй за важливістю цілісність - який сенс в інформаційній послузі, якщо вона містить спотворені відомості?

Нарешті, конфіденційні моменти є також у багатьох організацій (навіть у згадуваних вище навчальних інститутах прагнуть не розголошувати дані про екзаменаційні білети до іспиту та паролі).

### 3. Визначення та загальні властивості інформації

**Інформація** – це результат відображення та обробки в людській свідомості різноманіття навколишнього світу, відомостей про предмети, що оточують людину, явища природи, діяльність інших людей.

З такого визначення випливає, що будь-яка інформація може бути важливою. Однак, якщо обмежитися поняттям комп'ютерної системи (КС), що зараз дуже актуально, то можна дати більш просте визначення інформації – це все, що може бути представлене в КС. Відразу ж виникає питання про форми та види представлення інформації в КС.

Звичайно виділяють такі види представлення інформації, як букви, символи, цифри, слова, тексти, малюнки, схеми, формули, графіки, таблиці, плани, креслення, алгоритми і т. п. З цих видів можуть створюватися більш складні види та структури представлення інформації: команди, повідомлення, довідки, рішення, інструкції, масиви, файли, томи і т. п.

Інформація, що втілена і зафіксована в певній матеріальній формі, називається ***повідомленням***. Повідомлення можуть бути безперервними (аналоговими) і дискретними (цифровими).

***Безперервне повідомлення*** представляється деякою фізичною величиною (електричною напругою, струмом і т. д.), зміни якої відображають перебіг певного процесу. Фізична

величина, що передає безперервне повідомлення, може набувати будь-яких значень і змінюватися в довільні моменти часу. Таким чином, у безперервному повідомленні скінченої довжини може міститися велика кількість інформації.

Для ***дискретних повідомлень*** характерна наявність фіксованого набору окремих елементів, з яких у дискретні моменти часу формуються різні послідовності елементів. Важливою є не фізична природа елементів, а те, що набір елементів скінчений, і тому будь-яке дискретне повідомлення скінченої довжини передає скінчене число значень деякої величини, а отже, кількість інформації в такому повідомленні скінчена. При дискретній формі представлення інформації окремим елементам її можуть бути присвоєні числові (цифрові) значення. У таких випадках маємо цифрову інформацію.

Елементи, з яких складається дискретне повідомлення, називають буквами чи символами. Набір цих букв (символів) утворює алфавіт. Число символів в алфавіті називається об'ємом алфавіту. Дискретне повідомлення можна розбити на групи символів, що називаються словами. Зі слів можуть формуватися більш складні структури (записи, файли, томи і т. п.), але ці поняття ми не розглядаємо, тому що не є істотними для подальшого розгляду. Зауважимо лише, що найбільш простим є двійковий алфавіт.

Звичайно в КС інформація представляється двійковим алфавітом, що фізично реалізується сигналом, здатним приймати два добре помітних значення, наприклад електричну напругу високого і низького рівня, протилежні значення напруженості магнітного поля і т. п. Найважливішою вимогою до фізичних аналогів двійкового алфавіту є можливість надійного розпізнавання двох різних значень сигналу, що при описі функціонування схем позначають символами 0 (нуль) і 1 (одиниця).

Інформація в КС піддається різним процесам: введення, збереження, обробка, виведення.

Введення інформації у КС може здійснюватися з перфокарт, перфострічок, магнітних стрічок, барабанів, дисків,

дискет, клавіатури, спеціальних пультів і т. п. Збереження інформації здійснюється на запам'ятовуючих пристроях – оперативних запам'ятовуючих пристроях, різних регістрах пам'яті, магнітних стрічках, барабанах, дисках, дискетах і т. п. Обробляється у КС інформація відповідно до прийнятого в даній системі порядку (ОС, ПЗ і т. п.). Для виведення інформації є багато каналів (візуальний, звуковий, друк та ін.).

Найбільш загальними інформаційними процесами, що відбуваються в автоматизованих системах є такі:

- інформаційно-довідкове забезпечення;
- інформаційне забезпечення задач;
- обслуговування інформаційних баз.

Усі вони реалізуються персоналом за допомогою апаратних засобів, ПЗ та інформаційних баз автоматизованих систем.

## 4. Цінність та класифікація інформації

Крім представлення в КС, цікаво і важливо подивитися на інформацію з інших точок зору. Зокрема, виявляється, що інформація – це товар і, отже, є об'єктом товарних відносин. В Україні інформаційні відносини регулюються кількома законами, у тому числі і Законом «Про інформацію». Зокрема, у цьому законі в статті 18 подано класифікацію видів інформації:

- статистична інформація;
- масова інформація;
- інформація про діяльність державних органів влади й органів місцевого і регіонального самоврядування;
- правова інформація;
- інформація про особу;
- інформація довідково-енциклопедичного характеру;
- соціологічна інформація.

Оскільки інформацію можна продати, купити, імпортувати, фальсифікувати, вкрасти і т. д., то з цього випливає, що вона повинна якимось чином оцінюватися. Далі, інформація, якою обмінюється людина через машину з іншою людиною чи машиною, може бути важливою і, отже, є предметом захисту. Однак захисту підлягає не будь-яка

інформація, а тільки та, котра має ціну, тобто цінна інформація. Цінною ж стає та інформація, володіння якою дасть змогу її дійсному чи потенційному власнику одержати який-небудь виграш: моральний, матеріальний, політичний і т. д. Оскільки в суспільстві завжди існують люди, які бажають мати якісь переваги над іншими, то у них може виникнути бажання незаконним шляхом одержати цінну інформацію, а в її власника виникає необхідність її захищати. Цінність інформації є критерієм при прийнятті будь-якого рішення про її захист. Хоча було багато різних спроб формалізувати процес оцінки інформації з використанням методів теорії інформації та аналізу рішень, цей процес залишається дуже суб'єктивним.

Для оцінки потрібен розподіл інформації на категорії не тільки відповідно до її цінності, а й за важливістю. За рівнем важливості можна розділити інформацію на категорії таким чином:

- життєво важлива незамінна інформація, наявність якої необхідна для функціонування системи;
- важлива інформація – інформація, що може бути замінена чи відновлена, але процес її відновлення важкий і пов'язаний з великими витратами;
- корисна інформація – інформація, яку важко відновити, однак система може досить ефективно функціонувати і без неї;
- несуттєва інформація – інформація, без якої система продовжує існувати.

Хоча здається, що такий розподіл легко застосовувати, на практиці віднесення інформації до однієї з цих категорій може являти собою дуже важке завдання, тому що та сама інформація може бути використана багатьма підрозділами систем, кожний з яких може віднести цю інформацію до різних категорій важливості. Категорія важливості, як і цінність інформації, звичайно змінюється з часом і залежить від рівня її значущості для різних груп споживачів і потенційних порушників.

Існують визначення груп осіб, пов'язаних з обробкою інформації: власник – організація чи особа, що володіє інформацією; джерело – організація чи особа, що постачає

інформацію; **зловмисник** (ЗЛ) – організація чи особа, що прагне незаконно одержати інформацію. Для цих груп значущість однієї і тієї ж інформації може бути різною. Наприклад:

- оперативна інформація деякого підприємства (список замовлень на даний тиждень і графік виробництва) важлива для власника, а для джерела (замовника) чи порушника не має цінності;

- інформація про перспективи розвитку ринку може бути важливою для порушника, а для джерела чи власника, що завершили її аналіз, уже неважлива.

Наведені категорії важливості цілком узгоджуються з існуючим принципом розподілу інформації за рівнями таємності (або секретності). Рівень таємності – це адміністративні чи законодавчі заходи, що відповідають мірі відповідальності особи за витік конкретної інформації, регламентованої спеціальними документами, з урахуванням державних, воєнно-стратегічних, комерційних, службових чи особистих інтересів. Такою інформацією може бути державна, військова, комерційна, службова чи особиста таємниця. Рівень таємності визначається грифом, що присвоюється тій чи іншій інформації. В Україні в державних структурах встановлено такі рівні (грифи) таємності: несекретно, для службового користування, таємно, цілком таємно (Н, ДСК, Т, ЦТ). Аналогічна термінологія існує в більшості країн світу: unclassified, confidential, secret, top secret (U, C, S, TS). Така класифікація дає можливість визначити просту лінійну порядкову шкалу цінності інформації: Н < ДСК <Т<УТ(U<C<S< TS). За цією шкалою відразу видно, до якої категорії інформації необхідно висувати більш високі вимоги щодо її захисту.

Слід, однак, додати, що, у багатьох випадках захищати потрібно не тільки секретну інформацію. І несекретна інформація, що піддається несанкціонованим ознайомленням чи модифікації, може привести до витоку чи втрати пов’язаної з нею секретної інформації. Вона може привести також до невиконання автоматизованою системою функцій обробки секретної інформації. Існує також можливість витоку секретної

інформації шляхом аналізу сукупності несекретних відомостей. Усе це лише підтверджує тезу про складність класифікації інформації, яку необхідно захищати.

Як було раніше зазначено, останнім часом інформація стала найважливішим ресурсом, випереджаючи за важливістю навіть сировинні та енергетичні ресурси. Але для ефективного її використання необхідно вміти оцінювати значимість її для виконання відповідної діяльності, тобто оцінювати інформацію як об'єкт праці. Для такої оцінки необхідні показники двох видів:

- що характеризують інформацію як ресурс для забезпечення процесу отримання розв'язків різноманітних задач;
- що характеризують інформацію як об'єкт звичайної праці.

Зміст показників першого виду визначається важливістю інформації в процесі розв'язання задач, а також кількістю і складом інформації, яка використовується. Під кількістю інформації тут розуміється об'єм відомостей, які використовуються в процесі розв'язання задач, причому не абсолютний їх об'єм, а їх достатність для інформаційного забезпечення конкретних задач, їх адекватність задачам. Отже, показники першого виду можуть бути такими:

• ***важливість*** – це узагальнений показник, який характеризує значимість інформації з точки зору задач, для яких вона використовується, а також із точки зору організації її обробки. Тут оцінка може здійснюватися за: важливістю самих задач для даної діяльності; ступенем важливості інформації для ефективного виконання відповідних завдань; рівнем витрат при небажаних змінах інформації; рівнем витрат на відновлення порушень інформації. Слід зазначити, що для деяких видів інформації важливість можна досить точно оцінювати за так званим коефіцієнтом важливості, обчислення якого здійснюється на основі математичних, лінгвістичних або неформально-евристичних моделей:

• ***повнота*** – це показник, що характеризує міру

достатності інформації для розв'язання відповідної задачі. Для кількісного вираження цього показника також відомі формальні і неформальні моделі обчислення коефіцієнта повноти;

• ***адекватність*** – це ступінь відповідності інформації дійсному стану тих об'єктів, які вона відображає. Адекватність залежить від об'єктивності генерування інформації про об'єкт, а також від тривалості часу між моментом генерування та моментом оцінки адекватності. Зазначимо, що для оцінки адекватності також відомі формальні і неформальні підходи, які дозволяють отримати її кількісні оцінки;

• ***релевантність*** – це показник, що характеризує відповідність її потребам задачі, яка розв'язується. Відомий коефіцієнт релевантності – це відношення обсягу релевантної інформації до загального її обсягу. Існують моделі його обчислення;

• ***толерантність*** – показник, що характеризує зручність сприйняття та використання інформації в процесі розв'язання відповідної задачі. Це поняття є дуже широким, невизначеним і суб'єктивним, а отже, формальних методів його оцінки поки що немає.

Якщо повернутися до показників другого виду, то слід зазначити, що для них інформація виступає як:

- ***сировина,*** яку добувають та обробляють;
- ***напівфабрикат***, що виникає в процесі обробки сировини;
- ***продукт*** для використання.

Тобто тут маємо звичайний виробничий ланцюжок: добування сировини – переробка для отримання напівфабрикатів – їх переробка для отримання кінцевого продукту. Нагадаємо, що при цьому всі стадії переробки інформації мають задовольняти показники першого виду. Зрозуміло, що тут найбільш важливими є форма або спосіб представлення інформації, а також її об'єм. Отже, як показники другого виду можуть виступати: 1) спосіб або система кодування інформації, тобто ефективність кодування; 2) об'єм кодів, що відображають дану інформацію. Відзначимо, що

методи визначення цих показників досить повно розроблені в теорії інформації.

Таким чином, необхідний рівень захисту інформації слід визначати з урахуванням значень усіх розглянутих вище показників, а також грифів таємності.

Насамкінець звернемо увагу на розбіжність між визначеною вище таємністю і безпекою інформації. **Безпека інформації** – це захист інформації від негативних впливів на неї, і вона має відношення до технологічних процедур забезпечення захисту. **Таємність інформації** – це статус інформації, який фіксується залежно від її важливості і вимагає певного рівня її захищеності. Отже, це поняття має відношення до людей, окремих осіб, які відповідають за інформацію і вирішують, яку інформацію можна розкрити, а яку приховати від інших людей.

## 5. Інформація як об'єкт власності

Фактично сфера безпеки інформації – не захист інформації, а захист прав власності на неї. Щоб переконатися в цьому, розглянемо особливості інформаційної власності.

Історично традиційним об'єктом права власності є матеріальний об'єкт, а це означає, що фактично право власності було речовим правом.

Інформація ж не є матеріальним об'єктом, інформація – це знання, тобто відображення дійсності у свідомості людини (причому правильне чи помилкове відображення – неістотно, важливо, що у свідомості). І тільки згодом інформація може втілюватися в матеріальні об'єкти навколишнього світу. Однак, не будучи матеріальним об'єктом, інформація нерозривно пов'язана з матеріальним носієм: це – мозок людини чи відчужені від людини матеріальні носії, такі як книга, дискета та інші види «пам'яті» (запам'ятовуючі пристрої).

Можливо, з філософської точки зору можна говорити про інформацію як про деяку абстрактну субстанцію, що існує сама по собі. Але з конкретної, матеріальної точки зору ні збереження, ні передача інформації поки що без матеріального

носія неможливі. Унаслідок цього можна сформулювати такі особливості інформації як об'єкта власності.

1. Інформація як об'єкт права власності може копіюватися (тиражуватися) за допомогою матеріального носія. Матеріальний об'єкт права власності, як відомо, копіювати неможливо (принаймні поки що). Справді, якщо розглянути дві однакові речі (одяг, автомобіль тощо), то вони складаються з однакових структур, але все-таки вони різні (чим детальніше їх розглядати, тим більше вони будуть розрізнятися). Тим часом інформація при копіюванні залишається тою ж, це те саме знання.

2. Інформація як об'єкт права власності легко переміщується до іншого суб'єкта права власності без очевидного (принаймні помітного) порушення права власності на інформацію. Переміщення ж матеріального об'єкта від одного суб'єкта (без його згоди) до іншого неминуче спричиняє втрату первісним суб'єктом права власності на цей об'єкт, тобто відбувається очевидне порушення його права власності.

3. Небезпека копіювання і переміщення інформації збільшується тим, що вона, як правило, відчужувана від власника, тобто зберігається та обробляється в сфері доступності великого числа суб'єктів, що не є суб'єктами права власності на цю інформацію (автоматизовані системи, мережі ЕОМ і т. п.). Крім відзначених особливостей інформації як об'єкта власності в іншому, мабуть, вона нічим не відрізняється від традиційних об'єктів права власності.

Справді, право власності, як відомо, включає три повноваження власника, що становлять у цілому зміст права власності: право розпоряджання, право володіння, право користування. Стосовно інформації можна сказати, що суб'єкт права власності на інформацію може передати частину своїх прав (право розпоряджання), не втрачаючи їх сам, іншим суб'єктам, наприклад власнику матеріального носія інформації (це – володіння чи користування) чи користувачу (це – користування і, можливо, володіння).

Для інформації право розпоряджання передбачає

виключне право (ніхто інший, крім власника) визначати, кому ця інформація може бути надана (у володіння чи користування). Право володіння передбачає володіння цією інформацією в незмінному вигляді. Право користування передбачає право використовувати цю інформацію у своїх інтересах. Таким чином, до інформації, крім суб'єкта права власності на неї, можуть мати доступ і інші суб'єкти права власності як законно, санкціоновано, так і (внаслідок відзначених вище особливостей) незаконно, не санкціоновано. Тому виникає дуже складна система взаємин між різними суб'єктами права власності. Ці взаємини повинні регулюватися та охоронятися, тому що відхилення від них тягнуть порушення прав власності на цю інформацію. Реалізацією права власності на інформацію звичайно займається певна інфраструктура (державна чи приватна).

Як і для будь-якого іншого об'єкта власності, для інформації така інфраструктура складається з ланцюжка: законодавча влада – судова влада – виконавча влада (закон – суд – покарання). Тому, незважаючи на ряд особливостей, інформація поряд з матеріальними об'єктами може і повинна розглядатися законом як об'єкт права власності.

Будь-який закон про власність з метою захисту прав власника, зафіксувавши суб'єкти та об'єкти права власності, повинен регулювати відносини між ними. Особливості регулювання цих відносин залежать від специфіки об'єктів права власності. У випадку інформаційної власності закон повинен регулювати відносини суб'єктів, а також суб'єктів і об'єктів права власності на інформацію з метою захисту прав як власника, так і законних власників і користувачів інформації для захисту інформаційної власності від розголошення, витоку, обробки (копіювання, модифікації чи знищення) інформації.

В Україні прийнято закони «Про інформацію» і «Про захист інформації в автоматизованих системах». У першому законі в статті 39 установлено, що інформаційна продукція та інформаційні послуги громадян і юридичних осіб, що займаються інформаційною діяльністю, можуть бути об'єктами

товарних відносин, регульованих цивільним і іншим законодавством. Інакше кажучи, інформація – це товар. Інформаційна продукція (стаття 40) – це матеріалізований результат інформаційної діяльності, призначений для задоволення інформаційних потреб громадян, державних органів, підприємств, закладів і організацій. Інформаційна послуга (стаття 41) – це здійснення у визначеній законом формі інформаційної діяльності з доставки інформаційної продукції до споживачів з метою задоволення їхніх інформаційних потреб.

## 6. Інформація як комерційна таємниця

Поняття «комерційної таємниці» у нашій країні поки що в законодавчих актах не визначено. Для розуміння цього поняття подамо його визначення, що використовується в нормативних актах інших країн. Зокрема, у статті 33 закону «Про підприємства в СРСР» від 1 січня 1991 року дається таке визначення: «Під комерційною таємницею підприємства розуміються відомості, що не є державними секретами і пов'язані з виробництвом, технологією, керуванням, фінансами та іншою діяльністю підприємства, розголошення (передача, витік) яких може завдати шкоди його інтересам».

Склад і обсяг відомостей, що становлять комерційну таємницю, визначаються керівником підприємства.

Які саме відомості можуть вважатися комерційною таємницею? Наприклад, 5 грудня 1991 року уряд Росії прийняв постанову №35 «Про перелік відомостей, що можуть становити комерційну таємницю». В основному перелік подібних відомостей відомий із законів про банківську діяльність в інших країнах. Проте, для прикладу, наведемо перелік відомостей із згаданої постанови. У цій постанові вони групуються за тематичним принципом. Звичайно, відомості, що включені в даний перелік, можуть бути комерційною таємницею тільки з урахуванням особливостей конкретного підприємства (організації).

1. Відомості про фінансову діяльність:
   - прибуток, кредити, товарообіг;

- фінансові звіти і прогнози;
- комерційні задуми;
- фонд заробітної плати;
- вартість основних і оборотних коштів;
- кредитні умови платежу;
- банківські рахунки;
- планові і звітні калькуляції.

2. Інформація про ринок:

- ціни, знижки, умови договорів, специфікації продуктів;
- обсяг, історія, тенденції виробництва і прогноз для конкретного продукту;
- ринкова політика і плани;
- маркетинг і стратегія цін;
- відносини зі споживачами і репутація;
- чисельність і розміщення торгових агентів;
- канали і методи збуту;
- політика збуту;
- програма реклами.

3. Відомості про виробництво і продукцію:

- відомості про технічний рівень, техніко-економічні характеристики виробів, що розробляються;
- відомості про плановані терміни створення розроблюваних виробів;
- відомості про модифікацію і модернізацію раніше відомих технологій, процесів, устаткування;
- виробничі потужності;
- стан основних і оборотних фондів;
- організація виробництва;
- розміщення і розмір виробничих приміщень і складів;
- перспективні плани розвитку виробництва;
- технічні специфікації існуючої і перспективної продукції;
- схеми і креслення окремих вузлів, готових виробів, нових розробок;
- відомості про стан програмного і комп'ютерного забезпечення;
- оцінка якості й ефективності;

- номенклатура виробів;
- спосіб упакування;
- доставка.

4. Відомості про наукові розробки:

- нові технологічні методи, нові технічні, технологічні і фізичні принципи, заплановані до використання в продукції підприємства;
- програми НДР;
- нові алгоритми;
- оригінальні програми.

5. Відомості про матеріально-технічне забезпечення:

- відомості про склад торгових клієнтів, представників і посередників;
- потреби в сировині, матеріалах, комплектуючих вузлах і деталях, джерела задоволення цих потреб;
- транспортні й енергетичні потреби.

6. Відомості про персонал підприємства:

- чисельність персоналу;
- визначення осіб, що приймають рішення.

7. Відомості про принципи керування підприємством:

- відомості про застосовувані і перспективні методи керування виробництвом;
- відомості про факти ведення переговорів, предмети і цілі нарад і засідань органів керування;
- відомості про плани підприємства щодо розширення виробництва;
- умови продажу і злиття фірм.

8. Інші відомості:

- важливі елементи систем безпеки, кодів і процедур доступу до інформаційних мереж і центрів;
- принципи організації захисту комерційної таємниці.

У багатьох країнах існують закони, що регламентують банківську діяльність. У них визначене поняття «банківської таємниці». Під **банківською таємницею** (БТ) мається на увазі обов'язок кредитної установи зберігати таємницю про операції клієнтів, убезпечення банківських операцій від ознайомлення з

ними сторонніх осіб, насамперед конкурентів того чи іншого клієнта, таємницю щодо операцій, рахунків і внесків своїх клієнтів і кореспондентів. Інакше БТ можна визначити як особисту таємницю вкладника. У підсумку комерційна таємниця банку включає комерційну таємницю самого банку та особисту таємницю вкладника. В Україні подібного закону поки що немає.

Отже, під **інформаційною безпекою** слід розуміти захист інтересів суб'єктів інформаційних відносин.

### Контрольні питання до Лекції 1

1.Інформаційна безпека (поняття і визначення).
2.Які ви знаєте об’єкти захисту?
3.Що включає поняття «безпечна діяльність»?
4.Що таке інформація?
5.Класифікація інформації.
6.Категорії інформації.
7.Інформація як об’єкт власності.
8.Інформація як комерційна таємниця.

## Лекція 2

# ПРОБЛЕМА ЗАХИСТУ ОПЕРАЦІЙНИХ СИСТЕМ

**План лекції:**

1.Основні вимоги до безпеки комп'ютерних систем.
2.Класифікація загроз безпеці комп'ютерних систем.
3.Рівні (варіанти) захисту операційних систем.
4.Об'єкти захисту в операційних системах.

### 1. Основні вимоги до безпеки комп'ютерних систем

Для розуміння того, якого роду загрозам можуть підлягати комп'ютерні системи, визначимо вимоги до безпеки. Зазвичай висувають такі основні чотири вимоги.

***Конфіденційність***. Згідно цій вимозі, інформацію від комп'ютерних систем (КС) можуть отримувати тільки авторизовані особи. Це включає в себе виведення на друк або на екран та інші форми подання інформації, в тому числі і саме виявлення існування об'єкту.

***Цілісність.*** Передбачається, що характеристики КС можуть змінювати лише авторизовані особи. Під змінами тут маються на увазі запис, редагування, зміна статусу, видалення і створення нових об'єктів.

***Доступність.*** Необхідно, щоб характеристики КС були доступними авторизованим особам.

***Аутентичність.*** Комп'ютерна система повинна мати можливість перевіряти ідентичність користувача.

**Сучасні комп'ютерні системи** – це складний механізм, що складається з великої кількості компонентів різного ступеня автономності, які пов'язані між собою, і даних, якими вони обмінюються. Практично кожний механізм може вийти з ладу або піддатися зовнішньому впливу.

**Загроза безпеці** - потенційно можливий вплив на КС, який може прямо або побічно завдати шкоди користувачам або власникам КС.

**Атака** - реалізація загрози.

## 2. Класифікація загроз безпеці комп'ютерних систем

Загрози безпеці КС, враховуючи тільки навмисні загрози, можна класифікувати за ознакам.

### Загроза за метою реалізації

Розглянемо роботу КС в процесі надання інформації.

Взагалі, інформація якимось чином надходить від джерела (наприклад, від файла, від області основної пам'яті) до одержувача (наприклад, до іншого файла або до користувача) (рис.2а). В залежності від того, як ця інформація надається, розрізняють чотири категорії атак:

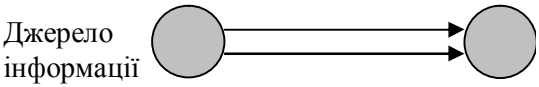
***Переривання,*** порушення працездатності КС (часткове або повне), тобто виведення з ладу або некоректна зміна режимів роботи КС чи заміна, в результаті чого отримуються невірні результати, КС не може правильно обробляти інформацію. Відмова від потоку інформації, тобто одна із взаємодіючих сторін не визнає факт передачі або прийому повідомлень, замовлень, фінансових узгоджень, донесень.

Компоненти системи виходять з ладу, стають недоступними або непридатними (рис.2б). Метою цієї атаки є ***порушення доступності***. Щодо даних, то інформація, яка зберігається і обробляється на комп'ютері, може мати велику цінність для її власника, і її використання іншими особами наносить значну шкоду власнику.

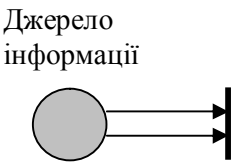
***Перехоплення.*** Це атака, метою якої є порушення конфіденційності, в результаті чого доступ до компонентів системи отримують несанкціоновані сторони (рис.2в). В ролі несанкціонованої сторони може бути особа, програма або комп'ютер. Прикладом цього можуть бути перехоплення повідомлень по мережі, незаконне копіювання файлів або програм.

***Зміна*** (повна або часткова, компрометація або дезінформація). Несанкціонована сторона не тільки отримує доступ до системи та її об'єктів, але й втручається в роботу її компонентів (рис.2г). Метою цієї атаки є порушення цілісності. Приклади: заміна значень у файлі даних, зміна програми таким чином, що вона працюватиме по-іншому, зміна вмісту

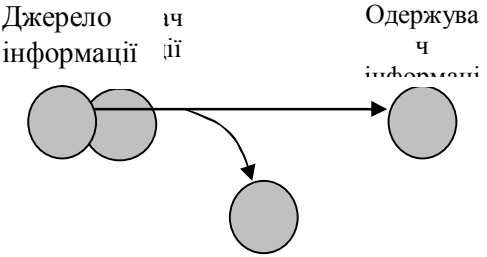
переданих по мережі повідомлень. Цінна інформація може бути втрачена або знецінена шляхом її незаконного вилучення або модифікації.



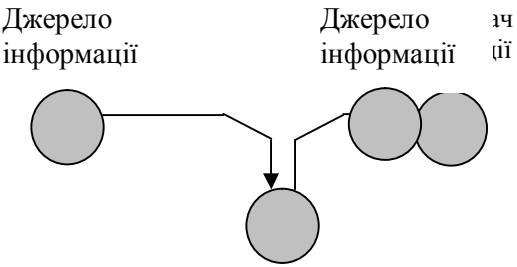

а) Нормальна передача інформації



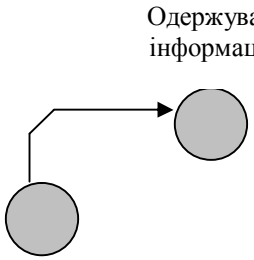

б) Переривання (порул




в) Перехоплення
(порушення конфіденційності)




г) Змі
(порушення ц




д) Підробка
(порушення автентичності)

Рис. 2. Загрози безпеці КС за метою реалізації

***Підробка.*** Несанкціонована сторона розміщує в системі підроблені об'єкти (рис.2д). Метою цієї атаки є порушення

аутентичності. Прикладами можуть служити розміщення в мережі підробних повідомлень або додавання записів у файл.

**Загроза за об'єктом атаки**

Впливу можуть піддаватися такі об'єкти комп'ютерної системи.

***Комп'ютерна система в цілому.*** Зловмисник намагається проникнути в систему для виконання несанкціонованих дій. Тут може бути використаний метод так званого "маскараду", перехоплення або підбір паролю, зламування КС або доступ до неї через мережу.

***Об'єкти комп'ютерної системи.*** Дані або програми в оперативній пам'яті (ОП) або на зовнішніх носіях, самі пристрої системи як зовнішні (дисководи, термінали, мережні пристрої), так і внутрішні (оперативна пам'ять, процесор), канали передачі даних. Метою такої загрози є або доступ до змісту інформації (порушення конфіденційності, цілісності), або порушення функціональності (заповнення всієї ОП безглуздою інформацією, завантаження процесора завданням з необмеженим часом виконання).

Розділяють такі об'єкти КС.

***Апаратне забезпечення***. Основна загроза для нього пов'язана з його доступністю. Апаратне забезпечення найбільше підлягає атакам і найменше піддається автоматичному керуванню. В число загроз входять випадкове або наперед сплановане виведення обладнання з ладу, а також його крадіжка. Розповсюдженість ПК та використання мереж призводять до збільшення потенційних можливостей втрат в цій області. Для запобігання загрозам подібного роду потрібні адміністративні міри щодо попередження фізичного доступу до систем.

***Програмне забезпечення (ПЗ).*** Основна небезпека для ПЗ полягає в його доступності. Програми, особливо прикладні, надзвичайно легко знищити. Крім того, ПЗ може бути спотворено або змінено, в результаті чого воно стане неприйнятним для використання. Акуратне управління налаштуванням конфігурації програм, зберігання резервних копій допоможе підвищити надійність їх роботи.

Складніше розв'язати проблему, якщо зміна програми призводить до того, що вона продовжує працювати, але поводить себе не так, як потрібно. Ця категорія атак пов'язана з комп'ютерними вірусами. Крім того, неабияке значення має захист програмного забезпечення від несанкціонованого доступу (НСД) та несанкціонованого копіювання (НСК);

***Дані.*** Безпека даних охоплює широкий круг питань, що включає себе і доступність, і секретність, і цілісність. Коли йдеться про доступність, мається на увазі захист від випадкового або передбаченого спотворення файлів з даними. Для забезпечення секретності даних необхідно турбуватись про несанкціоноване зчитування файлів даних і баз даних. Особливої уваги заслуговують статистичні бази даних, в яких зберігається інформація індивідуального характеру або інформація підприємств та відомств, що не повинна підлягати загальному перегляду чи розголошенню. Крім того, важливою задачею є зберігання цілісності даних, оскільки зміна файлів даних може мати різні наслідки – від незначних до катастрофічних;

***Лінії зв'язку та канали передач.*** Це пакети даних, які передаються по каналах зв'язку (атака на об'єкт мережі) або самі канали. Це може бути і підслуховування каналу (порушення конфіденційності) та аналіз трафіка (потік повідомлень), заміна або модифікація повідомлень в каналах зв'язку (порушення цілісності), заміна топології і характеристик мережі, правил комутації і адресації.

***Суб'єкти комп'ютерної системи.*** Процеси і підпроцеси користувачів, підсистеми, мережі. Мета таких атак – або прямий вплив на роботу процесу (призупинення, зміна привілеїв та характеристик), або зворотний вплив (використання зловмисником привілеїв, характеристик процесу для своєї мети), або впровадження зловмисником вірусу в середовище другого процесу і його виконання від імені цього процесу.

**Загроза за принципом впливу на КС**

Загрози з використанням доступу суб'єкта КС (користувача, процесу) до об'єкта (файла даних, канала зв'яку і т.д.).

***Загрози з використанням прихованих каналів.*** **Прихований канал (convert channel)** – це шлях передачі інформації, що дозволяє двом взаємодіючим процесам обмінюватись інформацією способом, який порушує системну політику безпеки КС. Вони бувають 2 видів:

- ***канали з пам'яттю*** (convert storage channel), коли здійснюється читання або записування інформації другого каналу за допомогою проміжних об'єктів для зберігання інформації (тимчасова пам'ять);
- ***тимчасові канали*** (convert timing channel), коли один процес може отримувати інформацію про хід другого, використовуючи інтервали між якими-небудь подіями (наприклад, аналіз часового інтервалу між запитом на введення-виведення дозволяє зробити висновок про розмір введеної інформації).

Взаємодія, яка базується на першому принципі, простіша, більш інформативна, тут відбувається взаємодія суб'єкта і об'єкта, що змінює стан КС. Від загрози такого роду легше захиститися, оскільки її легше виявити.

Взаємодія на основі другого принципу менш інформативна, використовуються лише побічні ефекти, що не впливає на стан КС, і тому така загроза більш складна для її виявлення та усунення.

***Загроза за характером впливу на КС та його об'єкти.*** Активний вплив пов'язаний з виконанням користувачем будь-яких дій, що виходять за рамки його обов'язків і порушують існуючу політику безпеки (доступ до певних наборів даних, програм, підбір пароля). Цей вплив призводить до зміни стану КС. Він може здійснюватись як з використанням доступу, так і при допомозі прихованих каналів.

***Активний вплив*** проявляється у випадках:

- безпосереднього впливу на об'єкт атаки (у тому числі з використанням привілеїв). Наприклад, безпосередній доступ до набору даних, програми, служби, каналу зв'язку, якщо використовувати яку-небудь помилку в захисті КС. Цим діям можна запобігти, використовуючи контроль доступу. Активні

атаки викликають деяку зміну потоків даних і розділяються на 4 категорії:

а) імітація – має місце, коли деякий об'єкт видає себе за інший Як правило, атака з імітацією використовується разом з активними атаками інших видів. Наприклад, може перехоплюватись, а потім використовуватись послідовність повідомлень, що передаються в процесі аутентифікації, в результаті чого авторизовані сторони з малими привілеями отримують додаткові привілеї, які їм не було надано;

б) відтворення – включає в себе пасивне перехоплення елементів даних з їх наступним повторним передаванням з метою здійснити не авторизований доступ;

в) зміна повідомлень – зміна якоїсь частини початкового повідомлення, видалення повідомлення або зміна порядку його отримання;

г) відмова від обслуговування – заважає нормальному використанню засобів зв'язку або керування ними чи їх стримування. Метою цієї атаки можуть бути, наприклад, підрив роботи мережі, виведенням її з ладу, перевантаження повідомленнями для зниження її працездатності, знищення повідомлень, призначених конкретному адресату (приватному або офіційному).

• впливу на систему дозволів (у тому числі захоплення привілеїв). Тут несанкціоновані дії виконуються щодо прав користувача на об'єкт атаки, а сам доступ потім здійснюється вже законним шляхом.

Опосередкований вплив (через інших користувачів) є дуже небезпечним Для запобігання йому необхідний постійний контроль як з боку адміністраторів, операторів, так і з боку користувачів за своїми даними.

***Опосередкований*** вплив проявляється у випадках:

- "маскараду" – користувач присвоює собі повноваження іншого користувача, видаючи себе за нього;

- "використання навмання" – один користувач заставляє іншого виконувати необхідні дії, причому останній може і не підозрювати про це. Для реалізації цих загроз може

використовуватись вірус (це так звані шкідливі програми: “троянський кінь” і “черв’як”).

Пасивний вплив здійснюється шляхом спостереження користувачем побічних ефектів та їх аналізу (підслуховування ліній зв’язку між двома вузлами мережі), що порушує конфіденційність. Стан КС при цьому не змінюється. До ***пасивних атак*** відносяться:

- добування вмісту повідомлень під час телефонної розмови або за допомогою електронної пошти, під час яких може бути передано важливу або конфіденційну інформацію;

- аналіз трафіка (traffic analysis). Припустимо, в змісті інформації, що передається, є можливість приховати повідомлення (припустимо, за допомогою шифрування), і тоді не можна добути з нього інформацію. Але опонент може отримати відомості про характер повідомлення: визначити місце розташування і параметри вузлів, що обмінюються інформацією, зібрати відомості про частоту передавання повідомлень, їх розмір, а потім зробити висновки щодо переданої інформації.

Пасивні атаки складно виявити, оскільки вони не приводять ні до яких змін даних. Але передбачити ці атаки можливо. Таким чином, слід зосередити увагу не на виявленні пасивних атак, а на їх попередженні.

***Загроза через появи помилок захисту***. Неадекватність політики безпеки КС. Розроблена політика безпеки для даної КС настільки не відображає реальні аспекти обробки інформації, що стає можливим використати цю невідповідність для НСД. Всі КС мають деяку невідповідність, але треба планувати політику безпеки таким чином, щоб вона не могла призвести до порушень. Спосіб запобігти цій загрозі – замінити засоби захисту для реалізації нової політики безпеки.

Помилки адміністративного керування – некоректна реалізація або неадекватна підтримка прийнятої політики безпеки в даній комп’ютерний системі. Наприклад, доступ користувачів до певного набору даних повинен бути закритий, а фактично (через неуважність адміністратора) цей набір

доступний всім користувачам. Для запобігання цій загрозі досить виправити таку помилку.

Помилки в алгоритмах програм – це помилки, допущені на етапі проектування програм, завдяки чому їх можна використати зовсім не так, як описано в документації. Приклад: помилки в програмі аутентифікації користувача, коли за допомогою певних дій користувач має можливість увійти в систему без пароля. Ці помилки важко знайти. Щоб усунути таку загрозу, слід змінити програму або комплекс програм.

Помилки реалізації алгоритмів програм (помилки кодування) – виникають на етапі реалізації або наладки. Прикладом таких помилок можуть служити так звані "люки". Їх виявити найважче.

***Загроза за способом впливу на КС***. Вплив на КС систему може здійснюватись в одному з режимів:

- в інтерактивному режимі – користувач може активно впливати на хід виконання програми, вводячи різні команди або дані. До них належать інтерпретатори командних мов, деякі утиліти, програми керування базами даних, тобто, програми, які орієнтовані на роботу з користувачем. При використанні програм цього класу (наприклад, для атаки на КС за допомогою командного інтерпретатора) вплив є довшим у часі і тому має більшу імовірність бути виявленим;

- у пакетному режимі – коли всю інформацію треба готувати заздалегідь. Це системні і прикладні програми, які орієнтовані на виконання яких-небудь суворо визначених дій без участі користувача. Вплив за допомогою цих програм є короткостроковим. Його важко діагностувати, він є більш небезпечний, вимагає більшої попередньої підготовки для того, щоб передбачити всі можливі наслідки втручання.

***Загроза за засобами атаки.*** Для впливу на КС зловмисник може використовувати:

- стандартне програмне забезпечення. У цьому випадку результати впливу здебільшого передбачені, оскільки ці програми здебільшого добре вивчені;

- спеціально розроблені програми. Це може бути

небезпечно, і тому в захищуваних системах бажано не допускати установлення в КС програм без дозволу адміністратора.

***Загроза за станом об'єкта атаки***. Стан об'єкта в момент атаки може зробити суттєвий вплив на результати атаки і на роботу з ліквідації її наслідків. Об'єкт може бути в одному зі станів:

- зберігання – на диску, в оперативній пам'яті, в іншому пасивному стані. При цьому доступ до об'єкта здійснюється з використанням доступу;
- передачі – по лінії зв'язку між вузлами мережі, або в середині вузла. Вплив передбачає підслуховування, перехоплення, спотворення і т.д.;
- обробки – це коли об'єктом стає процес користувача.

## 3. Рівні захисту операційних систем

У відповідності з якістю наданого захисту існують різни варіанти захисту. Конкретна ОС може надавати захист різного ступеню для різних об'єктів, користувачів та додатків.

***Відсутність захисту.*** Цей варіант підходить, коли відповідні процедури виконуються за часом окремо.

***Ізоляція.*** Кожний процес працює окремо від інших процесів, не використовуючи сумісно ніякі ресурси і не обмінюючись інформацією. Кожний процес має свій адресний простір, свої файли та інші об'єкти.

***Повний розподіл або повна його відсутність***. Власник об'єкта (файла, сегмента пам'яті) об'являє його відкритим або закритим. У першому випадку доступ до об'єкта може отримати будь-який процес; у другому – доступ до цього об'єкта надається тільки власнику.

***Спільне використання з обмеженням доступу***. ОС перевіряє дозволеність доступу кожного окремого користувача до кожного окремого об'єкта. В цьому сенсі ОС виступає в якості охоронця, гарантуючи, що доступ до об'єкта отримають тільки авторизовані користувачі.

***Спільне використання за допомогою динамічних***

***можливостей.*** Цей варіант розширює концепцію контролю доступу, дозволяючи динамічно створювати права спільного використання об'єкта.

***Обмеження на використання об'єкта.*** При цьому обмежується не стільки доступ до об'єкта, скільки його використання. Наприклад, користувачеві дозволено переглядати важливий документ, але не роздруковувати, або користувач має доступ до бази даних, може брати з неї статистичні зведення, але не має можливості визначити значення певних величин.

Необхідно, щоб в операційній системі підтримувався баланс між можливостями спільного використання компонентів КС, що сприяє підвищенню ефективності її використання, і ступенем захищеності ресурсів окремих користувачів.

## 4. Об'єкти захисту в операційних системах

В основі багатозадачності лежить здатність системи надавати користувачам можливість спільного використання ресурсів. Об'єктом спільного використання є не тільки процесор, але й такі елементи як:

- пам'ять;
- пристрої введення-виведення (наприклад, диски, принтери);
- програми;
- дані.

### Захист пам'яті

В багатозадачному середовищі захист пам'яті стає важливою проблемою. Тут виникають питання, пов'язані не тільки з безпекою, але й з правильною роботою різних активних процесів. Якщо один з процесів необережно запише що-небудь в область пам'яті іншого процесу, то цим він може порушити роботу останнього.

Розподіл простору пам'яті між різними процесами легко здійснюється за допомогою використання схеми віртуальної пам'яті (сегментної, сторінкової або комбінованої). Якщо треба забезпечити повну ізоляцію, то операційній системі достатньо буде впевнитись, що кожний сегмент і кожна сторінка доступні

тільки тому процесу, якому вона розподілена. Цього легко досягти, слідкуючи, щоб в таблицях сторінок та/або сегментів не було елементів, що повторюються. Якщо ж спільне використання дозволено, то один і той самий сегмент або сторінка можуть з'явитися в декількох таблицях. Цей тип спільного використання легше всього здійснити в ОС, що підтримують сегментацію або комбінацію сегментації з розбиттям на сторінки. У цьому випадку програмний додаток може об'явити окремі сегменти доступними або недоступними для спільного використання, і тоді в роботу вступають механізми синхронізації.

Прикладом апаратної підтримки захисту пам'яті є ОС, де кожному сторінковому блоку основної пам'яті ставиться у відповідність 7-бітовий ключ управління, значення якого встановлює операційна система. Два біти цього ключа вказують на те, чи були звернення до сторінки та чи здійснювались зміни сторінки (ці біти використовуються алгоритмами заміщення сторінок). Наступні чотири біти є розрядами керування доступом (R, W, E, A). Ще один біт – розряд захисту від вибірки, який і використовується механізмом захисту.

Для отримання дозволу на доступ до якоїсь сторінки, у зверненнях процесора до пам'яті і у зверненнях пристроїв прямого доступу до пам'яті (Direct Memory Access – DMA) повинен використовуватись відповідний ключ. В розряді захисту від вибірки вказується, чи дає ключ управління доступом право тільки записування, чи право записування і читання. В процесорі існує регістр "слово стану програми" (Program Status Word – PSW), який містить інформацію щодо виконуваного в даний момент процесу. Складовим елементом цього слова є чотирибітовий ключ PSW. Поточний ключ PSW порівнюється з кодом доступу, і дозвіл на запис буде отриманий лише при умові співпадіння ключів. Якщо біт вибірки встановлено, то ключ PSW повинен відповідати ключу доступу для читання.

**Контроль доступу як захист даних та програм**

Міри по контролю доступу можна розбити на дві категорії:

Контроль доступу, що здійснюється по відношенню до користувача. Його часто називають аутентифікацією, що не зовсім правильно, оскільки цей термін широко використовується у зв'язку з аутентифікацією повідомлень, тобто може бути використаний для різних цілей. Найбільш розповсюджений контроль доступу користувача – це процедура реєстрації, при якій користувачеві необхідно ввести свій ідентифікатор та пароль. Система ж дозволить увійти лише тоді, коли його ідентифікатор співпаде з відомим системі і коли користувач знає пароль, пов'язаний з цим ідентифікатором.

Контроль доступу, орієнтований на дані, полягає в тому, що кожному користувачеві може відповідати профіль, в якому вказуються дозволені операції і режими доступу до файлів.

**Категорії зломщиків**

Зломщики (хакери, крекери) є найбільш відомою загрозою для безпеки операційних систем (інший вид загроз – це віруси). Ідентифікують такі класи зломщиків:

- **удавальник** (рос. "притворщик") – особа, що не має повноважень щодо використання комп'ютера, яка проникає в систему, не дивлячись на контроль доступу системи, і використовує обліковий запис законного користувача. Удавальник – це, як правило, стороння людина;

- **правопорушник** – законний користувач (зазвичай не стороння людина), що отримує доступ до даних, програм та ресурсів, до яких у нього немає доступу, або той, у якого є доступ, але він зловживає своїми привілеями;

- **таємний користувач** – особа, яка заволоділа управлінням в режимі суперкористувача і використовує його для того, щоб запобігти аудита і перебороти контроль доступу, або для запобігання збору даних щодо аудита. Це може бути як стороння людина, так і не стороння.

Наслідки атак зловмисників можуть бути різними – від незначних до досить таки серйозних. Внизу шкали зломщиків розташовані ті, які хочуть просто використати мережі і

дізнатись, що і де знаходиться. На протилежному кінці шкали розміщуються індивідууми, що намагаються прочитати службові дані, використати несанкціоновану зміну даних або зруйнувати систему.

**Атаки на рівні операційних систем**

В загальному випадку програмне забезпечення будь-якої універсальної комп'ютерної системи складається з трьох основних компонентів: операційної системи, мережевого програмного забезпечення і системи управління базами даних. Тому і методи зламу захисту КС можна поділити на 3 групи:

- атаки на рівні операційних систем;
- атаки на рівні мережевого програмного забезпечення;
- атаки на рівні систем керування базами даних.

Внутрішня структура сучасних операційних систем є надзвичайно складною, і тому дотримання адекватної політики безпеки є досить складним завданням.

Успіх реалізації того чи іншого алгоритму хакерської атаки на практиці в значній мірі залежить від архітектури і конфігурації конкретної ОС, яка є об'єктом цієї атаки. Однак, існують атаки, яким може підлягати практично будь-яка операційна система.

**1**. ***Крадіжка пароля:***

- підглядання за користувачем, коли той вводить пароль, що дає право на роботу з ОС (навіть якщо під час його введення пароль не висвітлюється на екрані, хакер може легко дізнатись про нього, просто спостерігаючи за переміщенням пальців користувача по клавіатурі);

- отримання пароля з файла, в якому цей пароль було збережено користувачем, який не бажає утруднювати себе введенням паролю при підключенні до мережі (як правило, такий пароль зберігається у файлі навіть у незашифрованому вигляді);

- пошук пароля, який користувач, щоб не забути його, записує на календарях, в записниках або на зворотній стороні клавіатури (особливо часто таке трапляється, коли адміністратори заставляють користувачів застосовувати паролі,

що тяжко запам’ятати);

- крадіжка зовнішнього носія парольної інформації (дискети, електронного ключа, на яких зберігається пароль користувача, призначений для входу в систему);

- повний перебір всіх можливих варіантів паролю;

- підбір пароля за частотою зустрічаємих символів та біграм, за допомогою словників найуживаніших паролів, із залученням знань про конкретного користувача – його імені, прізвища, номера телефону, дати народження, тощо.

**2.** ***Сканування жорстких дисків комп’ютера***, коли хакер намагається послідовно звернутись до кожного файла. Якщо об’єм диска досить великий, то можна бути впевненим, що при описі доступу до файлів і каталогів адміністратор припустив хоча б одну помилку, в результаті чого, всі такі каталоги і файли будуть прочитані хакером. Для знищення слідів хакер може організувати цю атаку під чужим іменем.

**3.** ***Збирання “сміття”*** – якщо засоби ОС дозволяють відновлювати раніше знищені об’єкти. Тоді хакер може використати цю можливість щоб отримати доступ до об’єктів, знищених іншими користувачами (наприклад, передивляючись вміст їх “сміттєвих кошиків”).

**4.** ***Перевищення повноважень*** – використовуючи помилки в програмному забезпеченні або в адмініструванні ОС, хакер отримує повноваження, що перевищують надані йому згідно діючій політиці безпеки:

- запуск програм від імені користувача, що має ці необхідні повноваження, або в якості системної програми (драйвера, сервісу, домена і т.д.);

- підміна динамічно завантажуваної бібліотеки. що використовується системними програмами, або надання інших значень змінним середовища, що описують шлях до таких бібліотек;

- модифікація коду або даних підсистеми захисту самої операційної системи.

**5.** ***Відмова в обслуговуванні***. Метою цієї атаки є часткове або повне виведення з ладу операційної системи:

- захоплення ресурсів (хакерська програма здійснює захоплення всіх наявних в ОС ресурсів, а потім входить в нескінчений цикл);

- бомбардування запитами (хакерська програма постійно направляє операційній системі запити, реакція на які потребує залучення значних ресурсів комп'ютера);

- використання помилок в програмному забезпеченні або адмініструванні.

**Контрольні питання до Лекції 2:**

1. Навести основні вимоги до безпеки комп'ютерних систем.
2. Як класифікуються загрози безпеці КС за метою реалізації?
3. Назвіть об'єкти атаки КС, які можуть підлягати загрозам.
4. Охарактеризувати загрози безпеці КС за принципом та характером впливу.
5. Яким загрозам може підлягати КС через появу помилок у захисті?
6. Дати характеристику загрозам КС за способом впливу, засобами атаки та станом об'єкта атаки.
7. Назвати рівні атаки на операційну систему і охарактеризувати кожний з рівнів.
8. Що є об'єктом захисту в операційній системі?
9. На чому базується захист оперативної пам'яті в операційній системі?
10. Яким чином здійснюється захист даних та програм в операційній системі?
11. Навести категорії зломщиків захисту ОС, навести приклади кожній з категорій.
12. Які види атак можуть здійснювати зломщики на рівні операційних систем в цілому?
13. Яким чином може здійснюватися "крадіжка пароля"?
14. Що означає така хакерська атака, як відмова в обслуговуванні?

15. Що означає термін “збір сміття”?

16. Для чого хакер може використовувати сканування жорстких дисків?

## Лекція 3

# ХАРАКТЕРИСТИКА НАЙПОШИРЕНІШИХ ЗАГРОЗ БЕЗПЕЦІ КОМП'ЮТЕРНИХ СИСТЕМ

**План лекції:**

1. Несанкціонований доступ.
2. Незаконне використання привілеїв.
3. Атаки "салямі" (salami attack).
4. Приховані канали (convert channels).
5. "Маскарад" (masquerade).
6. "Збір сміття".
7. "Зламування системи" (break-in).
8. Шкідливе програмне забезпечення .
9. Утиліти схованого адміністрування (backdoor).
10. Intended-віруси.
11. Конструктори вирусів.
12. Поліморфні генератори.

Розглянемо наслідки, до яких може привести реалізація загроз і наведемо рекомендації щодо захисту від них. Кожна з розглянутих нижче загроз знаходить своє місце у проведеній класифікації загроз безпеці КС.

### 1. Несанкціонований доступ (unauthorized access) - 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 КС. 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 КС, від організації політики безпеки, від можливостей встановлених засобів захисту. НСД стає можливим через непродуманий вибір засобів захисту, їх некоректне

встановлення і настройку, 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. 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 КС взагалі і системою привілеїв зокрема.

**3. Атаки "салямі" (salami attack)**

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.

### 4. Приховані канали (convert channels)

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.

## 5. “Маскарад” (masquerade)

Під **“маскарадом”** (симуляція, моделювання), розуміється виконання яких-небудь дій одним користувачем КС від імені іншого, тобто права і привілеї одного користувача КС 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.

**“Маскарад”** – досить серйозне порушення, яке може призвести до тяжких наслідків (зміна конфігурації системи, відтік інформації, порушення роботи ОС).

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”.

### 6. “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”** (disk scavenging, garbage collection).

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 (erasure pattern)** – 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 (highewater marking)** – робить неможливим читання ділянок пам’яті, відведених для процесу записування, але не використаних ним. Верхня межа пам’яті, яка використовується, 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.

### 7. “Зламування системи” (break-in)

Це навмисне проникнення в КС 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 КС від зламування несе програма входу. Алгоритм вводу імені і пароля, їх шифрування, правила зберігання і заміни паролів не повинні мати помилок.

### 8. Шкідливе програмне забезпечення

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

**Шкідливі програми (malicious software або malware)** – 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.

1. 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 (trapdoor).*** **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 КС. 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 КС з використанням даної програми).

Програмні помилки не є люками. Люк – це механізм налагодження для підтримки і корегування програм. Якщо ж люки використовуються для отримання несанкціонованого доступу, то вони стають загрозою.

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

Прикладом люку може слугувати випадок, коли при розробці системи Multics, випробування на проникнення в яку проводилось групою "Tiger team" (команда тигрів) ВПС США, що зображала противника. Один з тактичних ходів полягав в тому, щоб відправити на вузол, працюючий під керуванням Miltics, підроблену (рос. "подложную") оновлену версію ОС. Версія містила в собі троянського коня, якого можна було активізувати за допомогою люка, і який дозволив команді отримати доступ до системи. Загроза була реалізована настільки добре, що розробники системи Multics не змогли віднайти її навіть тоді, коли вже знали про її наявність.

***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.

Другий приклад. В бібліотечній системі графства Монтгомері (Меріленд) підрядчик, якому доручили розробку комп'ютеризованої абонентської мережі, розмістив в ній логічну бомбу. При настанні певної дати ця бомба могла вивести систему із ладу, якщо замовник відмовлявся платити. Коли ж бібліотека затримала виплату грошей, підрядчик зізнався в існуванні "бомби" і пригрозив, що в разі неперерахування йому грошей він дасть "бомбі" спрацювати.

***Троянські коні (Trojan Horse).*** 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** – це програма, яка виконує на доповнення до основних (проектних і документованих) додаткові, але не описані в документацп, дії. 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.

Аналогія зі старогрецьким троянським конем, отже, виправдана – і в тому, і в іншому випадку в оболонці, яка не викликає ніякої підозри, існує загроза. Програми такого типу є серйозною загрозою для безпеки комп'ютерних систем.

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 ніби-то 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 троянського 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 Морріса (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.

Отже, як віруси троянські коні і черв'яки на сьогоднішній день є однією із найнебезпечніших загроз комп'ютерній системі. Для захисту від цих різновидностей шкідливих програм необхідно створювати замкнуте середовище виконання програм, розмежовувати доступ до виконуваних файлів, контролювати цілісність виконуваних файлів і системних областей, тестувати придбані програмні засоби.

***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 (greedy program).*** **"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 надходять із зовнішнього носія. Врешті-решт пам'ять може бути сконцентрована в одній програмі, і виконання інших стане не-

можливим.

***Захоплювачі паролів (password grabber).*** **Захоплювачі паролів** - це спеціально призначені 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 пересилаються власнику програми-захоплювача, після чого виводиться повідомлення про помилку введення і управління повертається операційній системі. Користувач думає, що зробив помилку 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.

### 9. Утиліти схованого адміністрування (backdoor)

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.

### 10. 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.

## 11. Конструктори вірусів

Конструктор вірусів - це утиліта, призначена для виготовлення нових комп'ютерних вірусів. Відомі конструктори вірусів для DOS, Windows і макро-вірусів. 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.

Інші конструктори (PS-MPC, G2) не мають інтерфейсу і зчитують інформацію про тип вірусу з конфігураційного файла.

## 12. 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 питання до Лекції 3**

1. У чому полягає несанкціонований доступ і як він реалізується?
2. Охарактеризуйте незаконне використання привілеїв як одну з поширених загроз комп'ютерній системі.
3. Що таке атаки "салямі" і за яких умов вони можуть відбутися?
4. Що таке "приховані канали", які їх види ви знаєте? В чому полягає принцип їх функціонування?
5. Що таке "маскарад" і як можна запобігти таким атакам?
6. У чому "збір сміття" можна використати як атаку на комп'ютерну систему, які механізми використовуються для захисту від них?
7. У чому полягає "зламування" системи і яким чином воно може бути реалізовано?
8. Що відносять до шкідливого програмного забезпечення і як його класифікують?
9. Що таке люки, звідки вони з'являються і як запобігти їх виникненню?
10. Охарактеризуйте логічні бомби як шкідливе програмне

забезпечення. Наведіть приклади.

11. Яке шкідливе програмне забезпечення відносять до програм-“троянів” і в чому їх особливості?

12. Охарактеризуйте коротко черв’яки, “зомбі” та “жадібні програми” як шкідливе програмне забезпечення.

13. Що являють собою програми-захоплювачі паролів і як можна запобігти їх шкідливому функціонуванню?

14. У чому суть утиліт прихованого адміністрування? Наведіть приклади відомих програмних продуктів цього напрямку.

15. Для чого існують і як функціонують конструктори вірусів і поліморфік-генератори?

## Лекція 4

# ВІРУСИ ЯК ШКІДЛИВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

**План лекції:**

1.Класифікація комп'ютерних вірусів.
2.Систематизація комп'ютерних вірусів.
3.Файлові віруси.
4.Завантажувальні (бутові) віруси.
5.Макро-віруси.
6.Мережеві віруси.
7.Стелс-віруси.
8.Поліморфік-віруси.
9.Способи захисту від вірусів.

## 1. Класифікація комп'ютерних вірусів

Віруси можна розділити на класи за такими ознаками:

- за середовище існування;
- за способом зараження;
- за особливостями використовуваних алгоритмів;
- за деструктивними можливостями.

***Віруси за середовищем їх існування.*** За цією ознакою віруси поділяються на:

- файлові віруси;
- завантажувальні віруси;
- макро-віруси;
- мережні віруси.

**Файлові віруси** діють одним з таких способів:

- впроваджуються в основному у виконувані файли, тобто у файли з розширеннями COM та EXE. Вони можуть впроваджуватись і у файли інших типів, але в такому випадку, як правило, вони ніколи не отримують управління, і, як наслідок, втрачають здатність до розмноження;
- створюють файли-двійники (компаньйон-віруси);
- використовують особливості організації файлової

системи (Link-віруси).

**Бутовські (завантажувальні) віруси** діють такими методами:

- впроваджуються в завантажувальний сектор диска (надалі Boot-сектор або бут-сектор);
- впроваджуються в сектор, що містить програму завантаження системного диска (Master Boot Record);
- змінюють покажчик на активний boot-сектор.

Існують файлово-бутівські віруси, що заражають і файли, і завантажувальні сектори дисків.

**Макро-віруси** заражають файли-документи і електронні таблиці відомих програмних продуктів.

**Мережні віруси** використовують для свого розповсюдження протоколи або команди комп'ютерних мереж та електронної пошти.

***Віруси за способом зараження***. **Резидентні віруси** при зараженні (інфікуванні) комп'ютера залишають в оперативній пам'яті свою резидентну частину, яка потім перехоплює звернення операційної системи до об'єктів зараження (файлів, завантажувальних секторів тощо) і впроваджується в них. Резидентні віруси знаходяться в пам'яті і є активними до самого вимкнення комп'ютера або до його перевантаження.

**Нерезидентні віруси** не заражають пам'ять комп'ютера і є активними лише обмежений проміжок часу. Деякі віруси лишають в пам'яті невеликі резидентні програми, які не розповсюджують вірус. Такі віруси вважаються нерезидентними.

***Віруси за особливостями використовуваних алгоритмів.*** **Прості віруси** – це віруси-паразити, вони змінюють вміст файлів і секторів дисків і можуть бути досить легко виявлені та знешкоджені.

**Стелс-віруси** (або віруси-невидимки) – це віруси, які повністю або частково приховують себе в системі. Найбільш розповсюдженим стелс-алгоритмом є перехоплення запитів ОС на читання-записування заражених об'єктів. Стелс-віруси при цьому або тимчасово виліковують їх, або підставляють замість

себе незаражені ділянки інформації. Це можуть бути і макро-віруси, які можуть забороняти виклики меню перегляду макросів. Це можуть бути і файлові (наприклад, вірус “Frodo”), і бутовські (вірус “Brain”) стелс-віруси.

Віруси-мутанти, які можуть використовувати алгоритми шифровки-розшифровки та поліморфізму, завдяки яким копії одного і того самого вірусу не мають жодного ланцюжка байтів, що повторюються. Поліморфік-віруси досить важко виявити, вони не мають сигнатури, тобто не містять жодної сталої ділянки коду.

***Віруси за деструктивними можливостями***. Опис деструктивних можливостей вірусу можна представити у такому розрізі:

- **нешкідливі віруси** ніяким чином не впливають на роботу комп’ютера, крім зменшення вільної пам’яті на диску в результаті свого поширення;

- **безпечні віруси** вплив вірусів обмежується зменшенням вільної пам’яті на диску і графічними, звуковими та іншими ефектами;

- **небезпечні віруси** можуть призводити до серйозних збійних ситуацій у роботі комп’ютера;

- **дуже небезпечні віруси** можуть призводити до втрати програми, знищення даних, стирання необхідної для роботи комп’ютера інформації, яка записана в системних областях пам’яті, і навіть сприяти прискореному зносу рухомих частин механізмів, наприклад, головок вінчестера.

## 2. Систематизація комп’ютерних вірусів

Важливість і довгостроковий характер проблеми захисту від комп’ютерних вірусів на практиці не викликає сумніву. Для зручності ідентифікації було би зручно, щоб кожний вірус мав своє ім’я. На превеликий жаль, старі вірусі у більшості випадків мають чимало назв, які виникли історично, а нові віруси, навпаки, таких назв взагалі не мають.

Відомим дослідником комп’ютерних вірусів М.М. Безрукавим було вироблено і запропоновано схему

класифікації, яка включає три основні елементи:

- **класифікаційний код вірусу;**
- **дескриптор вірусу** (формалізований список основних властивостей);
- **сигнатура вірусу** (рядок для контекстного пошуку даного вірусу в зараженій програмі).

***Класифікаційний код вірусу***. Кожному вірусу присвоюється код, який складається з літерного префікса, кількісної характеристики і факультативного літерного суфікса.

Наприклад, в коді RCE-1813c є такі складові: RCE - префікс, 1813 - корінь (характеристика), а c - суфікс. Крім того, факультативне розширення, що записується в кінці коду через крапку, характеризує групу, до якої належить даний вірус. Наприклад, RCE-1813.IER означає, що даний вірус належить до єрусалимської групи.

Головною вимогою до класифікаційного коду вірусу є можливість визначення більшості вхідних його властивостей на незараженому комп'ютері. Виконання будь-яких дій з дослідження вірусу на зараженому комп'ютері є найбільшою й найбільш розповсюдженою помилкою, якої припускаються недосвідчені користувачі. Необхідно підкреслити, що будь-які дії на комп'ютері, зараженому невідомим вірусом, пов'язані з певним ризиком викликати спрацьовування троянської компоненти вірусу. Крім того, резидентний вірус з метою маскування може перехоплювати запити і перекручувати інформацію, яка видається. Нині відомі віруси, що мають таку властивість. Наприклад, група файлових вірусів, відома під назвою TP-вірусів, починаючи з вірусу TP-34 (члени цієї групи мають номери, які зберігаються в передостанньому байті вірусу в шістнадцятковому вигляді), має властивість "самовикусування" при спробі трасувати заражену програму – резидентний вірус виконує "викусування" вірусу з програми, "підсовуючи" налагоджувачу вже вилікувану програму. Так само бутові віруси, які входять у пакистанську групу (віруси “Brain”, “Ashar”), при спробі переглянути бут-сектор на зараженому комп'ютері "підсовують" користувачу оригінальний

бут-сектор, який зберігається вірусом в одному із секторів, позначеному як дефектний (і, тим самим, вилученому з розподілу у файли).

**Літерний префікс** вказує на місце розміщення голови вірусу і складається з послідовності літер і цифр, що починається з великої літери. Відповідно до цього будемо розрізняти такі типи вірусів (розглядатимемо тільки реально існуючі типи, а не всі принципово можливі):

- **файлові** – коли голова вірусу розміщується в COM-, EXE-файлах і оверлеях (символи C, E в префіксі). При цьому додаткову літеру, яка відображає зараження оверлеїв, у префікс не вводиться, щоб запобігти його ускладненню, а виноситься у дескриптор;

- **бутові** – коли голова вірусу розміщується в бут-секторі або блоці MBR (символи B, R або M у префіксі);

- **пакетні** – коли голова вірусу розміщена в пакетному файлі, тобто являє собою фрагмент або програму на будь-якій мові програмування (префікс J).

Поряд із "чистими" вірусами, які використовують лише одне середовище, нині з'явились "гібридні" – комбінація файлових і бутових вірусів. У таких вірусах замість першої літери R використовують відповідну літеру префікса бутового вірусу, наприклад BCE або MCE (як і бутові, змішані віруси не можуть бути нерезидентними).

Характеристика вірусу являє собою кількісно вимірювану властивість вірусу, яка допускає просте визначення і розрізняється для більшості типів вірусів. Наприклад, для файлових вірусів як характеристика може використовуватися величина приросту довжини файлів при зараженні ("інфекційна довжина"), хоч тут є певні складності.

Суфікс використовується, коли два різних віруси або два штами одного і того самого вірусу мають однаковий префікс і характеристику. У цьому випадку, для отримання унікального коду використовується як суфікс латинська літера. Наприклад, в коді RC-1704f літера f означає "штам-f".

***Дескриптор вірусу.*** Безумовно, запропонований код вірусу

не охоплює та й не може охопити основні властивості вірусу. Водночас систематизація властивостей вірусу становить значний інтерес як для розробників антивірусних програм, так і для їх користувачів, оскільки дозволяє інтегрувати різнорідні факти, які стосуються поведінки того чи іншого вірусу в системі, тим самим полегшуючи їх запам'ятовування і порівняння. Тому як другий елемент класифікації пропонується так званий дескриптор.

**Дескриптор** є систематизацією основних характеристик вірусу в закодованому вигляді. Кодування складається з груп символів, що починаються з великої латинської літери, за якою йдуть маленькі латинські літери або цифри. При цьому велика латинська літера визначає вид характеристики, а наступні за нею малі літери або цифри – значення характеристики для конкретного вірусу. Наприклад, в дескрипторі "Xab Yc Zdmt" є три властивості: X – зі значенням "ab", Y – зі значенням "c" і Z – зі значенням "dmt".

***Сигнатура вірусу***. Оскільки більшість відомих нині вірусів допускають детектування за допомогою контекстного пошуку, то однією з важливих задач класифікації є складання сигнатур. **Сигнатура** – це список рядків для контекстного пошуку. Знання сигнатур дозволяє перевіряти нове програмне забезпечення на їх наявність вірусів, тим самим суттєво підвищуючи ступінь захищеності ЕОМ. Стандартизація сигнатур особливо важлива, коли вірус має багато штамів, оскільки формальні схеми, подібні описаним вище класифікаційному коду і дескриптору, мають той недолік, що деякі штами не будуть розрізнятися в заданому просторі ознак. Водночас порівняно легко забезпечується унікальність сигнатури, у крайньому разі для більшості відомих вірусів, хоч існують віруси, які не вміщують жодної постійної сигнатури, тобто які не можна знайти за допомогою контекстного пошуку.

Сигнатури вірусів можуть бути представлені як текстовими рядками або регулярними виразами, так і ділянками програмного коду.

Хоч далі частіше в якості сигнатури використовуються

тільки текстові рядки, для них застосовуються і регулярні вирази. Вони суттєво стійкіші до деяких мутацій і, крім того, при меншій довжині забезпечують кращу якість розпізнавання (менша кількість невірних спрацьовувань). Все це робить їх кращими за прості текстові рядки.

Очевидно, що сигнатура, яка відповідає ділянці з командами, надійніша за сигнатури ділянки з даними, наприклад з текстовими рядками (останні можуть бути модифіковані). Тому вибір сигнатури доцільно робити на основі аналізу дизасембльованого коду вірусу.

Довжина сигнатури не повинна бути дуже великою, оскільки довгу сигнатуру важко вірно набрати вручну. В той же час недостатня довжина або вибір нехарактерних ділянок коду сигнатури викликатимуть багато неправильних спрацювань, що зовсім небажано. Правильна сигнатура не повинна бути в жодній з найбільш розповсюджених в операційних системах службових програм, включаючи, безумовно, самі компоненти операційної системи. Отже, для вибору сигнатури відповідно до вказаних вимог необхідно провести ряд експериментів, де самі сигнатури можуть бути предметом порівняння і аналізу.

На сьогоднішній день є програми, які забезпечують детектування вірусів шляхом пошуку в файлах відповідних рядків, і використані в них сигнатури природно "прийняти за основу". Найбільшу цінність становлять рядки, які використовуються у відомому закордонному детекторі Scan, оскільки нові версії цього детектора з'являються регулярно і охоплюють практично всі віруси, які з'являються за кордоном. З інших закордонних детекторів необхідно відзначити Virscan фірми IBM і TNTVirus фірми Carnel (Ізраїль). Для визначеності називатимемо рядок, який використовується детектором Scan, M-сигнатурою, рядок, який використовується Virscan, – I-сигнатурою, а рядок, який використовується TNTVirus, – C-сигнатурою.

Водночас відзначимо, що сигнатур для ряду вірусів, розроблених у нашій країні, в існуючих версіях цих програм немає, а сигнатури для болгарських вірусів часто невдалі. У таких

випадках використовуються сигнатури, які позначені буквою B (B-сигнатури), або так звані J-сигнатури. Останні являють собою початкові байти коду вірусу, тобто перші виконувані команди тіла вірусу. Досвід показує, що вони досить специфічні і в більшості випадків дозволяють відрізнити один вірус від іншого. При цьому для файлових вірусів, які дописують своє тіло в кінець файла, вважають, що J-сигнатура починається з байта, на який передає керування команда JMP. Крім того, в тілі деяких вірусів зустрічаються характерні текстові рядки. Такі рядки називатимемо T-сигнатурами і використовуватимемо як допоміжні.

Необхідно відзначити, що контекстний пошук може використовуватися не тільки для пошуку заражених вірусом програм, але й для пошуку програм і файлів, які знищені або пошкоджені вірусом. Наприклад, вірус C-648.VEN при певних значеннях таймера замість зараження програми знищує її, записуючи в перші п'ять байтів рядок, який відповідає переходу на підпрограму перезавантаження BIOS. Для пошуку знищених вірусом програм можна використовувати рядок "EAFOFFOOFO". Аналогічно вірус RCE-1800.DAV знищує сектори на вінчестері, записуючи перші байти повідомлення "Eddie lives … somewhere in time". За цим повідомленням за допомогою Norton Utilites або інших програм можна виявити пошкоджені сектори і визначити, до яких файлів вони належать.

За наявності сигнатури перевірку зараженості файлів вірусом даного типу зручно виконувати, використовуючи спеціальні програми. Наприклад, вдалою є програма TBScan, яка здійснює пошук у каталогах або заданих його гілках. У випадку виявлення заражених програм доцільно додатково проконтролювати результати за допомогою, наприклад, Norton Utilites, оскільки для перегляду всіх файлів можна використовувати режим глобального пошуку на диску.

### 3. Файлові віруси

До даної групи відносяться віруси, що при своєму розмноженні тим чи іншим способом використовують файлову

систему якої-небудь ОС.

Впровадження файлового вірусу можливе практично в усі виконувані файли усіх популярних операційних систем. На сьогоднішній день відомі віруси, що вражають всі типи виконуваних об'єктів: командні файли (BAT), драйвери (SYS, у тому числі спеціальні файли IO.SYS і MSDOS.SYS) і виконувані двійкові файли (EXE, COM). Існують віруси, що вражають файли інших операційних систем - Windows, OS/2, Macintosh, UNIX, включаючи VxD-драйвери Windows.

Існують віруси, які заражають файли, що містять вихідні тексти програм, бібліотечні чи об'єктні модулі. Можливий запис вірусу й у файли даних, але це може бути або в результаті помилки вірусу, або при прояві його агресивних властивостей. Макро-віруси також записують свій код у файли даних або у документи та електронні таблиці, однак ці віруси настільки специфічні, що винесені в окрему групу.

Файлові віруси є найпоширенішим типом комп'ютерних вірусів. Вони становлять близько 80% загальної кількості вірусів, відомих для комп'ютерів, які сумісні з IBM PC. Цей клас комп'ютерних вірусів має дуже високу інфікуючу спроможність. За відсутності протидії вони викликають справжні епідемії. Так, наприклад, відбулося з вірусом RCE-1813.IER, відомого також під назвами Jerusalim (Єрусалим), Black Friday (Чорна п'ятниця).

Більшість розповсюджених файлових вірусів мають штами, які не дуже відрізняються від базової версії. Тому можна говорити про групи файлових вірусів і, відповідно, групові дескриптори і групові сигнатури. Нині кількість виявлених у країнах СНД файлових вірусів перевищує кілька сотень, тому запам'ятовування їх класифікаційних кодів суттєво полегшується, якщо вони використовуються з розширенням, яке показує, до якої групи належить даний вірус.

***Класифікаційний код файлового вірусу***. Файлові віруси можна розділити на резидентні і нерезидентні, оскільки це в багатьох випадках визначає поведінку вірусу і насамперед його інфікуючу спроможність (резидентні віруси мають значно вищу інфікуючу спроможність порівняно з нерезидентними).

Класифікаційний код файлових резидентних вірусів починається з префікса R, наприклад R-1701.CAS.

***Префікс файлового вірусa.*** Крім символу R, класифікаційний код файлового вірусу може включати символи C і E або їх комбінацію. Як уже зазначалося, символи C і E визначають типи файлів, заражених даним вірусом. Наприклад, якщо резидентний вірус заражає COM- і EXE-файли, то його класифікаційний код матиме префікс RCE.

***Кількісна характеристика***. У якості кількісної характеристики можна використовувати:

- нормований приріст або інфективну довжину (infective length);
- довжину коду вірусу;
- приріст довжини будь-якого зараженого файла.

До об'єктивних властивостей файлових вірусів, що безпосередньо спостерігаються, можна віднести насамперед приріст довжини файлів при зараженні. Цей приріст, який зумовлює наявність вірусу, можна використати для визначення його типу. Тут є дві основні проблеми. По-перше, величина приросту може варіюватися залежно від довжини зараженого файла (багато вірусів при дописуванні свого коду в кінець зараженого файла вирівнює своє тіло на найближчу адресу, кратну 16, тобто на межу параграфа). По-друге, величина приросту може не збігатися для COM- і EXE-файлів. Тому як кількісну характеристику частіше використовують нормований приріст – інфективну довжину, яку визначається за такими правилами:

- для вірусів з префіксом C і CE (RC, RKCE) характеристика класифікаційного коду має дорівнювати мінімальному приросту довжини COM-файла (для вірусів типу C і CE) або EXE-файла (для вірусів типу E);
- для вірусів, які не змінюють довжину файла, вказується нуль, а через дефіс дійсна довжина тіла вірусу, наприклад RC-O-346.LEH;
- для вірусів, які маскують збільшення довжини файла на зараженій програмі, до характеристики, визначеної за першим правилом, зліва додається незначущий нуль (наприклад,

RCE-02000. DAV).

Відзначимо, що запропонований першим правилом підхід дозволяє зняти вплив вирівнювання на межу параграфа для вірусів, які вирівнюють свій приріст вказаним способом. Крім того, для вірусів, які змінюють свій приріст визначеним способом, наприклад шляхом підгонки до величини, кратної 51, мінімальний приріст також дає можливість позбутися впливу вставних байтів (цей випадок можна розглядати як різновидність вирівнювання). І нарешті, для вірусів, які багато разів заражують один і той самий файл, використання мінімального приросту дозволяє звільнитися від впливу багаторазового зараження.

Для визначення інфективної довжини не треба буде проводити спеціальні експерименти із зараження файлів. Здебільшого її можна досить просто визначити, порівнявши прирости довжин двох або більше заражених файлів типу COM. Найчастіше файлові віруси заражають командний процесор MS-DOS (файл COMMAND.COM) і програми, назви яких знаходяться у файлі AUTOEXEC.BAT. При аналізі кількох заражених файлів можливі два найтиповіші (хоч і не єдино можливі) випадки.

Якщо прирости довжин двох або більше заражених файлів збігаються, а залишки від ділення довжин початкових файлів на 16 відрізняються один від одного, то, ймовірно, вірус не виконує вирівнювання свого коду на межу параграфа й інфективну довжину L даного вірусу можна дістати за формулою:

L=D- 16-mod (LEN, 16)), тобто відніманням із отриманого приросту D доповнення 16 залишку від ділення початкової довжини LEN файла на 16. Наприклад, файл COMMAND.COM, який файлові віруси здебільшого пошкоджують в числі перших, у найпоширеніших нині версіях M8-DOS має довжину 25307. При цьому залишок від ділення 25307 на 16 дорівнює 11 (mod(25307,16)=11). Очевидно, що доповнення до 16 дорівнює 5, і для вирівнювання на межу параграфа необхідна вставка п'яти додаткових байтів. У цьому випадку інфективна довжина буде на 5 менша, ніж приріст довжини файла COMMAND.COM. Перевагою прийнятого підходу є те, що, за окремим винятком

(наприклад, вірус RCE-1813.IER), визначена таким чином інфективна довжина збігається з довжиною коду вірусу.

Як кількісна характеристика класифікаційного коду можуть застосовуватися й інші параметри. Найпоширенішими вважають такі два підходи.

Використання як кількісної характеристики довжини коду вірусу, визначеної за константою, яка вміщується у фрагменті, що забезпечує дописування коду вірусу в заражений файл (цю константу можна порівняно легко визначити, аналізуючи дизасембльований код вірусу). Така характеристика є об'єктивною, тому її часто використовують розробники антивірусних програм, які досить добре володіють мовою асемблера. Але визначена так характеристика в ряді випадків не збігається зі значенням приросту довжин файлів, який спостерігається. Це знижує її цінність з погляду використання при спробі класифікації користувачем, який не володіє мовою асемблера, нового, ще невідомого йому вірусу. Наприклад, для згаданого вище єрусалимського вірусу довжина коду вірусу становить 1808 байтів, а приріст довжини при зараженні файлів типу COM - 1813 байтів, що пояснюється додатковим записуванням в кінець зараженого файла типу COM п'ятибайтної константи "Ms-Dos" (використовується як ознака зараженості файла).

Використання як кількісної характеристики приросту довжини якого-небудь конкретного файла, отриманого в результаті його зараження. Цей дійсно зручний підхід втратив свою привабливість з появою ряду вірусів, які не заражають командний процесор, з розповсюдженням MS-DOS версій 4.0 і вище, в якій довжина файла COMMAND.COM становить 37637, з появою нових сучасних операційних систем.

***Дескриптор файлового вірусу.*** Для зручності сприйняття дескриптор вірусу розбивається на декілька складових, для яких використовуються такі позначення:

DM – головний дескриптор;

DP – предикат зараження (для зручності сприйняття записаний в близькій до алгебраїчної нотації);

DR (тільки для резидентних вірусів) – положення в оперативній пам'яті, реакція на "тепле" перезавантаження і розмір зайнятої пам'яті;

DH – перехоплювані переривання (в шістнадцятковій системі числення).

***Сигнатура файлового вірусу***. Як уже відзначалося, для сигнатур доцільно використовувати рядки в шістнадцятковій системі числення, які відповідають характерним послідовностям команд у тілі вірусу. Розміщення сигнатур підпорядковується такому правилу: якщо M-сигнатура входить у V-сигнатуру, то вона додається після V-сигнатури. Як уже відзначалося раніше, T-сигнатури існують не для всіх файлових вірусів. Однією із найзручніших сигнатур для файлових вірусів є J-сигнатура. Їх можна дуже швидко визначити за допомогою будь-якого налагоджувача (Debug, Turbo Debugger, AFD і т.д.). Користувачі, які не вміють працювати з налагоджувачами, можуть використовувати для визначення J-сигнатур програму "маскошукач", яка входить в пакет VL (непоганий детектор, заснований на контекстному пошуку заданих рядків).

Необхідно відзначити, що контекстний пошук можна використовувати не тільки для пошуку заражених вірусом програм, але й для пошуку програм і файлів, які знищені або пошкоджені вірусом. Наприклад, вірус C-648.VEN при певних значеннях таймера замість зараження програми знищує її, записуючи в перші 5 байтів рядок, який відповідає переходу на підпрограму перезавантаження BIOS.

**Види файлових вірусів**

За способом зараження файлів віруси поділяються на:

- overwriting-віруси;
- паразитичні ("parasitic");
- компаньйони-віруси ("companion");
- link-віруси;
- віруси-хробаки;
- віруси, що заражають об'єктні модулі (OBJ), бібліотеки компіляторів (LIB) і вихідні тексти програм.

***Overwriting-віруси***. Даний метод зараження є найбільш

простий: вірус записує свій код замість коду файла, що заражається, знищуючи його вміст. Природно, що при цьому файл перестає працювати і не відновлюється. Такі віруси дуже швидко виявляють себе, оскільки операційна система і додатки досить швидко перестають працювати.

До різновиду overwriting-вірусів відносяться віруси, що записуються замість заголовка EXE-файлів. Основна частина файла при цьому залишається без змін і продовжує нормально працювати у відповідній операційній системі, однак заголовок виявляється зіпсованим.

***Parasitic-віруси***. До паразитичних відносяться усі файлові віруси, що при поширенні своїх копій обов'язково змінюють вміст файлів, залишаючи самі файли при цьому цілком чи частково працездатними. Паразитичні віруси розділяються на три типи в залежності від того, куди вони записують своє тіло.

1. Впровадження вірусу в початок файла ("prepending"-віруси) може відбуватися двома способами:

- вірус переписує початок файла, що заражається, у його кінець, а сам копіюється в місце, що звільнилося;
- вірус створює в оперативній пам'яті свою копію, дописує до неї файл, що заражається, і зберігає отриману конкатенацію на диск.

Деякі віруси при цьому дописують у кінець файла блок додаткової інформації (наприклад, вірус "Jerusalem" по цьому блоку відрізняє заражені файли від незаражених).

Впровадження вірусу в початок файла застосовується в переважній більшості випадків при зараженні BAT- і COM-файлів MS DOS. Відомо кілька вірусів, що записують себе в початок EXE-файлів операційних систем DOS, Windows і навіть Linux. При цьому віруси для збереження працездатності програми або лікують заражений файл, повторно запускають його, чекають закінчення його роботи і знову записуються в його початок (іноді для цього використовується тимчасовий файл, у який записується знешкоджений файл), або відновлюють код програми в пам'яті комп'ютера і надбудовують необхідні адреси в її тілі (тобто дублюють роботу ОС).

2. Впровадження вірусу в кінець файла ("appending"-віруси) – це найбільш розповсюджений спосіб упровадження вірусу у файл. При цьому вірус змінює початок файла таким чином, що першими виконуваними командами програми є команди вірусу:

а) у COM-файлі в більшості випадків це досягається зміною його перших трьох (чи більш) байтів на коди інструкції JMP Loc_Virus (чи в більш загальному випадку – на коди програми, що передає керування на тіло вірусу);

б) у заголовку EXE-файла змінюються значення стартової адреси (CSIP), кількість секцій у файлі, характеристики секцій, довжина виконуваного модуля (файла), рідше – регістри-покажчики на стек (SSSP), контрольна сума файла і т.д.;

в) у виконуваних файлах Windows і OS/2 (NewEXE – NE, PE, LE, LX) змінюються поля в NewEXE-заголовку. Структура цього заголовка значно складніша за заголовок DOS EXE-файлів, тому зміні підлягає більше число полів – значення стартової адреси, кількість секцій у файлі, характеристики секцій і т.д. Додатково до цього довжини файлів перед зараженням можуть збільшуватися до значення, кратного параграфу (16 байт) в DOS або секції в Windows і OS/2 (розмір секції залежить від параметрів заголовка EXE-файла);

г) віруси, що впроваджуються у SYS-файли, приписують свої коди до тіла файла і модифікують адреси програм стратегії (Strategy) і переривання (Interrupt) драйвера, що заражається (зустрічаються віруси, що змінюють адресу тільки однієї з цих програм). При ініціюванні зараженого драйвера вірус перехоплює відповідний запит ОС, передає його драйверу, чекає відповіді на цей запит, коректує його і залишається разом із драйвером в одному блоці оперативної пам'яті. Такий вірус може бути надзвичайно небезпечним і живучим, він впроваджується в оперативну пам'ять при завантаженні ОС раніш за будь-яку антивірусну програму, якщо вона теж не є драйвером. Але існують віруси, які заражають системні драйвери іншим способом: вірус модифікує заголовок драйвера так, що DOS розглядає інфікований файл як ланцюжок з двох (або більше)

драйверів.

3. Впровадження вірусу в середину файла ("inserting"-віруси) може здійснюватись кількома методами:

а) вірус переносить частину файла в його кінець або "розсовує" файл і записує свій код у простір, що звільнився. Цей спосіб багато в чому аналогічний методам, перерахованим вище. Деякі віруси при цьому компресують перенесений блок файла так, що довжина файла при зараженні не змінюється (вірус "Mutant");

б) метод "cavity", при якому вірус записується у свідомо невикористовувані області файла. Вірус може бути скопійований у незадіяні області таблиці настроювання адреси EXE-файла (вірус "BootExe"), у заголовок EXE-файла, в область стека файла COMMAND.COM, в область текстових повідомлень популярних компіляторів. Існують віруси, що заражають тільки ті файли, що містять блоки, заповнені яким-небудь постійним байтом, при цьому вірус записує свій код замість такого блоку;

в) копіювання вірусу в середину файла може відбутися в результаті помилки вірусу, у цьому випадку файл може бути зіпсований.

4. Віруси без точки входу. Окремо слід зазначити досить незначну групу вірусів, що не має "точки входу" (EPO-віруси - Entry Point Obscuring viruses). До них відносяться віруси, що не записують команд передачі керування в заголовок COM-файлів (JMP) і не змінюють адресу точки старту в заголовку EXE-файлів. Такі віруси записують команду переходу на свій код у будь-яке місце в середину файла й одержують керування не безпосередньо під час запуску зараженого файла, а під час виклику процедури, що містить код передачі керування на тіло вірусу. Причому виконуватися ця процедура може вкрай рідко (наприклад, при виведенні повідомлення про якусь специфічну помилку). Тому вірус може довгі роки "спати" усередині файла і “вискочити на волю” тільки при деяких обмежених умовах.

Перед тим, як записати в середину файла команду переходу на свій код, вірусу необхідно вибрати "правильну" адресу у файлі – інакше заражений файл може виявитися

зіпсованим. Відомі кілька способів, за допомогою яких віруси визначають такі адреси усередині файлів.

Перший спосіб – пошук у файлі послідовності стандартного коду (віруси "Lucretia", "Zhengxi"). Ці віруси шукають у заражуваних файлах стандартні заголовки процедур C/Pascal і пишуть замість них свій код.

Другий спосіб – трасування чи дизасемблювання коду файла ("CNTV", "MidInfector", "NexivDer"). Такі віруси завантажують файл у пам'ять, потім трасують чи дизасемблюють його й у залежності від різних умов вибирають команду (чи команди), замість яких записується код переходу на тіло вірусу.

Третій спосіб застосовується тільки резидентними вірусами – при запуску файла вони контролюють будь-яке переривання (частіше – INT 21h). Як тільки файл, що тільки заражається, викликає це переривання, вірус записує свій код замість команди виклику переривання (віруси "Avatar.Positron", "Markiz").

Четвертий спосіб базується на так званих настроюваннях коду програми. Таблиця настроювань (relocation table) в EXE-файлах вказує на адреси в тілі програми, які при завантаженні програми повинні бути приведені у відповідність до реальних адрес пам'яті. Зазвичай адреси, що надбудовуються, містять асемблерні інструкції з обмеженого набору. Вірус може легко ідентифікувати конкретну інструкцію, замінити її на виклик свого коду JMP_Virus і занулити відповідний запис у таблиці настроювань (щоб команда JMP_Virus не виявилася зіпсованою при завантаженні файла в пам'ять).

***Компаньйон-віруси.*** До категорії "компаньйон" відносяться віруси, які не змінюють файлів, що заражаються. Алгоритм роботи цих вірусів полягає в тому, що для файла, який заражається, створюється файл-двійник, причому при запуску зараженого файла керування одержує саме цей двійник, тобто вірус.

Найбільш поширені компаньйон-віруси, що використовують особливість DOS першим виконувати .COM-

файл, якщо в одному каталозі присутні два файли з тим самим ім'ям, але різними розширеннями імені – .COM і .EXE. Такі віруси створюють для EXE-файлів файли-супутники, що мають те ж саме ім'я, але з розширенням .COM, наприклад, для файла XCOPY.EXE створюється файл XCOPY.COM. Вірус записується в COM-файл і ніяк не змінює EXE-файл. При запуску такого файла DOS першим знайде і виконає COM-файл, тобто вірус, який потім запустить і EXE-файл. Деякі віруси використовують не тільки варіант COM-EXE, але також і BAT-COM-EXE.

Другу групу складають віруси, що при зараженні перейменовують файл у яке-небудь інше ім'я, запам'ятовують його (для наступного запуску файла-хазяїна) і записують свій код на диск під іменем файла, що заражається. Наприклад, XCOPY.EXE перейменовується в XCOPY.EXD, а вірус записується під ім'ям XCOPY.EXE. При запуску керування одержує код вірусу, що потім запускає оригінальний XCOPY, що зберігається під ім'ям XCOPY.EXD. Цікавий той факт, що даний метод працює, напевно, у всіх операційних системах: в DOS, в Windows і OS/2.

У третю групу входять так називані "Path-companion" віруси, що "грають" на особливостях PATH. Вони або записують свій код під іменем зараженого файла, але "вище" на один рівень PATH (ОС, таким чином, першим знайде і запустить файл-вірус), або переносять файл-жертву на один підкаталог вище і т.д.

Можливе існування й інших типи компаньйонів-вірусів, що використовують інші особливості інших операційних систем.

***Link-віруси.*** Link-віруси, як і компаньйон-віруси не змінюють фізичного вмісту файлів, однак під час запуску зараженого файла "змушують" ОС виконати свій код. Цієї мети вони досягають модифікацією необхідних полів файлової системи. На сьогоднішній день відомий єдиний тип Link-вірусів – віруси сімейства "Dir_II". При зараженні системи вони записують своє тіло в останній кластер логічного диска. При

зараженні файла віруси коректують лише номер першого кластера файла, розташований у відповідному секторі каталогу. Новий початковий кластер файла буде вказувати на кластер, що містить тіло вірусу. Отже, при зараженні файлів їх довжини і вміст кластерів диска, що містять ці файли, не змінюються, а на всі заражені файли на одному логічному диску буде приходитися тільки одна копія вірусу.

Таким чином, до зараження дані каталогу зберігають адресу першого кластера файла, а після зараження дані каталогу вказують на вірус, тобто при запуску файла керування одержують не самі файли, а вірус.

***Файлові хробаки.*** Файлові хробаки (worms) є, у деякому сенсі, різновидом компаньйон-вірусів, але при цьому ніяким чином не пов'язують свою присутність з якимось виконуваним файлом. При розмноженні вони лише копіюють свій код у які-небудь каталоги дисків у надії, що ці нові копії будуть колись запущені користувачем. Іноді ці віруси дають своїм копіям спеціальні імена, щоб підштовхнути користувача на запуск своєї копії, наприклад, INSTALL.EXE чи WINSTART.BAT.

Існують віруси-хробаки, що використовують досить незвичайні прийоми, наприклад, записують свої копії в архіви (ARJ, ZIP та ін.). До таких вірусів відносяться "ArjVirus" і "Winstart". Деякі віруси записують команду запуску зараженого файла в BAT-файли (наприклад, "Worm.Info").

Не слід плутати файлові віруси-хробаки з мережними хробаками. Перші використовують тільки файлові функції якої-небудь ОС, другі ж при своєму розмноженні користуються мережними протоколами.

***OBJ-, LIB-віруси і віруси у вихідних текстах.*** Віруси, що заражають бібліотеки компіляторів, об'єктні модулі і вихідні тексти програм досить екзотичні і практично не поширені. Усього їх біля десятка.

Віруси, що заражають OBJ- і LIB-файли, записують у них свій код у форматі об'єктного модуля чи бібліотеки. Заражений файл, таким чином, не є виконуваним і нездатний на подальше поширення вірусу у своєму поточному стані. Носієм же

"живого" вірусу стає COM- чи EXE-файл, одержуваний у процесі лінковки зараженого OBJ/LIB-файла з іншими об'єктними модулями і бібліотеками. Таким чином, вірус поширюється в два етапи: на першому заражаються OBJ/LIB-файли, на другому етапі (лінковка) виходить працездатний вірус.

Зараження вихідних текстів програм є логічним продовженням попереднього методу розмноження. При цьому вірус додає до вихідних текстів свій вихідний код (у цьому випадку вірус повинен містити його у своєму тілі) чи свій шістнадцятковий дамп (що технічно легше). Заражений файл здатний на подальше поширення вірусу тільки після компіляції і лінковки (віруси "SrcVir", "Urphin").

**Алгоритм роботи файлового вірусу**

Одержавши керування, вірус здійснює такі дії (приведений список найбільш загальних дій вірусу при його виконанні, і для конкретного вірусу список може бути доповнений, пункти можуть помінятися місцями і значно розширитися):

- резидентний вірус перевіряє оперативну пам'ять на наявність своєї копії і інфікує пам'ять комп'ютера, якщо копія вірусу не знайдена. Нерезидентный вірус шукає незаражені файли в поточному і\або кореневому каталозі, у каталогах, відзначених командою PATH, сканує дерево каталогів логічних дисків, а потім заражає виявлені файли;

- виконує, якщо вони є, додаткові функції, деструктивні дії, графічні чи звукові ефекти і т.д. Додаткові функції резидентного вірусу можуть викликатися через деякий час після активізації в залежності від поточного часу, конфігурації системи, внутрішніх лічильників чи вірусу інших умов; у цьому випадку вірус при активізації обробляє стан системного годинника, встановлює свої лічильники і т.д.;

- повертає керування основній програмі (якщо вона є). Паразитичні віруси при цьому або лікують файл, виконують його, а потім знову заражають, або відновлюють програму (але не файл) у вихідному виді (наприклад, у COM-програм відновлюються декілька перших байтів, у EXE-програми обчис-

люється справжня стартова адреса, у драйвера відновлюються значення адрес програм стратегії і переривання). Компаньйони-віруси запускають на виконання свого "хазяїна", віруси-хробаки і overwriting-віруси повертають керування ОС.

Метод відновлення програми у первісному вигляді залежить від способу зараження файла.

Якщо вірус впроваджується в початок файла, то він або зрушує коди зараженої програми на число байтів, рівне довжині вірусу, або переміщає частину коду програми з її кінця в початок, або відновлює файл на диску, а потім запускає його.

Якщо вірус записався в кінець файла, то при відновленні програми він використовує інформацію, збережену у своєму тілі при зараженні файла. Це може бути довжина файла, декілька байтів початку файла у випадку COM-файла або декілька байтів заголовка у випадку EXE-файла.

Якщо ж вірус записується в середину файла спеціальним чином, то при відновленні файла він використовує ще і спеціальні алгоритми.

**Приклади файлових вірусів**

**Abba.9849**. Безпечний резидентний вірус. Перехоплює INT 21h і записується в кінець COM- і EXE-файлів при їх запуску. Містить рядки

\COMMAND.COM

Program too big to fit in memory

\ABBAл|*.* E\ABBAл|

Створює на поточному диску файли ABBAл|nn з атрибутами HIDDEN і READONLY, 'nn' - число файлів, заражених на цьому диску. Це число збільшується при зараженні чергового файла – вірус перейменовує цей файл в ABBAл|(nn+1). Залежно від числа nn вірус проявляє себе якимсь відео-ефектом на відео карті Hercules.

**Lenin 943.** Безпечний нерезидентний вірус. При запуску шукає EXE-файли і записується в їх кінець. При зараженні не змінює значення регістрів в EXE-заголовку, а вставляє в точку входу у файл команду CALL FAR virus і коректує Relocation Table. Залежно від своїх лічильників виводить тексти

САМЫЙ! ЧЕЛОВЕЧНЫЙ! ЧЕЛОВЕК!

Ленин и сегодня живее всех живых держит мертвой хваткой упыря

Також містить рядки

*.EXE PATH=

**Metall.557**. Дуже небезпечний нерезидентний вірус. Шукає .COM-файлы окрім COMMAND.COM і записується в їх кінець. Коректно заражає тільки файли, на початку яких присутня команда JMP/CALL NEAR. Решта файлів після зараження виявляється зіпсованими. Залежно від системного таймера перемішує символи на екрані. Містить рядок: METALL\I

**Scorpion.2278.** Дуже небезпечний нерезидентний зашифрований вірус. При запуску заражає файл C\COMMAND.COM, потім шукає COM- і EXE-файли і записується в їх кінець. При зараженні COMMAND.COM записується в кінець файла в область стека COMMAND.COM і, таким чином, не збільшує його довжину. Знищує файли з ім'ям CHKLIST.MS. В деяких випадках також шукає інші файли і знищує їх. Залежно від системної дати і встановленого BIOS'a форматує вінчестер, виводить текст:

DEATH ON TWO LEGS V2.8

(c) BLACK SCORPiON, 1996

Written in Moscow

потім перехоплює INT 1Ch і програє мелодію. Вірус також містить рядки:

*.* *.EXE *.COM

C\COMMAND.COM

DEATH ON TWO LEGS WAS HERE

**Sisters.2221.** Дуже небезпечний резидентний зашифрований вірус. Перехоплює INT 21h, 16h і записується в кінець COM- і EXE-файлів при їх запуску. Знищує антивірусні файли даних CHKLIST.MS і CHKLIST.CPS.

Залежно від значень внутрішнього лічильника і поточної дати вірус відключає драйвер миші, стирає 40 секторів на диску C, видаляє CMOS пам'ять, завішує комп'ютер, виводить повідомлення:

TEMPLE OF LOVE V1.0 MS 95.

FoUnD VIRUS SYSTERS OF MERCY iN yOuR sYsTeM !!!

Вірусний обробник INT 16h (клавіатура) стирає CMOS-пам'ять комп'ютера після 700 натиснень на клавіші. Вірус також містить рядки тексту:

SyStEm is now halted.

### 4. Завантажувальні (бутові) віруси

Цей тип вірусів називають так через те, що вони впроваджуються в завантажувальний сектор диска (Boot-сектор) або в сектор, який вміщує системний завантажувач вінчестера (Master Boot Record).

Як і для файлових вірусів, виділимо групи бутових вірусів, а для кожного окремого вірусу – класифікаційний код, дескриптор і сигнатури.

#### Класифікаційний код завантажувального вірусу

Класифікаційний код бутового вірусу складається з префікса і кількісної характеристики.

***Префікс.*** Оскільки майже всі бутові віруси є резидентними, то використання символу R у префіксі їх класифікаційного коду недоцільне. Найважливішою властивістю бутових вірусів, які порівнюються за значенням з резидентністю файлових вірусів, є спроможність деяких бутових вірусів зберігатися в пам'яті після "теплого" перезавантаження шляхом натискування комбінації клавіш Ctrl-Alt-Del. Цю властивість позначають літерою W (service Warm reboot) в префіксі. Всі бутові віруси заражають дискети, але деякі з них заражають і вінчестер. Віруси, які інфікують тільки дискети (віруси “Brain”, “Den Zuk”), позначатимемо префіксом D.

При зараженні бут-сектора можливі два випадки: зараження бут-сектора – розділу С вінчестера (префікс B) і зараження MBR – виконуваної частини таблиці розділів (префікс M). Оскільки одним з найпоширеніших випадків розміщення хвоста бутового вірусу є його розміщення в псевдозбійних кластерах (що легко визначити, переглянувши їх вміст за допомогою Norton Utilites), то для таких вірусів у суфікс включають літеру

x, за якою стоїть кількість цих кластерів, наприклад, Bxl.

***Кількісна характеристика.*** Вибір кількісної характеристики для бутових вірусів має певну специфіку: якщо для файлових вірусів найхарактернішою ознакою зараження є збільшення довжини файла, то для бутових вірусів аналогічну роль відіграє зменшення розмірів оперативної пам'яті, яка доступна ОС.

Важливою вимогою до вибору властивостей вірусу, який використовується для класифікації, є можливість їх визначення на незараженій машині. Кількість блоків пам'яті, які використовуються бутовим вірусом, цьому критерію не відповідає, тому від цієї характеристики довелося відмовитися. Отже, використовують іншу характеристику бутового вірусу – вміст зараженого бут-сектора (вірніше, вміст перших його байтів). Разом з тим аналіз об'єму пам'яті, який повідомляє ОС, є дуже корисним для діагностики. При підозрюванні на зараження тим чи іншим вірусом можна виконати програму СНКТ38К, яка повідомляє значення об'єму пам'яті, а також дає ряд інших корисних повідомлень, включаючи об'єм пам'яті, зайнятий на диску збійними кластерами. Цю програму доцільно вставляти в код програми початкового завантаження.

За характеристику вибрано значення другого байта зараженого бут-сектора. Водночас зміст цього байта записується в 16-річній системі числення, що створює певну неузгодженість з характеристикою файлових вірусів, яка є десятковим числом. Тому у варіанті класифікаційного коду вірусу префікс і характеристика розділяються знаком "-" (мінус).

Слід підкреслити, що переглядати зміст бут-сектора можна лише тоді, коли попередньо завантажитись із захищеної від запису резервної дискети з операційною системою і необхідними антивірусними програмами, оскільки сама операція перегляду на зараженій машині може або перехоплюватися вірусом для підстановки "чистого" бут-сектора (так,наприклад, маскується вірус Dx3-E9.BRN – “Brain”), або, що ще гірше, бути тригером для яких-небудь несанкціонованих дій. Необхідно використовувати "холодне" (за допомогою клавіші RESET), а не "тепле" (за допомогою комбінації клавіш Ctrl-Alt-Del)

перезавантаження. Ця вимога базується на тому факті, що ряд бутових вірусів перехоплює переривання від клавіатури і при "теплому" перезавантаженні зберігає себе в пам'яті, навіть якщо перезавантаження здійснюється із захищеної системної дискети.

**Дескриптор завантажувального вірусу**

В головному дескрипторі відображені такі властивості:

A – деструктивні дії, які використовуються вірусом;

B – прояв вірусу;

L – довжина голови і хвоста вірусу в байтах, які розділені знаком "±";

M – маскування за наявності вірусу в пам'яті;

N – номер першого байта, що не збігається при порівнянні зараженого і нормального секторів початкового завантаження;

S – стратегія зараження (метод вибору "жертви", метод зберігання хвоста вірусу і оригінальної копії бут-сектора);

R (resident) – положення в оперативній пам'яті, реакція на "тепле" перезавантаження і розмір зайнятої пам'яті;

Z – побічні прояви дій вірусу.

**Сигнатура бутового вірусу**

Для бутових вірусів M-, I-, B-сигнатури використовуються аналогічно тому, як це було для файлових вірусів, а J-сигнатура – в дещо іншому вигляді. На відміну від J-сигнатури для файлових вірусів, в якій байти відповідають команді переходу і не враховуються, в J-сигнатурі для бутових вірусів вони враховуються. Це пов'язано з тим, що першою командою бут-сектора завжди є команда обходу таблиці параметрів диска, розмір якої, на відміну від розміру зараженого файла, не змінюється. Тому для бутових вірусів використовують переважно J-сигнатуру, яка складається з перших трьох байтів бут-сектора, і лише при необхідності доповнюється, починаючи з байта, на якому виконується команда переходу.

Для незараженого бут-сектора (наприклад, для MS-DOS версії 3.3) J-сигнатура дорівнює EB3490h (об'єктний код команди JMP, який служить для обходу таблиці параметрів). Цінність цієї еталонної J-сигнатури в тому, що вона порівняно легко запам'ятовується. Тому невідповідність перших трьох

байтів бут-сектора, що аналізується, вказаній еталонній J-сигнатурі свідчить про зараження бут-сектора.

**Принцип дії завантажувальних вірусів**

Завантажувальні віруси заражають завантажувальний сектор флопі-диска або boot-сектор вінчестера (MBR). Принцип дії завантажувальних вірусів оснований на алгоритмах запуску операційної системи при включенні або перезавантаженні комп'ютера – після необхідних тестів встановленого устаткування (пам'яті, дисків і т.д.) програма системного завантаження зчитує перший фізичний сектор завантажувального диску (A, C чи CD у залежності від параметрів, встановлених у BIOS Setup) і передає на нього керування.

У випадку дискети чи компакт-диску керування одержує boot-сектор, що аналізує таблицю параметрів диска (BPB - BIOS Parameter Block), вираховує адреси системних файлів операційної системи, зчитує їх у пам'ять і запускає на виконання. Системними файлами звичайно є MSDOS.SYS і IO.SYS, або IBMDOS.COM і IBMBIO.COM, або інші в залежності від установленої версії DOS, Windows чи інших ОС. Якщо ж на завантажувальному диску відсутні файли операційної системи, програма, розташована в boot-секторі диска, видає повідомлення про помилку і пропонує замінити завантажувальний диск.

У випадку вінчестера керування одержує програма, розташована в MBR вінчестера. Ця програма аналізує таблицю розбиття диска (Disk Partition Table), обчислює адресу активного boot-сектора (зазвичай цим сектором є boot-сектор диску C), завантажує його в пам'ять і передає на нього керування. Одержавши керування, активний boot-сектор вінчестера здійснює певні дії.

При зараженні дисків завантажувальні віруси "підставляють" свій код замість якої-небудь програми, що одержує керування при завантаженні системи. Принцип зараження, таким чином, однаковий: у всіх описаних вище способах вірус "змушує" систему під час перезапуску зчитувати її в пам'ять і віддати керування не оригінальному коду

завантажувальника, а коду вірусу.

Зараження дискет здійснюється єдиним відомим способом – вірус записує свій код замість оригінального коду boot-сектора дискети.

Вінчестер заражається трьома можливими способами – вірус записується або замість коду MBR, або замість коду boot-сектора завантажувального диска (звичайно диска C), або модифікує адреса активного boot-сектора в Disk Partition Table, розташованої в MBR вінчестера.

При інфікуванні диска вірус у більшості випадків переносить оригінальний boot-сектор (чи MBR) у який-небудь інший сектор диска (наприклад, у перший вільний). Якщо довжина вірусу більше довжини сектора, то в сектор, що заражається, поміщається перша частина вірусу, інші частини розміщаються в інших секторах (наприклад, у перших вільних).

**Розташування завантажувального вірусу**

Існує декілька варіантів розміщення на диску первинного завантажувального сектора і продовження вірусу в сектори вільних кластерів логічного диска, у не використовувані чи рідко використовувані системні сектори або у сектори, розташовані за межами диска.

Якщо продовження вірусу розміщається в секторах, що належать вільним кластерам диска (для пошуку цих секторів вірусу приходиться аналізувати таблицю розміщення файлів – FAT-таблицю), то, як правило, вірус позначає ці кластери як збійні (псевдозбійні кластери). Цей спосіб використовується вірусами "Brain", "Ping-Pong" і деякими іншими.

У вірусах сімейства "Stoned" задіяний інший метод. Ці віруси розміщують первинний завантажувальний сектор у не використовуваному чи рідко використовуваному секторі – в одному із секторів вінчестера (якщо такі є), розташованих між MBR і першим boot-сектором, а на дискеті такий сектор вибирається з останніх секторів кореневого каталогу.

Деякі віруси записують свій код в останні сектори вінчестера, оскільки ці сектори використовуються тільки тоді, коли вінчестер цілком заповнений інформацією (що є досить

рідким явищем, якщо врахувати розміри сучасних дисків). Однак такі віруси приводять до псування файлової системи OS/2, що у деяких випадках зберігає активний boot-сектор і системні дані саме в останніх секторах вінчестера.

Рідше використовується метод збереження продовження вірусу за межами диска. Досягається це двома способами:

- зменшення розмірів логічних дисків – вірус віднімає необхідні значення з відповідних полів BPB boot-сектора і Disk Partition Table вінчестера (якщо заражається вінчестер), зменшує в такий спосіб розмір логічного диску і записує свій код у "відрізані" від нього сектори;

- запис даних за межами фізичної розбивки диска. У випадку флопі-дисків вірусу для цього приходиться форматувати на диску додатковий трек (метод нестандартного форматування), наприклад, 80-й трек на дискеті. Існують віруси, що записують свій код за межами доступного простору вінчестера, якщо, зрозуміло, це допускається встановленим устаткуванням (вірус "Hare").

Звичайно, існують і інші методи розміщення вірусу на диску, наприклад, віруси сімейства "Azusa" містять у своєму тілі стандартний завантажник MBR і при зараженні записуються поверх оригінального MBR без його збереження.

При зараженні більшість вірусів копіює в код свого завантажника системну інформацію, що зберігається в первісному завантажнику (для MBR цією інформацією є Disk Partition Table, для Boot-сектора дискет – BIOS Parameter Block). В іншому випадку система виявиться нездатною завантажити себе, оскільки дискові адреси компонентів системи вираховуються на основі цієї інформації. Такі віруси досить легко видаляються переписуванням заново коду системного завантажника в boot-секторі і MBR - для цього необхідно завантажитися з незараженої системної дискети і використовувати команди SYS для знешкодження дискет і логічних дисків вінчестера чи FDISK/MBR для лікування зараженого MBR-сектора.

Однак деякі 100%-стелс віруси не зберігають цю

інформацію чи навіть, більш того, навмисно шифрують її. При звертанні системи або інших програм до заражених секторів вірус підставляє їхні незаражені оригінали, і завантаження системи відбувається без якихось збоїв, однак лікування MBR за допомогою FDISK/MBR у випадку такого вірусу приводить до втрати інформації про розбивку диска (Disk Partition Table). У цьому випадку диск може бути "оживлений" або переформатуванням із втратою всієї інформації, або відновленням Disk Partition Table "вручну", що вимагає значної кваліфікації.

Слід також зазначити той факт, що завантажувальні віруси дуже рідко "уживаются" разом на одному диску – часто вони використовують ті самі дискові сектори для розміщення свого коду (даних). У результаті код (дані) першого вірусу виявляється зіпсованим при зараженні другим вірусом, і система або зависає при завантаженні, або зациклюється, що також приводить до її зависання.

Користувачам сучасних ОС завантажувальні віруси також можуть спричинити неприємності. Незважаючи на те, що ці системи працюють з дисками напряму, минаючи виклики BIOS (що блокує вірус і унеможливлює подальше його поширення), код вірусу все-таки, хоч і дуже рідко, одержує керування при перезавантаженні системи. Тому вірус "March6", наприклад, може роками "жити" у MBR сервера і ніяк не впливати при цьому на його (сервера) роботу і продуктивність. Однак при випадковому перезавантаженні 6-го березня цей вірус цілком знищить усі дані на диску.

**Алгоритм роботи завантажувального вірусу**

Практично всі завантажувальні віруси резидентні.

Резидентні завантажувальні віруси впроваджуються в пам'ять комп'ютера при завантаженні з інфікованого диска. При цьому системний завантажник зчитує вміст першого сектора диска, з якого здійснюється завантаження, поміщає прочитану інформацію в пам'ять і передає на неї (тобто на програму-вірус) керування. Після цього починають виконуватися інструкції вірусу, що

- як правило, зменшує обсяг вільної пам'яті (слово за адресою 00400013), копіює у місце, що звільнилося, свій код і зчитує з диска своє продовження (якщо воно є). Надалі деякі віруси "чекають" завантаження ОС і відновлюють це слово в його первісне значення. У результаті вони виявляються розташованими не за межами ОС, а в окремих блоках пам'яті;

- перехоплює необхідні вектори переривань (зазвичай INT 13H), зчитує в пам'ять оригінальний boot-сектор і передає на нього керування.

Надалі завантажувальний вірус поводиться так само, як резидентний файловий: перехоплює звертання операційної системи до дисків і інфікує їх, у залежності від деяких умов робить деструктивні дії чи викликає звукові відеоефекти.

Існують нерезидентні завантажувальні віруси – при завантажені вони заражають MBR вінчестера і дискети, якщо ті присутні в дисководах. Потім такі віруси передають керування оригінальному завантажнику і на роботу комп'ютера більш не впливають.

**Приклади завантажувальних вірусів**

***Brain, сімейство***. Складається з двох практично співпадаючих нешкідливих вірусів "Brain-Ashar" і "Brain-Singapore". Вони заражають завантажувальні сектори дискет при зверненні до них (INT 13h, AH=02h). Продовження вірусу і первинний завантажувальний сектор розміщуються у вільних кластерах диска. При пошуку цих кластерів аналізують таблицю розміщення файлів (FAT). У FAT ці кластери позначаються як збійні ("псевдозбійні" кластери). У зараженого диска встановлюється нова мітка "(C) Brain". Віруси використовують "стелс"-механізм – при спробі проглянути завантажувальний сектор зараженого диска вони "підставляють" справжній сектор.

***CMOS.*** Небезпечний резидентний завантажувальний стелс-вірус. Псує CMOS. Копіює себе за адресою 9F800000, перехоплює INT 13h і записується в MBR вінчестера і boot-сектори дискет. Оригінальний MBR зберігає за адресою 0/0/2, оригінальний boot-сектор флопі-диска – в останньому секторі кореневого каталога.

***Pentagon.*** Небезпечний резидентний завантажувальний вірус. Частково зашифрований. Перехоплює INT 9, 13h і вражає boot-сектор флопі-дисків при зверненні до них. При зараженні диска оголошує в FAT збійні сектори і записує туди своє продовження і первинний boot-сектор (див. вірус "Brain"). Якщо при цьому дискета вже була уражена вірусом "Brain", то "Pentagon" лікує boot-сектор цього диска, змінює його мітку і потім заражає своєю копією. На диску, що заражається, створюється файл PENTAGON.TXT. Вірус "виживає" при теплому перезавантаженні. Містить тексти:

- (c) 1987 The Pentagon, Zorell Group
- first sector in segment
- Stoned

При завантаженні із зараженого флопі-диска з вірогідністю 1/8 на екрані з'являється повідомлення "Your PC is now Stoned!". Крім вказаної, містять рядок "LEGALISE MARIJUANA!". "Stoned.c" при зараженні MBR вінчестера знищує Disk Partition Table, після цього комп'ютер можна завантажити тільки з флопі-диска. "Stoned.d" 1 жовтня знищує інформацію на вінчестері.

***Hare.*** Дуже небезпечні резидентні файлово-завантажувальні стелс-поліморфік-вирусы. Записуються в кінець COM- і EXE-файлів, в MBR вінчестера і boot-сектора дискет. У файлах зашифровані тричі. Застосовують поліморфізм як у файлах, так і в заражених секторах.

При запуску зараженого файла вірус розшифровує себе, заражає MBR, трасує і перехоплює INT 21h і повертає управління програмі-носію. Під Win95 він перехоплює також INT 13h. Потім вірус записується у файли при їх запуску, закритті або при виході в DOS (AH=0,31h,4Ch). При відкритті заражених EXE-файлів лікує їх.

При завантаженні із зараженої дискети вірус записується в MBR і повертає управління первинному boot-коду, при цьому вірус не залишає в пам'яті своєї резидентної копії.

При зараженні MBR вірус трасує INT 13h або напряму працює з портами контролера, потім записує своє продовження (15 секторів) в трек, що знаходиться за межами оголошеного

розміру диска (LandZone?). Потім затирає Disk Partition Table (в результаті цього команда FDISK/MBR може привести до повної втрати даних на диску).

При завантаженні із зараженого MBR-сектора вірус відновлює Partition Table для того, щоб нормально завантажилася DOS (у цей момент стелс на рівні INT 13h ще не працює), потім зменшує розмір пам'яті (слово за адресою 00000413), копіює свій код в "відрізану" ділянку пам'яті, перехоплює INT 1Ch і передає управління первинному MBR-сектору. Перехопивши INT 1Ch, вірус чекає завантаження DOS, потім відновлює розмір системної пам'яті і перехоплює INT 13h, 21h, 28h. При першому виклику INT 28h він знову псує Disk Partition Table.

При викликах INT 13h вірус перехоплює звернення до флопі-дисків і заражає їх, для свого основного коду вірус форматує додатковий трек. При зверненнях до вже заражених дисків виконує стелс-програму.

## 5. Макро-віруси

Багато табличних і графічних редакторів, системи проектування, текстові процесори мають свої макро-мови для автоматизації виконання повторюваних дій. Ці макро-мови часто мають складну структуру і розвинений набір команд. Макро-віруси є програмами на макро-мовах, вбудованих у системи обробки даних. Для свого розмноження віруси цього класу використовують можливості макро-мов і при їх допомозі переносять себе з одного зараженого файла (документа, таблиці) в інші.

На кінець 1999 року відомо декілька систем, у яких виявлені макро-віруси. Це основні додатки Microsoft Office:

- редактор MS Word – мова WordBasic у MS Word 6/7 і VBA (Visual Basic for Applications), починаючи з MS Word 8;
- редактор таблиць MS Excel – мова VBA;
- редактор баз даних MS Access – мова VBA;
- редактор презентацій MS PowerPoint – мова VBA;
- менеджер проектів MS Project – мова VBA.

Піддався зараженню макро-вірусами також редактор AmiPro – спеціальна скрипт-мова.

Найбільше поширення одержали макро-віруси для Microsoft Office (Word, Excel і PowerPoint). Віруси в інших додатках MS Office досить рідкі, а для AmiPro відомий всего один макро-вирус. Можливе також існування макро-вірусів і для інших систем, що підтримують макро-мови достатньої потужності.

**Причини зараження макро-вірусами**

Для існування вірусів у конкретній системі (редакторі) необхідна наявність вбудованої в систему макро-мови з такими можливостями:

- програми на макро-мові прив'язані до документів (AmiPro) чи зберігаються в них (додатки MS Office);

- у макро-мові присутні команди копіювання макро-програм з одного файла в інший (AmiPro) або переміщати макро-програми у службові файли системи і файли, що редагуються (MS Office);

- є можливість одержання керування макро-програмою без втручання користувача (автоматичні чи стандартні макроси), тобто при роботі з файлом за певних умов (відкриття, закриття і т.д.) викликаються макро-програми (якщо такі є), що визначені спеціальним чином (AmiPro) чи мають стандартні імена (MS Office).

Дані можливості макро-мов призначені для автоматичної обробки даних у великих організаціях чи у глобальних мережах і дозволяють організувати так званий "автоматизований документообіг". З іншого боку, можливості макро-мов таких систем дозволяють вірусу переносити свій код в інші файли, і в такий спосіб заражати їх.

Більшість макро-вірусів можна вважати резидентними, оскільки вони присутні в області системних макросів протягом усього часу роботи редактора. Вони так само, як резидентні завантажувальні і файлові віруси, перехоплюють системні події і використовують їх для свого розмноження. До подібних подій відносяться різні системні виклики, що виникають при роботі з

документами Word і таблицями Excel (відкриття, закриття, створення, печатка і т.д.), виклик пункту меню, натискання на якунебудь клавішу чи досягнення певного моменту часу. Для перехоплення подій макро-віруси перевизначають один чи декілька системних макросів або функцій.

При зараженні деякі макро-віруси перевіряють наявність своєї копії в об'єкті, що заражається, і повторно себе не копіюють. Інші макро-вируси не роблять цього і переписують свій код при кожному зараженні. Якщо при цьому у файлі, що заражається, чи області системних макросів уже визначений макрос, ім'я якого збігається з макросом вірусу, то такий макрос виявляється знищеним.

**Загальні відомості про віруси в MS Office**

Фізичне розташування вірусу всередині файла залежить від його формату, що у випадку продуктів Microsoft надзвичайно складний - кожен файл-документ Word, Office чи таблиця Excel являють собою послідовність блоків даних (кожний з який також має свій формат), об'єднаних між собою за допомогою великої кількості службових даних. Цей формат називається OLE2 (Object Linking and Embedding). Структура файлів Office (OLE2) нагадує ускладнену файлову систему дисків DOS: "кореневий каталог" документа або таблиці вказує на основні підкаталоги різних блоків даних, кілька таблиць FAT містять інформацію про розташування блоків даних у документі і т.д.

Більш того, система Office Binder, що підтримує стандарти Word і Excel дозволяє створювати файли, що одночасно містять один чи декілька документів у форматі Word і одну чи декілька таблиць у форматі Excel. При цьому Word-віруси здатні вражати Word-документи, а Excel-віруси - Excel-таблиці, і все це можливо в межах одного дискового файла. Те ж справедливо і для Office.

Слід зазначити, що Word версій 6, 7 і вище дозволяє шифрувати присутні в документі макроси. Таким чином, деякі Word-віруси присутні в заражених документах у зашифрованому (Execute only) виді.

Більшість відомих вірусів для Word несумісні з

національними (у тому числі з російською) версіями Word, чи навпаки - розраховані тільки на локалізовані версії Word і не працюють під англійською версією. Однак вірус у документі все рівно залишається активним і може заражати інші комп'ютери з установленої на них відповідною версією Word.

Віруси для Word можуть заражати комп'ютери будь-якого класу, а не тільки IBM-PC. Зараження можливе у тому випадку, якщо на даному комп'ютері встановлений текстовий редактор, цілком сумісний з Microsoft Word версії 6 чи 7 (наприклад, MS Word for Macintosh). Те ж справедливо для Excel і Office.

Слід також зазначити, що складність форматів документів Word, таблиць Excel і особливо Office має таку особливість: у файлах-документах і таблицях присутні "зайві" блоки даних, тобто дані, що ніяк не пов'язані з текстом, що редагується, чи таблицями, або є випадковими копіями інших даних файла. Причиною виникнення таких блоків даних є кластерна організація даних у OLE2-документах і таблицях - навіть якщо введений всього один символ тексту, то під нього виділяється один чи навіть декілька кластерів даних. При збереженні документів і таблиць у кластерах, не заповнених "корисними" даними, залишається "сміття", що попадає у файл разом з іншими даними. Кількість "сміття" у файлах може бути зменшено скасуванням пункту настроювання Word/Excel "Allow Fast Save", однак це лише зменшує загальну кількість "сміття", але не забирає його цілком.

Наслідком цього є той факт, що при редагуванні документа його розмір змінюється незалежно від здійснених з ним дій – при додаванні нового тексту розмір файла може зменшитися, а при видаленні частини тексту – збільшитися. Те ж і з макро-вірусом при зараженні файла: його розмір може зменшитися, збільшитися чи залишитися незмінним.

Слід також зазначити той факт, що деякі версії OLE2.DLL містять невеликий недолік, у результаті якого при роботі з документами Word, Excel і особливо Office у блоки "сміття" можуть потрапити випадкові дані з диска, включаючи конфіденційні (вилучені файли, каталоги і т.д.). У ці блоки

можуть потрапити також команди вірусу. У результаті після лікування заражених документів активний код вірусу видаляється з файла, але в блоках "сміття" можуть залишитися частина його команд. Такі сліди присутності вірусу іноді видимі за допомогою текстових редакторів і навіть можуть викликати реакцію деяких антивірусних програм. Однак ці залишки вірусу зовсім нешкідливі,Word і Excel не звертають на них ніякої уваги.

**Принципи роботи Word/Excel/Office-вірусів**

При роботі з документом Word виконує різні дії: відкриває документ, зберігає, друкує, закриває і т.д. При цьому Word шукає і виконує відповідні вбудовані макроси – при збереженні файла по команді File/Save викликається макрос FileSave, при збереженні по команді File/SaveAs - FileSaveAs, при друці документів – FilePrint і т.д., якщо, звичайно, такі макроси визначені.

Існує також декілька "авто-макросів", що автоматично викликаються при різних умовах. Наприклад, при відкритті документа Word перевіряє його на наявність макросу AutoOpen. Якщо такий макрос присутній, то Word виконує його. При закритті документа Word виконує макрос AutoClose, при запуску Word викликається макрос AutoExec, при завершенні роботи – AutoExit, при створенні нового документа – AutoNew.

Схожі механізми (але з іншими іменами макросів і функцій) використовуються в Excel (Auto_Open, Auto_Close, Auto_Activate, Auto_Deactivate) і в Office (Document_Open, Document_Close, Document_New), у яких роль авто- і вбудованих макросів виконують авто- і вбудовані функції, що є присутніми у якому-небудь макросі чи макросах, причому в одному макросі можуть бути присутні декілька вбудованих функцій.

Автоматично (тобто без участі користувача) виконуються також макроси/функції, асоційовані з якою-небудь клавішею або моментом часу або датою, тобто Word/Excel викликають макрос/функцію при натисканні на яку-небудь конкретну клавішу (чи комбінацію клавіш) або при досягненні якого-небудь моменту часу. У Office97 можливості по перехопленню

подій дещо розширені, але принцип використовується той самий.

Макро-віруси, що вражають файли Word, Excel чи Office, як правило, користуються одним із трьох перерахованих вище прийомів – у вірусі або присутній авто-макрос (авто-функція), або перевизначений один зі стандартних системних макросів (асоційований якимось пунктом меню), або макрос вірусу викликається автоматично при натисканні на якусь клавішу чи комбінацію клавіш. Існують також напіввіруси, що не використовують цих прийомів і розмножуються, тільки коли користувач сам запускає їх.

Таким чином, якщо документ заражений, при відкритті документа Word викликає заражений автоматичний макрос AutoOpen (чи AutoClose при закритті документа) і, таким чином, запускає код вірусу, якщо це не заборонено системною змінною DisableAutoMacros. Якщо вірус містить макроси зі стандартними іменами, вони одержують керування під час виклику відповідного пункту меню (File/Open, File/Close, File/SaveAs). Якщо ж перевизначений який-небудь символ клавіатури, то вірус активізується тільки після натискання на відповідну клавішу.

Більшість макро-вірусів містять свої функції у вигляді стандартних макросів. Існують, однак, віруси, що використовують прийоми приховування свого коду і зберігають свій код у вигляді не-макросів. Відомо три подібних прийоми, усі вони використовують можливість макросів створювати, редагувати і виконувати інші макроси. Як правило, подібні віруси мають невеликий (іноді – поліморфний) макрос-завантажник вірусу, що викликає вбудований редактор макросів, створює новий макрос, заповнює його основним кодом вірусу, виконує і потім, як правило, знищує (щоб сховати сліди присутності вірусу). Основний код таких вірусів присутній або в самому макросі вірусу у вигляді текстових рядків (іноді – зашифрованих), або зберігається в області змінних документа чи в області Auto-text.

**Алгоритм роботи Word макро-вірусів**

Більшість відомих Word-вірусів під час запуску переносять свій код (макроси) в область глобальних макросів ("загальні" макроси), для цього вони використовують команди копіювання макросів MacroCopy, Organizer. Copy або за допомогою редактора макросів – вірус викликає його, створює новий макрос, вставляє в нього свій код і зберігає його в документі.

При виході з Word глобальні макроси (включаючи макроси вірусу) автоматично записуються в DOT-файл глобальних макросів (NORMAL. DOT). Таким чином, при наступному запуску редактора MS-Word вірус активізується в той момент, коли WinWord вантажить глобальні макроси, тобто відразу. Потім вірус перевизначає (чи вже містить у собі) один чи декілька стандартних макросів (наприклад, FileOpen, FileSave, FileSaveAs, FilePrint) і перехоплює в такий спосіб команди роботи з файлами. Під час виклику цих команд вірус заражає файл, до якого йде звертання. Цей вірус конвертує файл у формат Template (що унеможливлює подальші зміни формату файла, тобто конвертування в який-небудь не-Template формат) і записує у файл свої макроси, включаючи Auto-макрос.

Таким чином, якщо вірус перехоплює макрос FileSaveAs, то заражається кожен файл, що зберігається через перехоплений вірусом макрос. Якщо перехоплений макрос FileOpen, то вірус записується у файл при його зчитуванні з диска.

Другий спосіб впровадження вірусу в систему використовується значно рідше – він базується на так званих "Add-in" файлах, тобто файлах, що є службовими доповненнями до Word. У цьому випадку NORMAL.DOT не змінюється, а Word під час запуску завантажує макроси вірусу з файла (чи файлів), визначеного як "Add-in". Цей спосіб практично цілком повторює зараження глобальних макросів за тим виключенням, що макроси вірусу зберігаються не в NORMAL.DOT, а в якому-небудь іншому файлі.

Можливо також впровадження вірусу у файли, розташовані в каталозі STARTUP, – Word автоматично довантажує файли-темплейти з цього каталогу, але такі віруси поки що рідко

зустрічаються.

Розглянуті вище способи впровадження в систему являють собою деякий аналог резидентних вірусів. Аналогом нерезидентности є макро-віруси, що не переносять свій код в область системних макросів – для зараження інших файлів-документів вони або шукають їх за допомогою вбудованих у Word функцій роботи з файлами, або звертаються до списку останніх редагованих файлів (Recently used file list). Потім такі віруси відкривають документ, заражають його і закривають.

**Алгоритм роботи Excel макро-вірусів**

Методи розмноження Excel-вірусів в цілому аналогічні методам Word-вірусів. Розходження полягають у командах копіювання макросів (наприклад, Sheets.Copy) і у відсутності NORMAL.DOT – його функцію (у вірусному сенсі) виконують файли в STARTUP-каталозі Excel.

Слід зазначити, що існує два можливих варіанти розташування коду макро-вірусів у таблицях Excel. Переважна більшість таких вірусів записують свій код у форматі VBA, однак існують віруси, що зберігають свій код у старому форматі Excel версії 4.0. Такі віруси по своїй суті нічим не відрізняються від VBA-вірусів, за винятком відмінностей у форматі розташування кодів вірусу в таблицях Excel.

Незважаючи на те, що у нових версіях Excel (версія 5 і вище) використовуються досконаліші технології, можливість виконання макросів старих версій Excel була залишена для підтримки сумісності. З цієї причини всі макроси, написані у форматі Excel 4, цілком працездатні у всіх наступних версіях, незважаючи на те, що Microsoft не рекомендує використовувати їх і не включає необхідну документацію в комплект постачання Excel.

**Алгоритм роботи вірусів для Access**

Оскільки Access є частиною пакета Office, то віруси для Access являють собою такі ж самі макроси мовою Visual Basic, як і інші віруси, що заражають програми Office. Однак, у даному випадку замість авто-макросів у системі присутні автоматичні скрипти, що викликаються системою при різних подіях

(наприклад, Autoexec). Дані скрипти потім можуть викликати різні макро-програми.

Таким чином, при зараженні баз даних Access вірусу необхідно замінити який-небудь авто-скрипт і скопіювати в базу, що заражається, свої макроси.

Зараження скриптів без додаткових макросів не є можливим, оскільки мова скриптів досить примітивна і не містить необхідних для цього функцій. Слід зазначити, що в термінах Access скрипти називаються макросами (macro), а макроси – модулями (module), однак частіше використовується уніфікована термінологія - скрипти і макроси.

Лікування баз даних Access є більш складною задачею, ніж видалення інших макро-вірусів, оскільки у випадку Access необхідно знешкодити не тільки вірусні макроси, але й авто-скрипти. А оскільки значна частина роботи Access покладена саме на скрипти і макроси, то некоректне видалення чи деактивація якого-небудь елемента може привести до неможливості операцій з базою даних. Те саме справедливо і для вірусів – некоректне заміщення авто-скриптів може привести до втрати даних, що зберігаються в базі.

**Приклади макро-вірусів**

***Macro.Word.Box.*** Містить сім макросів AutoOpen, AutoClose, Box, Dead, FilePrint, FilePrintDefault, ToolsMacro. При викликах AutoOpen і AutoClose заражає глобальні макроси і документи. Макрос ToolsMacro використовується для заборони меню Tools/Macro. Решта макросів містить процедури зараження і різні ефекти. Виявляється декількома способами. При друці вставляє в документи рядки китайською мовою, виводить MessageBox, записує на диск і запускає вірус "OneHalf.3544", створює і програє WAV-файл (звуковий ефект). Викликає команди DOS:

- echo y|format c/u
- echo y|format c/u/vTwnos1

Містить рядки:

- Taiwan Super No.1 Macro Virus
- Twno1-S

- Today Is My Birthday
- Macro.Word.Catch

Містить шість макросів AutoOpen, encrypt1, FileSave, AutoClose, infectdoc, infectnorm. Зараження системної області макросів і документів відбувається при відкритті файлів. Вірус замінює в документах букви "і" на "о", "о" на "і", "а" на "е", "е" на "а". Причому підміна символів непомітна. При відкритті заражених файлів вірус відновлює текст документа в первинному вигляді, а при збереженні документів на диск – знову замінює. В результаті після лікування вірусу текст документів може виявитися зіпсованим. Після кожної заміни вірус видає в StatusBat крапку. При закритті документів залежно від лічильника випадкових чисел (з вірогідністю 1%) вірус видає повідомлення:

Its a Catch 22 Situation!

***Macro.Excel.Soldier.*** Поліморфний макро-вірус, заражає електронні таблиці Excel. Містить чотири функції з постійними іменами Auto_Open, Auto_Close, Delay, Poly; і декілька функцій з випадковими іменами. При відкритті зараженої таблиці видаляє рядки меню Format/Sheet/Hide і Format/Sheet/Unhide (стелс). При закритті заражає файли, що знаходяться у поточному каталозі. При зараженні залежно від системного датчика випадкових чисел вставляє в початок тексту функції з випадковими іменами і випадковими рядками. Також залежно від випадкового числа виводить в заголовок Excel рядок, що біжить: Microsoft Excel .

## 6. Мережеві віруси

Мережеві віруси поширюються по комп'ютерних мережах. Існують комбінації, наприклад, файлово-бутові віруси, які заражають і файли, і бут-сектори дисків. Крім того, у ком-п'ютерних мережах можуть розповсюджуватись віруси будь-яких типів. Віруси розповсюджуються від одного користувача до іншого внаслідок обміну програмними продуктами. Локальні мережі, як відомо, призначені для сумісного використання програмних пакетів кількома користувачами. Мережі

дозволяють широко обмінюватися програмами і даними. Очевидно, що при цьому створюється зручне середовище для розповсюдження вірусу. Так, якщо вірус заразив програму login.exe, яка знаходиться на сервері і яку запускає кожен користувач при вході в мережу, то дуже швидко вірус з'явиться на всіх робочих станціях. Але на практиці ситуація виглядає не дуже драматично, бо мережні операційні системи мають механізми захисту і розподілу користувачів. При грамотному використанні цих можливостей можна обмежити область, в якій може розповсюдитись вірус, внесений з робочої станції, тільки робочою областю того користувача, який вніс його в систему.

Існують мережні віруси, які розраховані на спеціальні мережні диски, наприклад, на систему Netware – вони несанкціоновано входять у систему і, використовуючи максимальні повноваження супервізора, пошкоджують програми, які знаходяться на мережних дисках.

Все це свідчить про те, що рівень захищеності сучасних операційних систем поки що бажає бути кращим. Доки залишається принципова можливість такого втручання, доти існує небезпека появи таких вірусів.

***IRC-хробаки***. **IRC (Internet Relay Chat)** – це спеціальний протокол, розроблений для комунікації користувачів Інтернет у реальному часі. Цей протокол надає можливість так званої Інтернет-розмови за допомогою спеціально розробленого програмного забезпечення. IRC чимось схожий на телефонну розмову, за винятком того, що в розмові можуть брати участь більш двох співрозмовників, що поєднуються по інтересах у різні групи IRC-конференцій. Для підтримки IRC-конференцій створені різні IRC-сервера, до яких підключаються учасники IRC-розмов. В усьому світі нараховується величезна кількість IRC-серверів, об'єднаних у так звані мережі. Найбільшою є мережа EFnet, сервер якої щодня одночасно відвідують кілька десятків тисяч користувачів.

Для підключення до IRC-сервера і ведення IRC-розмов розроблені спеціальні програми – IRC-клієнти. Підключившись до IRC-сервера за допомогою програми-клієнта, користувач

зазвичай вибирає тему IRC-конференції, командою join входить в одну або декілька конференцій ("канали" у термінах IRC) і починає спілкування з іншими "мешканцями" цих каналів.

Крім відвідування загальних (public) конференцій користувачі IRC мають можливість спілкуватися один-на-один з будь-яким іншим користувачем (private), при цьому вони навіть не обов'язково повинні бути на одному каналі. Крім цього існує досить велика кількість IRC-команд, за допомогою яких користувач може одержати інформацію про інших користувачів і канали, змінювати деякі установки IRC-клієнта та інше. Існує також можливість передавати і приймати файли – саме на цій можливості і базуються IRC-хробаки.

***IRC-клієнти***. На комп'ютерах з MS Windows найпоширенішими клієнтами є mIRC і PIRCH. Це не дуже об'ємні, але досить складні програмні продукти, що крім надання основних послуг IRC (підключення до серверів і каналів) мають ще і масу додаткових можливостей.

До таких можливостей відносяться, наприклад, сценарії роботи (скрипти) і завдання автоматичної реакції на різні події. Наприклад, з появою під час розмови визначеного слова IRC-клієнт передає повідомлення користувачу, що послав його. Можливе відключення користувача від каналу; посилка персональних повідомлень новим користувачам, що підключаються до каналу; і багато чого іншого. У PIRCH-клієнті, наприклад, подій, на які передбачена реакція, більше 50.

***Скрипт-хробаки***. Як виявилося, могутня і розгалужена система команд IRC-клієнтів дозволяє на основі їх скриптів створювати комп'ютерні віруси, що передають свій код на комп'ютери користувачів мереж IRC, так називані IRC-хробаки.

Перший інцидент із IRC-хробаком зафіксований наприкінці 1997 року: користувачами mIRC-клієнта був виявлений скрипт (файл SCRIPT.INI), що переносив свій код через канали IRC і заражав mIRC-клієнтів на комп'ютерах користувачів, що підключалися до заражених каналів. Як виявилося, скрипт-хробаки є досить простими програмами, і через досить короткий час на основі першого mIRC-хробака

були створені і "випущені" у мережі декілька десятків різних скрипт-хробаків.

Принцип дії таких IRC-хробаків приблизно однаковий. За допомогою IRC-команд файл сценарію роботи (скрипт) чи реакції на IRC-події автоматично посилається з зараженого комп'ютера кожному користувачу, що під'єднується до каналу. Надісланий файл-сценарій заміщає стандартний і при наступному сеансі роботи вже знову заражений клієнт буде розсилати хробака. Хробаки при цьому використовують особливості конфігурації клієнта, завдяки якій прийняті файли всіх типів розміщуються в кореневий каталог клієнта. Цей каталог також містить і основні скрипти клієнта, включаючи завантажувальні mIRC-скрипти SCRIPT.INI, MIRC.INI і PIRCH-скрипт EVENTS.INI. Ці скрипти автоматично виконуються клієнтом при старті і далі використовуються як основний сценарій його роботи.

Деякі IRC-хробаки також містять троянський компонент – по заданих ключових словах здійснюють руйнівні дії на уражених комп'ютерах. Наприклад, хробак "pIRCH.Events" по визначеній команді стирає усі файли на диску користувача.

У скрипт-мовах клієнтів mIRC і PIRCH також існують оператори для запуску звичайних команд ОС і модулів, що виконуються, програм DOS і Windows. Ця можливість IRC-скриптів послужила основою для появи скрипт-хробакв нового покоління, що крім скриптів заражали комп'ютери користувачів EXE-вірусами, установлювали "троянських коней", і т.п.

Скрипт-хробаки працездатні тільки в тому випадку, якщо користувач дозволяє копіювання файлів з мережі на свій комп'ютер. Дана опція IRC-клієнтів називається "DCC autoget" – одержання файлів по протоколу DCC автоматично і без попереджуючого повідомлення. При відключеній опції заражений файл приймається, розміщується в каталозі клієнта і в наступному сеансі роботи продовжує своє поширення. При цьому користувач не одержує ніяких попереджуючих повідомлень.

Слід зазначити, що фірма-виготовлювач клієнта mIRC

відреагувала досить оперативно і буквально через кілька днів після появи першого хробака випустила нову версію свого клієнта, у якій були закриті пробіли в захисті.

**Приклади мережевих вірусів**

***mIRC.Acoragil і mIRC.Simpsalapim***. Перші відомі mIRC-хробаки виявлені наприкінці 1997 року. Назви одержали по кодових словах, що використовуються хробаками: якщо в тексті, переданому в канал будь-яким користувачем, присутній рядок "Acoragil", то всі користувачі, заражені хробаком mIRC.Acoragil автоматично відключаються від каналу. Те саме відбувається з хробаком mIRC. Simpsalapim – він аналогічно реагує на рядок "Simpsalapim". При розмноженні хробаки командами mIRC пересилають свій код у файлі SCRIPT. INI кожному новому користувачу, що підключається до каналу. Містять троянську частину. Хробак mIRC.Simpsalapim містить код захоплення каналу IRC – якщо mIRC власника каналу заражений, то по введенню кодового слова "ananas", зловмисник перехоплює керування каналом. Хробак mIRC. Acoragil по кодових словах пересилає системні файли ОС. Деякі кодові слова обрані так чином, що не привертають уваги жертви – hi, суа чи the. Одна з модифікацій цього хробака пересилає зловмиснику файл паролів UNIX.

***Win95.Fono.*** Небезпечний резидентный файлово-завантажувальний поліморфік-вірус. Використовує mIRC як один із способів свого поширення: перехоплює системні події Windows і при запуску файла MIRC32.EXE активізує свою mIRC-процедуру. При цьому відкриває файл MIRC.INI і записує в його кінець команду, що знімає захист:

[fileserver]

Warning=Off .

Потім створює файли SCRIPT.INI і INCA.EXE. Файл INCA.EXE містить дроппер вірусу, скрипт файла SCRIPT.INI пересилає себе і цей дроппер у канал IRC кожному, хто приєднується до каналу або виходить з нього.

***pIRCH.Events***. Перший відомий PIRCH-хробак. Розсилає себе кожному користувачу, що приєднався. По ключових словах

виконує різні дії, наприклад:

- по команді ".query" відбувається свого роду переклик, по якому заражені системи відповідають <Так, я вже заражена>;
- по команді ".exit" завершує роботу клієнта;

По інших командах хробак видаляє всі файли з диска С, надає доступ до файлів на зараженому комп'ютері і т.д.

***IRC-Worm.Pron.*** Мережевий вірус-черв'як, зашифрований. Розмножується в IRC-каналах і використовує для свого розмноження mIRC-клієнта. Має дуже невелику довжину – всього 582 байти. Передається з мережі на комп'ютер у вигляді файла PR0N.BAT. При його запуску вірус копіює себе у файл PR0N.COM і запускає його на виконання. Заголовок вірусу влаштований таким чином, що він може виконуватися як BAT-, так і COM-програма, і в результаті управління передається на основну процедуру зараження системи. При зараженні системи вірус використовує дуже простий прийом: він копіює свій BAT-файл в поточний каталог і в каталог C\WINDOWS\SYSTEM (якщо такий відсутній, то вірус не заражає систему). Потім вірус записує свій код у файл WINSTART.BAT. Для розповсюдження свого коду через mIRC вірус створює новий файл SCRIPT.INI в каталозі mIRC-клієнта. Цей каталог вірус шукає за чотирма варіантами:

C\MIRC; C\MIRC32; C\PROGRA~1\MIRC; C\PROGRA~1\MIRC32 .

Скрипт вірусу містить всього одну команду – кожному користувачу, що підключається до зараженого каналу, передається вірусний файл PR0N.BAT. Вірус містить рядок-"копірайт".

## 7. Стелс-віруси

Стелс-віруси тими чи іншими способами приховують факт своєї присутності в системі, підставляючи замість себе незаражені ділянки інформації. Крім того, такі віруси при зверненні до файлів використовують досить оригінальні алгоритми, що “обманюють” резидентні антивірусні програми.

Відомі стелс-віруси всіх типів – завантажувальні віруси,

файлові DOS-віруси і навіть макро-віруси.

***Завантажувальні стелс-віруси.*** Завантажувальні стелс-віруси для приховання свого коду використовують два основних способи.

Перший спосіб полягає в тому, що вірус перехоплює команди читання зараженого сектора (INT 13h) і підставляє замість нього незаражений оригінал. Цей спосіб робить вірус невидимим для будь-якої DOS-програми, включаючи антивіруси, нездатні "лікувати" оперативну пам'ять комп'ютера. Можливе перехоплення команд читання секторів на рівні більш низькому, чим INT 13h.

Другий спосіб спрямований проти антивірусів, що підтримують команди прямого читання секторів через порти контролера диска. Такі віруси при запуску будь-якої програми (включаючи антивірус) відновлюють заражені сектори, а після закінчення її роботи знову заражають диск. Оскільки для цього вірусу приходиться перехоплювати запуск і закінчення роботи програм, то він повинен перехоплювати також DOS-переривання INT 21h.

З деяким застереженням стелс-вірусами можна назвати віруси, що вносять мінімальні зміни в сектор, що заражається (наприклад, при зараженні MBR змінюють тільки активну адресу завантажувального сектора – зміні підлягають тільки 3 байти), або маскуються під код стандартного завантажника.

***Файлові стелс-віруси***. Більшість файлових стелс-вірусів використовує ті самі прийоми, що приведено вище: вони або перехоплюють DOS-виклики звертання до файлів (INT 21h), або тимчасово лікують файл при його відкритті і заражають при закритті. Так само, як і для boot-вірусів, існують файлові віруси, що використовують для своїх стелс-функцій перехоплення переривань більш низького рівня – виклики драйверів DOS, INT 25h і навіть INT 13h.

Повноцінні файлові стелс-віруси, що використовують перший спосіб приховання свого коду, здебільшого досить громіздкі, оскільки їм потрібно перехоплювати велику кількість DOS-функцій роботи з файлами: відкриття/закриття,

читання/записування, пошук, запуск, перейменування і т.д., причому необхідно підтримувати обидва варіанти деяких викликів (FCB/ASCII), а після появи Windows 95/NT їм стало необхідно також обробляти третій варіант – функції роботи з довгими іменами файлів.

Деякі віруси використовують частину функцій повноцінного стелс-віруса. Найчастіше вони перехоплюють функції DOS FindFirst і FindNext (INT 21h, AH=11h, 12h, 4Eh, 4Fh) і зменшують розмір заражених файлів. Такий вірус неможливо визначити по зміні розмірів файлів, якщо, звичайно, він резидентно знаходиться в пам'яті.

Програми, що не використовують зазначені функції DOS (наприклад, "Нортоновські утиліти"), а прямо використовують вміст секторів, що зберігають каталог, показують правильну довжину заражених файлів.

***Макро-стелс-віруси***. Реалізація стелс-алгоритмів у макро-вірусах є, напевно, найбільш простою задачею – досить усього лише заборонити виклик меню File/ Templates або Tools/Macro. Досягається це або видаленням цих пунктів меню зі списку, або їхньою підміною на макроси FileTemplates і ToolsMacro.

Частково стелс-вірусами можна назвати невелику групу макро-вірусів, що зберігають свій основний код не в самому макросі, а в інших областях документа – у його змінних чи в Auto-text.

**Приклади стелс-вірусів**

***Crusher.*** Безпечний резидентний MBR-EXE-стелс-вірус. При запуску зараженого файла він записується в MBR вінчестера, потім перехоплює INT 21h і записується в початок EXE-файлів при їх копіюванні. При завантаженні з ураженого диска перехоплює INT 1Ch, чекає завантаження DOS, потім відновлює INT 1Ch, перехоплює INT 21h і приступає до зараження файлів. Якщо при роботі вірусу йому не вистачає пам'яті, він повідомляє "Insufficient memory" і повертається в DOS. При запуску CHKDSK вірус виводить текст:

Crusher… You are damned. Bit Addict / Trident.

***Ekoterror.*** Резидентний небезпечний стелс-вірус, при

запуску зараженого файла записується в MBR вінчестера і передає управління програмі-носію, при завантаженні з ураженого MBR перехоплює INT 8, 13h, а потім, використовуючи INT 8, перехоплює INT 21h і записується в початок .COM-файлів при їх створенні. Періодично розшифровує і виводить текст:

EkoTerror (C) 1991 ATK-toimisto P.Linkola Oy

Kovalevysi on poistettu kДутФstД luonnonsuojelun nimessД.

VihreДssД yhteiskunnassa ei saa olla ydinsДhkФllД toimivia kovalevyjД.

а потім завішує комп'ютер. В деяких випадках некоректно уражає MBR, в результаті DOS гине при завантаженні.

***Rasek, сімейство***. Дуже небезпечні файлово-завантажувальні віруси, що самошифруються. При запуску зараженого файла записують себе в MBR вінчестера, потім перехоплюють INT 13h, 12h. Переривання INT 13h використовується для реалізації стелс-механізму при читанні ураженої MBR. Віруси також записують в Boot-сектори флопі-дисків програму, яка при завантаженні з такого флопі стирає FAT вінчестера. Переривання INT 21h використовується вірусом для зараження COM- і EXE-файлів при їх запуску, вірус записується в кінець файлів. У тілі вірусу міститься рядок "AND.COM", вірус шукає цей рядок в імені файла і не вражає такі файли (COMMAND.COM). У тілі вірусів також міститься і інші рядки:

"Rasek.1310" ASEK v1.1,from LA CORUeA(SPAIN) .Jan93

"Rasek.1489" RaseK v2.1,from LA CORUeA(SPAIN) .Mar93

***Vecna.*** Дуже небезпечний резидентний файлово-завантажувальний стелс-вірус. Заражає boot-сектори дискет, MBR вінчестера і записується поверх EXE-файлів (псує їх). При запуску зараженого EXE-файла записується в MBR вінчестера, розшифровує і виводить текст:

Out of memory.

Потім повертає управління DOS. При завантаженні з диска перехоплює INT 13h, залишається резидентним в пам'яті і заражає дискети і EXE-файли на дискетах. Під налагоджувачем і

на Pentium-комп'ютерах виводить текст:

Vecna Live ...

Має досить серйозну помилку – може повернути управління оригінальному обробнику INT 13h із зіпсованим вмістом регістра AX, що може привести до втрати даних на диску і навіть до його форматування.

***Kyokushinkai.*** Дуже небезпечний резидентний файлово-завантажувальний вірус. При запуску зараженого записується в MBR вінчестера, перехоплює INT 12h, 13h, 1Ch, 21h і при запуску програм шукає EXE-файли і записується в їх кінець. Заражений MBR-сектор не видно при активному в пам'яті вірусі (стелс). Залежно від поточного часу стирає системні сектори рядком:

+++++++ КШФkБshЛдkДЛ +++++++.- 39-mynrazCmeroizв.....

## 8. Поліморфік-віруси

Не так давно виявлення вірусів було простою справою: кожен вірус створював точну копію самого себе при тиражуванні і інфікуванні нових файлів і завантажувальних секторів, тому антивірусним програмам необхідно було тільки знати послідовність байтів, що становлять вірус. Для кожного вірусу фахівці виявляли унікальну послідовність байтів – його сигнатуру. Наявність такої сигнатури служила високонадійним індикатором присутності небажаного коду, що і примусило авторів вірусів спробувати приховувати будь-яку послідовність байтів, здатну видати присутність їх творінь. Вони стали робити це шляхом шифрування вірусів.

Віруси, що шифрують свій код, відомі досить давно. Проте самі процедури розшифрування досить легко виявити, зокрема, тому, що далеко не всі автори вірусів мають досить знань для написання власних процедур шифрування і розшифрування, тому багато вірусів використовують для цього один і той самий код. Тепер сканери вірусів шукають певні процедури розшифрування. Хоча виявлення такої процедури ще нічого не говорить про те, який саме вірус присутній у зашифрованому

вигляді, але це вже сигнал про наявність вірусу. Тому останнім прийомом зловмисників стає поліморфізм.

Перші поліморфні віруси Tequila і Maltese Amoeba з'явилися в 1991 році. Все б нічого, але в 1992 році автор, відомий під псевдонімом Dark Avenger, написав свого роду комплект «Зроби сам» для мутаційного механізму, який він зробив частиною вірусу Maltese Amoeba. До 1992 року розробники вірусів працювали насправді дарма. Абсолютно ясно, що кваліфікація професіоналів у сфері антивірусної безпеки ніяк не нижча, і тому багатомісячні зусилля "вірусописьменників" коштували в крайньому випадку зайвих годин роботи для фахівців. Адже всі зашифровані віруси обов'язково містили якийсь незашифрований фрагмент: сам розшифровувач або деяку його частину, по яких можна було б побудувати сигнатуру даного вірусу і потім вже боротися з ним звичними способами.

Ситуація змінилася, коли були придумані алгоритми, що дозволяють не тільки шифрувати код вірусу, але і міняти розшифровувач. Сама постановка такої задачі питань не викликає: ясно, що можна побудувати різні розшифровувачі. Суть у тому, що цей процес автоматизований, і кожна нова копія вірусу містить новий розшифровувач, кожен біт якого може відрізнятися від бітів розшифровувача копії, що породила її.

Отже, до поліморфік-вірусів відносяться ті з них, які неможливо (чи вкрай важко) знайти за допомогою так званих вірусних масок – ділянок постійного коду, специфічних для конкретного вірусу. Досягається це двома основними способами:

- шифруванням основного коду вірусу з непостійним ключем і випадковим набором команд розшифровувача;

- зміною самого виконуваного коду вірусу.

Існують також інші, досить екзотичні приклади поліморфізму: DOS-вірус "Bomber", наприклад, не зашифрований, однак послідовність команд, що передає керування коду вірусу, є цілком поліморфною. Поліморфізм різного ступеня складності зустрічається у вірусах усіх типів –

завантажувальних, файлових і навіть у макро-вірусах.

**Поліморфні розшифровувачі**

Найпростішим прикладом частково поліморфного розшифровувача є наступний набір команд, в результаті застосування якого жоден байт коду самого вірусу і його розшифровувача не є постійним при зараженні різних файлів:

```
MOV  reg_1, count; reg_1, reg_2, reg_3 вибираються з
MOV  reg_2, key; AX,BX,CX,DX,SI,DI,BP
MOV  reg_3, _offset; count, key, _offset також можуть
```

мінятися

```
_LOOP:
xxx    byte ptr [reg_3], reg_2  ; xor, add чи sub
DEC   reg_1
Jxx   _loop; ja чи jnc; далі йдуть зашифровані код і дані
```

вірусу.

Більш складні поліморфік-віруси використовують значно більш складні алгоритми для генерації коду своїх розшифровувачів. Приведені вище інструкції переставляються місцями від зараження до зараження, розбавляються нічого не змінюючими командами типу

NOP, STI, CLI, STC, CLC, DEC <невикористовуваний регістр>

або

XCHG <невикористовувані регістри>.

Повноцінні ж поліморфік-віруси використовують ще більш складні алгоритми, у результаті роботи яких у розшифровувачі вірусу можуть зустрітися операції SUB, ADD, XOR, ROR, ROL і інші в довільній кількості і порядку. Завантаження і зміна ключів і інших параметрів шифровки виробляється також довільним набором операцій, у якому можуть зустрітися практично всі інструкції процесора Intel (ADD, SUB, TEST, XOR, OR, SHR, SHL, ROR, MOV, XCHG, JNZ, PUSH, POP ...) із усіма можливими режимами адресації. З'являються також поліморфік-віруси, розшифровувач яких використовує інструкції аж до Intel386. В результаті на початку файла, зараженого подібним вірусом, йде набір безглуздих на перший

погляд інструкцій, причому деякі комбінації, що цілком працездатні, не аналізуються фірмовими дизасемблерами (наприклад, сполучення CSCS чи CSNOP). І серед цієї "каші" з команд і даних зрідка прослизають MOV, XOR, LOOP, JMP – інструкції, що дійсно є "робітниками".

**Рівні поліморфізму**

Існує розподіл поліморфік-вірусів на рівні в залежності від складності коду, що зустрічається в розшифровувачах цих вірусів. Такий розподіл уперше запропонував доктор Алан Соломон, а згодом Весселин Бончев розширив його.

Рівень 1: віруси, що мають деякий набір розшифровувачів з постійним кодом і при зараженні вибирають один з них. Такі віруси є "напів-поліморфіками" і носять також назву "олігоморфік" (oligomorphic). Приклади: "Cheeba", "Slovakia", "Whale".

Рівень 2: розшифровувач вірусу містить одну чи кілька постійних інструкцій, основна ж його частина непостійна.

Рівень 3: розшифровувач містить невикористовувані інструкції – "сміття" типу NOP, CLI, STI і т.д.

Рівень 4: у розшифровувачі використовуються взаємозамінні інструкції і зміна порядку проходження (перемішування) інструкцій. Алгоритм розшифрування при цьому не змінюється.

Рівень 5: використовуються всі перераховані вище прийоми, алгоритм розшифрування не постійний, можливе повторне зашифрування коду вірусу і навіть часткове зашифрування самого коду розшифровувача.

Рівень 6: permutating-віруси. Зміні підлягає основний код вірусу – він поділяється на блоки, що при зараженні переставляються в довільному порядку. Вірус при цьому залишається працездатним. Подібні віруси можуть бути незашифрованими.

Наведений вище розподіл не вільний від недоліків, оскільки створений за єдиним критерієм – можливість детектувати вірус по коду розшифровувача за допомогою стандартного прийому вірусних масок: рівень 1 – для

детектування вірусу досить мати кілька масок; рівень 2 – детектування по масці з використанням "wildcards"; рівень 3 – детектування по масці після видалення інструкцій-"сміття"; рівень 4 – маска містить кілька варіантів можливого коду, тобто стає алгоритмічною; рівень 5 – неможливість детектування вірусу по масці.

Недостатність такого розподілу продемонстрована у вірусі 3-го рівня поліморфічності, що так і називається – "Level3". Цей вірус, будучи одним з найбільш складних поліморфік-вірусів, за приведеним вище розподілом попадає в Рівень 3, оскільки має постійний алгоритм розшифровки, перед яким стоїть велика кількість команд-"сміття". Однак у цьому вірусі алгоритм генерування "сміття" доведений до досконалості в коді розшифровувача можуть зустрітися практично всі інструкції процесора i8086.

Якщо зробити розподіл на рівні з погляду антивірусів, що використовують системи автоматичної розшифровки коду вірусу (емулятори), то розподіл на рівні буде залежати від складності емуляції коду вірусу. Можливо детектування вірусу й інших прийомів, наприклад, розшифровка за допомогою елементарних математичних законів і т.д.

**Зміна виконуваного коду**

Найбільш часто подібний спосіб поліморфізму використовується макро-вірусами, що при створенні своїх нових копій випадковим чином змінюють імена своїх змінних, вставляють порожні рядки чи змінюють свій код яким-небудь іншим способом. У такий спосіб алгоритм роботи вірусу залишається без змін, але код вірусу практично цілком міняється від зараження до зараження.

Рідше цей спосіб застосовується складними завантажувальними вірусами. Такі віруси впроваджують у завантажувальні сектори лише досить коротку процедуру, що зчитує з диска основний код вірусу і передає на нього керування. Код цієї процедури вибирається з декількох різних варіантів (які також можуть бути розведені "порожніми" командами), команди переставляються між собою і т.д.

Ще рідше цей прийом зустрічається у файлових вірусів – адже їм приходиться цілком змінювати свій код, а для цього вимагаються досить складні алгоритми.

На сьогоднішній день відомі всего два таких віруси, один із яких ("Ply") випадковим образом переміщає свої команди по своєму тілу і заміняє їх на команди JMP чи CALL. Інший вірус ("TMC") використовує більш складний спосіб – щоразу при зараженні вірус змінює місцями блоки свого коду і даних, вставляє "сміття", у своїх асемблерних інструкціях встановлює нові значення оффсетів на дані, змінює константи і т.д. В результаті, хоча вірус і не шифрує свій код, він є поліморфік-вірусом – у коді не присутній постійний набір команд. Більш того, при створенні своїх нових копій вірус змінює свою довжину.

Деякі віруси (наприклад, віруси сімейства Eddie, Murphy) використовують частину функцій повноцінного вірусу-невидимки. Зазвичай вони перехоплюють функції DOS FindFirst і FindNext і «зменшують» розмір заражених файлів. Такий вірус неможливо визначити за зміною розмірів файлів, якщо, звичайно, він резидентно знаходиться в пам'яті. Програми, що не використовують вказані функції DOS (наприклад, Norton Commander), а напряму звертаються до вмісту секторів, які зберігають каталог, показують правильну довжину заражених файлів.

При інфікуванні файла вірус може здійснювати дії, що маскують і прискорюють його розповсюдження. До подібних дій можна віднести обробку атрибуту Read-only, зняття його перед зараженням і подальше відновлення цього атрибуту. Багато файлових вірусів прочитують дату останньої модифікації файла і відновлюють її після зараження. Для маскування свого розповсюдження деякі віруси перехоплюють переривання DOS, що виникає при зверненні до диска, захищеного від запису, і самостійно обробляють його. Тому серед особливостей алгоритму файлового вірусу можна назвати наявність або відсутність обробки і швидкість його розповсюдження. Швидкість розповсюдження файлових вірусів, що заражають

файли тільки під час їх запуску на виконання, буде нижчою, ніж у вірусів, що заражають файли при їх відкритті, перейменуванні, зміні їх атрибутів і т.д. Деякі віруси при створенні своєї копії в оперативній пам'яті намагаються зайняти область пам'яті з найстаршими адресами, руйнуючи тимчасову частину командного інтерпретатора COMMAND.COM. Після закінчення роботи зараженої програми тимчасова частина інтерпретатора відновлюється, при цьому відбувається відкриття файла COMMAND.COM і його зараження, якщо вірус вражає файли при їх відкритті.

Приклади поліморфік-вірусів

***Amoeba.2367***. Дуже небезпечний резидентний поліморфик-вірус. Перехоплює INT 21h і записується в кінець COM- і EXE-файлів при їх запуску або відкритті. 21 березня і 1 листопада знищує інформацію на вінчестері. Містить тексти:

Tosee aworld in a grain of sand,
And a heaven in a wildflower
Hold Infinity in the palm of your hand
And Eternity in an hour.

"THE VIRUS 16/3/91 AMOEBA virus by the Hacker Twins (C) 1991 This is nothing, wait for the release of AMOEBA II-The universal infector, hidden to any eye but ours! Dedicated to the University of Malta- the worst educational system in the universe,and the destroyer of 5X2 years of human life.

***Simulation.*** Безпечний нерезидентний поліморфік-вірус. Шукає .COM-файли і записується в їх кінець. Періодично виводить одне з повідомлень, після чого завішує комп'ютер:

HA HA HA YOU HAVE A VIRUS ! FRODO LIVES!
Have you ever danced with the Devil in the pale moonlight?
DATACRIME VIRUS RELEASED 1 MARCH 1989 ALIVE...
Your system is infected by the SIMULATION virus.
Have a nice day!
Predator (файлово-завантажувальні)

Нешкідливі резидентні COM-EXE-MBR-Boot-поліморфік-віруси. Під час запуску зараженого файла трасують і

перехоплюють INT 13h, 21h і записуються в MBR вінчестера. Потім записуються в кінець COM- і EXE-файлів при зверненнях до них. Вражають Boot-сектори дискет. При завантаженні з ураженого флопі-диска перехоплюють INT 13h і чекають завантаження DOS, потім перехоплюють INT 21h і приступають до зараження. Містять текст:

THE PREDATOR. TORPNACSAELCFASVVAPC.
VANOCED

Останній рядок містить частини імен файлів (задом наперед), які не вражаються вірусом PROT, SCAN, CLEA, VSAF, CPAV, NAV, DECO.

Також містять рядки:

"Predator.2248" Predator virus #2 (c) 1993 Priest - Phalcon/Skism

"Predator.2424" Predator virus #2 (c) 1993
Here comes the Predator!

***Samara.1536.*** Безпечний резидентний файлово-завантажувальний поліморфік-вірус. При старті з інфікованого файла заражає MBR вінчестера, перехоплює INT 21h і записується в кінець COM- і EXE-файлів при їх запуску (окрім COMMAND.COM). Забороняє запуск антивірусів AVPLITE, AIDSTEST, AVP, DRWEB, SCAN.

При завантаженні з інфікованого MBR вірус перехоплює INT 13h, чекає завантаження DOS і потім перехоплює INT 21h. При завантаженні з boot-сектора дискети вірус ще заражає MBR. При зараженні MBR і boot-секторів не зберігає їх оригінали. Для збереження працездатності системи вірус при завантаженні із зараженого диска самостійно прочитує і запускає на виконання перший логічний сектор диска C, який містить завантажувальний код операційної системи.

***OneHalf, сімейство***. Дуже небезпечні резидентні файлово-завантажувальні поліморфік-віруси. При запуску заражають MBR вінчестера, при завантаженні з ураженого диска перехоплюють INT 13h, 1Ch, 21h і записуються в COM- і EXE-файли при зверненні до них. Не заражають файли SCAN, CLEAN, FINDVIRU, GUARD, NOD, VSAFE, MSAV, CHKDSK.

Код-розшифровувач цих вірусів розкиданий по всьому файла з випадковими зсувами. При зараженні вінчестера вірус прочитує його MBR і сканує таблицю розділів диска (DPT). У ній він шукає останній DOS'івський диск – логічний диск (FAT-12,16/BIGDOS) або Extended partition, і коли знаходить, підраховує номер першого і останнього циліндрів знайденого диска (або Extended partition). При цьому вірус досить грамотно обробляє диски, що мають більше 1024 циліндрів і не вписуються в стандарти INT 13h. Вірус запам'ятовує адреси цих циліндрів і заражає вінчестер.

Потім при завантаженні із зараженого вінчестера вірус шифрує два останні циліндри диска, при наступному завантаженні – ще два і т.д., поки не дійде до першого циліндра. При цьому вірус використовує адреси першого і останнього циліндрів диска, які запам'ятав при зараженні вінчестера. Коли кількість зашифрованих циліндрів перевалить за половину диска, вірус повідомляє (залежно від поточної дати і свого "покоління"):

Disk is one half.

Press any key to continue...

Таким чином, чим частіше перезавантажується заражений комп'ютер, тим більше дані виявляються зашифрованими. Після завантаження в пам'ять вірус розшифровує/зашифровує ці сектори "на льоту", тому користувач не помічає того, що його дані зіпсовані. Проте, якщо вилікувати MBR, то всі дані виявляються втраченими.

"OneHalf.3518" не шифрує себе у файлах. Виводить текст:

A20 Error !!! Press any key to continue ...

"OneHalf.3544.b" не заражає файли AIDS*.*, ADINF*.*, DRWEB*.*, ASD*.*, MSAV*.*. Виводить повідомлення:

Dis is TWO HALF. Fucks any key to Goping...

***Cheeba, сімейство.*** Резидентні безпечні віруси. Активізуються, коли вектор INT 13h вказує на область з адресою меншою, ніж адреса першого MCB. В обробниках INT 13h, 21h, 22h замінюють перші 5 байтів на код "FAR JMP на тіло вірусу",

потім записуються в кінець COM- і EXE-файлів. Містять текст:

CHEEBA Makes Ya High Harmlessly F**K THE LAMERS.

У вірусі присутні також коди, які розшифровуються і виконуються при відкритті файла USERS.BBS, використовуючи ім'я файла як ключ розшифровки. При цьому вірус записує у файл USERS.BBS якусь інформацію (створює ім'я з максимальними привілеями?).

***Bomber.*** Нешкідливий резидентний поліморфік-вірус. Перехоплює INT 21h і заражає COM-файли, окрім COMMAND.COM, при їх запуску. Містить усередині себе текст:

COMMANDER BOMBER WAS HERE. [DAME]

Характерною рисою цього вірусу є те, що він використовує досить незвичайний поліморфік-алгоритм. При зараженні вірус прочитує 4096 байтів з середини файла і переносить їх в його кінець. Себе він записує в "диру", що утворилася, і приступає до генерації поліморфік-коду. Вірус містить декілька підпрограм генерації випадкового (але цілком працездатного!) коду, який записується у випадкові місця файла, що заражається. У цьому коді може бути присутньою близько 90% всіх інструкцій процесора i8086. Управління з однієї ділянки в іншу передається командами CALL, JMP, RET, RET xxxx. Перша ділянка записується в початок файла, а остання передає управління на основне тіло вірусу. У результаті заражений файл виглядає як би покритий "плямами" коду вірусу, а процедура виявлення основного тіла вірусу стає надзвичайно складною.

## 9. Способи захисту від вірусів

Насамперед, необхідно відзначити, що захистити комп'ютер від вірусів може лише сам користувач. Тільки систематичне архівування інформації, обмеження ненадійних контактів і своєчасне застосування антивірусних засобів може захистити комп'ютер від зараження або забезпечити мінімальний збиток, якщо зараження все-таки відбулося.

### Систематичне архівування важливої інформації

Єдиним стовідсотковим по надійності методом захисту від втрати важливої інформації є її резервне копіювання на

захищені від записування пристрої зберігання даних. Більше того, архівуванням також не можна нехтувати, оскільки втратити інформацію можна не лише через віруси, але й через стрибки напруги в мережі, поломки обладнання й т.д.

Жодна антивірусна програма не зрівняється по надійності з архівуванням інформації. Справа в тому, що на будь-який алгоритм антивірусу завжди найдеться алгоритм вірусу, невидимого для цього антивірусу.

**Обмеження ненадійних контактів**

Друге правило, що частково гарантує збереження інформації, – це обмеження копіювання даних з ненадійних джерел. Як би ми не старалися, обмін інформацією з іншими користувачами і робота в локальних або глобальних мережах неминучі. Однак, деякі правила для себе все-таки виділити можна.

По-перше, необхідно намагатися не запускати неперевірені файли, у тому числі отримані по комп'ютерній мережі. Бажано використовувати тільки програми, отримані з надійних джерел. Перед запуском нових програм обов'язково варто перевірити їх одним або декількома антивірусами.

По-друге, варто обов'язково користуватися тільки тими джерелами та іншими файлами, які добре зарекомендували себе, хоча це не завжди рятує (наприклад, на WWW-сервері Microsoft досить довгий час перебував документ, заражений макровірусом “Wazzu”).

По-третє, необхідно обмежити коло людей, які допущені до роботи на конкретному комп'ютері. Практика показує, що найбільш уразливі комп'ютери – багатокористувацькі.

І нарешті, відповідно до четвертого правила, варто купувати тільки дистрибутивне програмне забезпечення в офіційних продавців. Безкоштовні, умовно безкоштовні або піратські копії можуть призвести до зараження.

**Використання антивірусних програм**

При існуючому різноманітті вірусів і їх мутацій запобігти зараженню може тільки повнофункціональна антивірусна система, що має в своєму арсеналі всі відомі технології

боротьби з «інфекційними хворобами»: не тільки сканер-поліфаг, але і резидентний on-line-монітор, засоби контролю програмної цілісності (CRC) і евристичного пошуку вірусних сигнатур.

Кожен новий вірус необхідно знайти щонайшвидше (а деякі віруси навмисно довго себе не проявляють, щоб у них було досить часу на розповсюдження). Проблема у тому, що немає чіткого способу визначити наперед, що при своєму виконанні дана програма проявить вірусоподібну поведінку. Як немає єдиних ліків від усіх хвороб, так немає універсальної «вакцини» від усіх видів шкідливого програмного забезпечення. На всі 100% захиститися від вірусів практично неможливо!

У такій сфері, як виявлення атак на комп'ютерні системи, процес вдосконалення нескінченний. Хакери не втомлюються винаходити все нові схеми проникнення в комп'ютерні системи. Розробники детектуючих додатків, що стоять по іншу сторону барикад, відстежують новинки, що з'являються, і поспішають запропонувати свої контрзаходи. От чому продукти, що випускаються, вимагають постійної модернізації, і користувачам настійно рекомендується встановлювати оновлені сигнатури, що дозволяють ідентифікувати нові види мережевих атак.

Старе антивірусне програмне забезпечення подібно лікам з минулим терміном придатності – толку від нього мало. Якщо не обновляти файли сигнатур, то рано чи пізно можна опинитися беззахисними проти нових вірусів. Більшість фірм, що розробляють антивіруси, випускають нові файли сигнатур принаймні двічі в місяць або й частіше, якщо з'являється серйозний вірус. Для отримання нових сигнатур зручно користуватися функцією автоматичного оновлення через Web, що є в антивірусному пакеті.

Супровід через Internet програми PC-cillin, наприклад, володіє унікальною особливістю. Можна не тільки отримати консультацію по електронній пошті, але і поговорити у реальному часі з фахівцем служби супроводу Trend. (Хоча від цієї чудової послуги не багато толку, якщо комп'ютер заблокований і увійти в Internet неможливо, проте вона

безкоштовна.) Такі антивірусні продукти, як Norton AntiVirus 2000 і McAfee VirusScan, підтримують найкрупніші дослідницькі групи галузі: відповідно Symantec AntiVirus Research Center і AntiVirus Emergency Response Team. Тому Norton і McAfee швидко реагують на загрозу нового вірусу.

Всі основні фірми-постачальники антивірусного забезпечення регулярно і досить часто оновлюють файли сигнатур вірусів, а при появі особливо шкідливого вірусу створюють додатковий екстрений випуск. Ще зовсім недавно вважалося, що сигнатури потрібно оновлювати щомісячно, але в нашу епоху нових вірусів, можливо, буде розумним перевіряти їх щотижня вручну або за допомогою автоматичного оновлення антивірусної програми. В утилітах McAfee, Symantec і Trend Micro для оновлення достатньо один раз клацнути кнопкою миші.

Певний набір засобів антивірусного захисту присутній у всіх утилітах основних фірм-виробників програмного забезпечення. Серед них: постійний захист від вірусів (антивірусний монітор), перевірка системи за розкладом і оновлення сигнатур через Internet, а також створення аварійної завантажувальної дискети, що дозволяє запустити комп'ютер навіть тоді, коли у нього заражений вірусом завантажувальний сектор (природно, дискету треба створити до того, як вірус потрапив в комп'ютер). Крім цих стандартних засобів, деякі пакети містять «архітектурні надмірності»: наприклад, спеціальний додатковий захист від поштових вірусів (тривога з приводу яких наростає), а також шкідливих модулів ActiveX і Java-аплетів. А такі програми, як Panda Antivirus Platinum і PC-cillin, навіть дозволяють батькам заблокувати доступ дітей до небажаних Web-сторінок.

Оскільки у нових вірусів є нові сигнатури, файли сигнатур необхідно підтримувати в актуальному стані. При виході нової версії антивіруса формат файла сигнатур звичайно міняється, і оновлені сигнатури виявляються несумісними з попередніми версіями програми. Саме тому антивірусне програмне забезпечення вже досить давно продається по тій же схемі, що

бритви і леза: одного разу купивши основну утиліту (бритву), ви потім вимушені постійно купувати оновлені файли сигнатур (леза).

Так, компанії McAfee і Symantec надають право необмеженого оновлення сигнатур протягом року з моменту придбання утиліти, але за кожен наступний рік потрібно в обох випадках заплатити 4 долари. Таку суму навряд можна вважати серйозним ударом по кишені (на відміну від підписки на оновлені файли сигнатур F-Secure, яка коштує 63 доларів); крім того, через рік ми з великою вірогідністю захочемо відновити саму програму. Їх основні конкуренти – Command AntiVirus, Inoculate IT, Panda Antivirus Platinum і PC-cillin – пропонують безкоштовне оновлення сигнатур протягом всього життя продукту.

В даний час способи надання антивірусного захисту істотно змінюються. Компанія McAfee.com вже пропонує перевірку на віруси через Internet в своїй «електронній лікарні» McAfee Clinic (разом з ще декількома видами діагностики). Послуга надається по підписці і коштує 50 доларів в рік, але часто з'являються спеціальні пропозиції, а за перші два тижні платня не береться – це випробувальний період. Перевірку віддалених комп'ютерів на віруси здійснює модуль ActiveX, який бере сигнатури з Web-серверу виробника програми.

**Види антивірусних програм**

Самими популярними й ефективними антивірусними програмами є антивірусні сканери, монітори, фаги (поліфаги), ревізори. Застосовуються також різного роду блокувальники і іммунізатори (вакцини). Розглянемо характеристики кожного з цих видів програм.

***Сканери (scanner).*** Сканери (детектори) здатні виявити фіксований набір суттєвих вірусів у файловій системі, секторах і системній пам’яті, а потім – негайно видалити більшість з них. Для пошуку вірусів сканери використовують так звані "маски" (або сигнатуру) – деяку постійну послідовність коду, специфічну для конкретного вірусу.

У випадку, якщо вірус не містить у собі постійної маски

(наприклад, поліморфік-вируси), використовуються інші методи, засновані на описі всіх можливих варіантів коду на алгоритмічній мові.

У багатьох популярних сканерах (наприклад Антивірус Касперського, Doctor Web, Norton Antivirus, McAfee, Panda Antivir, AntiVir Personal Edition і ін.) застосовується режим евристичного сканування. Цей режим полягає в тому, що програма не просто шукає віруси, а проводить аналіз послідовності команд у кожному об'єкті, який перевіряється, здійснює набір деякої статистики, згодом приймає ймовірне рішення типу: "можливо заражений" або "не заражений".

Евристичне сканування являє собою ймовірнісний метод пошуку вірусів, що, в решті решт, забезпечує можливість визначення невідомих програмі вірусів, але разом з цим збільшує кількість помилкових спрацьовувань (повідомлень, знайдених вірусах у файлах, де насправді їх немає).

Основна ідея такого підходу полягає у тому, що евристика спочатку розглядає поведінку програми, а потім зіставляє його з характерним для зловмисної атаки, на зразок поведінки троянського коня. Встановити модель поведінки і ухвалити рішення щодо нього можна за допомогою декількох механізмів. Для того, щоб виявити і визначити всі можливі дії програми, використовують два підходи: сканування і емуляція.

Підхід зі скануванням припускає пошук «поведінкових штампів», наприклад, найтиповіших низькорівневих способів відкриття файлів. Або процедура сканування звичайного виконуваного файла проглядає всі місця, де програма відкриває інший файл, і визначає, якого роду файли вона відкриває і що в них записує.

Другий метод визначення поведінки – емуляція. Такий підхід дещо складніший. Програма пропускається через емулятор Windows або макроемулятор Macintosh або Word з метою подивитися, що вона робитиме. Проте виникають питання, тому що в цьому випадку багато що залежить від чудасій вірусів. Наприклад, якщо вірус запрограмований на форматування жорсткого диска 25 лютого о 10 годині ранку, а

при емуляції цього вірусу на симуляторі дата встановлена на 24 лютого, то вірус поки не проявить свої наміри.

Вся хитрість швидкого розпізнавання полягає в поєднанні двох підходів і отриманні найдокладнішого каталогу поведінкових штампів за можливо коротший час. Для перевірки факту зараження файла вірусом фахівці можуть використовувати різні варіанти штучного інтелекту – експертні системи і нейронні мережі.

Недолік евристичного підходу полягає якраз в його евристичності. Завжди є вірогідність, що надзвичайно підозрілий файл насправді абсолютно нешкідливий. Проте останній евристичний механізм Symantec під назвою Bloodhound дозволяє знайти до 80% невідомих вірусів виконуваних файлів і до 90% невідомих макровірусів. Варто також помітити, що програми-детектори не дуже універсальні, оскільки здатні знайти тільки відомі віруси. Деяким таким програмам можна повідомити спеціальну послідовність байт, характерну для якогось вірусу, і вони зможуть знайти інфіковані ним файли: наприклад, це уміють NotronAntiVirus або AVP-сканер.

Різновидом сканерів є так звані таблетки – спеціалізовані програми, орієнтовані на пошук певного типу або сімейства вірусів, наприклад, троянів, макровірусів та інших (наприклад, Anti-Trojan, Trojan Remover).

Слід зазначити, що використання спеціалізованих сканерів, розрахованих тільки на макровіруси, іноді буває більше зручним і надійним рішенням для захисту документів MS Word і MS Excel.

До недоліків сканерів варто віднести тільки те, що вони охоплюють далеко не всі відомі віруси й вимагають постійного відновлення антивірусних баз. З огляду на частоту появи нових вірусів і їх короткий життєвий цикл, для використання сканерів необхідно налагодити одержання свіжих версій не рідше одного-двох разів на місяць. В іншому випадку їхня ефективність істотно знижується.

***Монітори.*** **Монітори** – це різновид сканерів, які, постійно перебуваючи в пам'яті, відслідковують вірусоподібні ситуації,

які відбуваються з диском і пам'яттю (тобто виконують безперервний моніторинг). Прикладом таких антивірусів може бути програма Kaspersky Anti-Virus або SpiDer Guard.

До недоліків цих програм можна віднести, наприклад, імовірність виникнення конфліктів з іншим програмним забезпеченням, як і для сканерів – залежність від нових версій вірусних баз, а також можливість їхнього обходу деякими вірусами.

***Фаги (поліфаги) (scanner/cleaner, scaner/remover*). Фаги** – це програми, здатні не тільки знаходити, але і знищувати віруси, тобто лікувати «хворі» програми (поліфаг може знищити багато вірусів). До поліфагів відноситься і така стара програма, як Aidstest, яка знаходить і знешкоджує близько 2000 вірусів.

Основний принцип роботи традиційного фага простий і не є секретом. Для кожного вірусу шляхом аналізу його коду, способів зараження файлів і т.д. виділяється деяка характерна тільки для нього послідовність байтів – сигнатура. Пошук вірусів в простому випадку зводиться до пошуку їх сигнатур (так працює будь-який детектор).

Сучасні фаги використовують інші методи пошуку вірусів. Після виявлення вірусу в тілі програми (або завантажувального сектора, який теж містить програму початкового завантаження) фаг знешкоджує його. Для цього розробники антивірусних засобів ретельно вивчають роботу кожного конкретного вірусу: що він псує, як він псує, де він ховає те, що зіпсує (якщо ховає). В більшості випадків фаг може видалити вірус і відновити працездатність зіпсованих програм. Але необхідно добре розуміти, що це можливо далеко не завжди.

***Ревізори.* Ревізори** – це програми, принцип роботи яких заснований на підрахунку контрольних сум (CRC-сум) для присутніх на диску файлів і системних секторів.

Прикладом такого антивірусу може бути програма ADinf32. Ці контрольні суми потім зберігаються в базі даних антивірусу (у таблицях) разом із відповідною інформацією: довжинами файлів, датами їх останньої модифікації і т.д. При наступному запуску ревізори звіряють відомості, що містяться в

базі даних, з реально підрахованими значеннями. Якщо інформація про файл, записана в базі даних, не збігається з реальними значеннями, то ревізор попереджає про те, що файл, можливо, був змінений або заражений вірусом.

Ревізори вміють вчасно виявляти зараження комп'ютера практично кожним з існуючих на сьогодні вірусів, не допускаючи розвитку епідемії, а сучасні версії ревізора вміють негайно видаляти більшість навіть раніше незнайомих їм вірусів.

До недоліків ревізорів можна віднести те, що для забезпечення безпеки вони повинні використовуватися регулярно. Але безсумнівними їхніми перевагами є висока швидкість перевірок і те, що вони не вимагають частого відновлення версій.

***Антивірусні блокувальники***. **Антивірусні блокувальники** – це резидентні програми, які перехоплюють небезпечні ситуації, і повідомляють про це користувача (наприклад, AVP Office Guard). До ситуацій, що відслідковуються, належать, наприклад, відкриття виконуваних файлів для записування і записування в boot-сектори дисків або MBR вінчестера, спроби програми залишитися резидентною і т.д. До речі, відзначені події характерні для вірусів у моменти їх розмноження.

Блокувальники дозволяють обмежити розповсюдження епідемії, поки вірус не буде знищений. Практично всі резидентні віруси визначають факт своєї присутності в пам'яті машини, викликаючи яке-небудь програмне переривання з «хитрими» параметрами. Якщо написати просту резидентну програму, яка імітуватиме наявність вірусу в пам'яті комп'ютера, правильно «відзиваючись» на певний пароль, то вірус, швидше за все, визнає цю машину вже зараженою.

Навіть якщо деякі файли на комп'ютері містять в собі код вірусу, при використанні блокувальника зараження всієї решти файлів не відбудеться. Для нормальної роботи такої програми необхідно запустити блокувальник раніше всієї решти програм, наприклад, у файлі CONFIG.SYS. Але якщо вірус встиг заразити COMMAND.COM або стартує із завантажувального сектора, то

антивірус-блокувальник не допоможе.

До переваг блокувальників можна віднести вміння виявляти вірус на самій ранній стадії його розмноження, а до недоліків – здатність деяких вірусів обходити блокувальники, а також наявність неправдивих спрацьовувань.

***Імунізатори або вакцини.*** **Імунізатори** – це невеликі програми, які змінюють файли або проникають у них. У першому випадку вірус буде приймати файли як заражені, а в другому – антивірус буде щоразу перевіряти файл на зміни. Слід зазначити, що в наш час цей тип антивірусів не має великого розповсюдження серед користувачів.

Спеціальні вакцини призначені для обробки файлів і завантажувальних секторів. Вакцини бувають пасивними і активними.

Активна вакцина, «заражаючи» файл, подібно вірусу, оберігає його від будь-якої зміни і у ряді випадків здатна не тільки знайти сам факт зараження, але і вилікувати файл. Пасивні вакцини застосовують тільки для запобігання зараженню файлів деякими вірусами, що використовують прості ознаки їх зараженості – «дивні» час або дату створення, певні символьні рядки і т.д. В даний час вакцинація широко не застосовується. Бездумна вакцинація всього і всіх здатна викликати цілі епідемії неіснуючих вірусних хвороб. Так, протягом декількох років на території колишнього СРСР лютувала страшна епідемія жахливого вірусу TIME. Жертвою цього вірусу стали сотні абсолютно здорових програм, оброблених антивірусною програмою ANTI-KIT.

Наведемо приклад. В даний час існує досить багато вірусів, що запобігають повторному зараженню файлів деякою «чорною міткою», якою вони мітять інфіковану програму. Існують, наприклад, віруси, що виставляють в полі секунд часу створення файла значення 62.Уже досить давно з'явився вірус, який до всіх заражених файлів дописував п'ять байт – MsDos. Нормальних файлів, що містять в кінці такий символьний рядок, не буває, тому вірус і використовував цю ознаку як індикатор зараження файла. Вакцинація файлів проти такого вірусу зовсім не

складна. Достатньо дописати в кінець вище згаданий символьний рядок – і зараження таким вірусом не страшне. Страшне інше – деякі антивірусні програми, зустрівши в кінці файла нещасливий рядок, починають негайно лікувати його. Шансів на те, що після такого «лікування» «інвалід» нормально працюватиме, практично ніяких.

### Контрольні питання до Лекції 4

1. Класифікуйте комп'ютерні віруси за середовищем їх існування та за способом зараження комп'ютерів.
2. Наведіть класифікацію вірусів за алгоритмами, які вони використовують при функціонуванні, та за своїми деструктивними можливостями?
3. З чого складається класифікаційний код вірусу?
4. Що таке дескриптор та сигнатура вірусів?
5. У чому полягають особливості файлових вірусів, якими вони бувають?
6. Де можуть бути розташовані файлові віруси?
7. Яким буває класифікаційний код файлового вірусу і які його складові?
8. Охарактеризуйте дескриптор та сигнатуру файлового вірусу.
9. Які різновиди файлових вірусів ви знаєте? Дайте характеристику "overwriting"-вірусам?
10. У чому полягає принцип функціонування та розташування паразитичних вірусів?
11. У чому різниця між вірусами типу "prepending", "appending" і "inserting"?
12. Як працюють віруси-компаньйони? В чому їх особливості?
13. Що таке файлові хробаки? Наведіть відомі вам приклади файлових вірусів-хробаків.
14. Link-віруси та їх особливості.
15. Охарактеризуйте групу OBJ- і LIB-вірусів. Де вони розташовуються і як себе проявляють?
16. Наведіть алгоритм роботи файлових вірусів.

17. Які можливості відновлення файлів, що заражені файловим вірусом?

18. Дайте означення завантажувального вірусу. В чому його особливості?

19. Яким може бути класифікаційний код бутівського вірусу? Наведіть приклади і поясніть значення кожної складової.

20. Що може містити дескриптор та сигнатура бутівського вірусу?

21. У чому полягає принцип дії завантажувального вірусу?

22. Наведіть можливі місця розташування завантажувальних вірусів.

23. Наведіть алгоритм роботи завантажувального вірусу: резидентного і нерезидентного.

24. Дайте загальну характеристику макро-вірусам, їх особливостям та розташуванню.

25. Якими можуть бути причини зараження макро-вірусами?

26. Які принципи роботи і алгоритм функціонування Word-вірусів?

27. Які принципи функціонування Excel-вірусів?

28. Як працюють Access-віруси і як вони себе проявляють?

29. У чому особливості мережних вірусів?

30. Що таке IRC-хробаки і які їх різновиди ви знаєте?

31. Що таке IRC-сервери та IRC-клієнти?

32. Охарактеризуйте стелс-віруси. Які різновиди цих вірусів ви знаєте?

33. Як проявляють себе завантажувальні стелс-віруси?

34. У чому особливість файлових стелс-вірусів?

35. Що таке макро-стелс-віруси?

36. Які віруси відносять до групи поліморфічних вірусів?

37. Наведіть і охарактеризуйте рівні поліморфік-вірусів і можливості детектування вірусу на кожному з рівнів.

38. Дайте характеристику такому способу поліморфізму, як зміна виконуваного коду.

39. Що таке поліморфні розшифровувачі?

40. Наведіть перелік основних прийомів для захисту

операційних систем від вірусів.

41. Доведіть необхідність систематичного архівування інформації та обмеження ненадійних контактів

42. Для чого існує антивірусне програмне забезпечення?

43. Що таке програми-сканери та які їх основні характеристики?

44. У чому полягає особливість таких антивірусних програм, як таблетки?

45. Яка особливість антивірусних програм-моніторів?

46. Що таке програми-ревізори, як вони працюють?

47. Дайте характеристику антивірусним блокувальникам та програмам-імунізаторам.

## Лекція 5

# ПАКУВАННЯ, АРХІВАЦІЯ І ШИФРУВАННЯ ДАНИХ В ОПЕРАЦІЙНИХ СИСТЕМАХ

**План лекції:**

1. Історичні відомості.
2. Стискання файлів під Windows 9x\NT.
3. Продуктивність пакування файлів.
4. Принципи роботи програм-архіваторів.

## 1. Історичні відомості

У ті далекі часи, коли обсяг жорстких дисків (вінчестерів) вимірювався мегабайтами – цих мегабайт завжди не вистачало, і більшість файлів (особливо рідко використовуваних) зберігали в упакованому вигляді. Перед запуском файл розпаковували, а після завершення роботи – упаковували знову, щоб звільнити місце для розпакування інших.

Коли ці махінації всім остаточно набридли, програмісти (згадавши, що комп'ютер повинен служити людині, а не навпаки), додумалися до автоматичного розпакування файлів, що виконуються "на льоту". Ідея полягає в дописуванні до стиснутого файла крихітного розпаковщика, якому передається керування при запуску файла, і який розпаковує код, що виконується, не на диск, а безпосередньо в оперативну пам'ять. Звичайно, час завантаження при цьому відчутно збільшувався (особливо на машинах з повільними процесорами), але це з надлишком виправдовувалося простотою запуску й економією дискового простору.

Незабаром пакувальників розвелося сила-силенна (їх тоді писали всі, кому хотілося) – AINEXE, DIET, EXEPACK, LZEXE, PKLITE і інших – усіх не перелічити! І не дивно: процесори з дня у день ставали усе продуктивнішими – уже на "трійці" розпакування займало настільки незначний час, що їм можна було зневажити. До того ж приємним побічним ефектом виявився захист від дизасемблювання. Дійсно, безпосередньо

дизасемблювати упакований файл неможливо – колись його необхідно розпакувати. Звичайно, на кожен щит знайдеться свій меч – з-під пера хакерів вийшло чимало чудових універсальних розпаковщиків (UNP, Intruder, UUP, а вершиною усьому став CPU386 з вбудованим емулятором реального режиму процесора 80386), але якість автоматичного розпакування залишала бажати кращого (часом розпаковані файли зависали при запуску чи в процесі роботи), а ручним трасуванням володіли далеко не всі.

Словом, при усіх своїх достоїнствах, пакування виконуваних файлів не мало ніяких недоліків і не збиралося здавати позиції навіть із приходом ємних (на той час) одно-, двогігабайтних дисків і CD-ROM.

## 2. Стискання файлів під Windows 9x\NT

Пройшов невеликий час і світ повільно, але неминуче переходив на нову операційну систему – Windows 95. Користувачі обережно освоювали мишу і графічний інтерфейс, а програмісти тим часом гарячково переносили старе програмне забезпечення на нову платформу. Обсяги вінчестерів на той час виросли настільки, що розробники могли забути слово "оптимізація", так вони, судячи з розміру сучасних додатків, його і забули. Сто мегабайт туди, триста сюди, – запросто.

Тоді і згадали про розпакування виконуваних файлів "на льоту".

### Стискання виконуваних файлів.

На ринку з'явилося декілька програм-компресорів, з яких найбільшу популярність завоювала програма ASPack, що вміє стискати і розтискати не тільки "екзешники", але і динамічні бібліотеки. А до складу самої Windows 95 увійшла динамічна бібліотека LZEXPAND.DLL, яка підтримувала базові операції пакування-розпакування і "прозору" роботу зі стиснутими файлами. Користувачі і програмісти швидко скористалися новими засобами, але...

На відміну від MS-DOS, у Windows 9x\NT за автоматичне розпакування приходиться платити більше, ніж одержувати. Згадаємо, як у MS-DOS відбувалося завантаження виконуваних

модулів. Файл цілком зчитувався з диска і копіювався в оперативну пам'ять, причому найбільш вузьким місцем була саме операція читання з диска. Пакування навіть прискорювало завантаження, оскільки фізично читався менший обсяг даних, а їх розпакування займало дуже короткий час.

У Windows же завантажник читає лише заголовок і таблицю імпорту файла, а потім проектує його на адресний простір процесу так, ніби-то файл є частиною віртуальної пам'яті, що зберігається на диску (взагалі ж, все відбувається набагато складніше). Підкачування з диска відбуваються динамічно – у міру звернення до відповідних сторінок пам'яті, причому завантажуються тільки ті з них, що дійсно потрібні.

Наприклад, якщо в текстовому редакторі є модуль роботи з таблицями, він не буде завантажений з диска доти, поки користувач не захоче створити (чи відобразити) свою таблицю. Причому неважливо – чи знаходиться цей модуль у динамічній бібліотеці, чи в основному файлі. Завантаження таких "монстрів", як Microsoft Visual Studio і Word, ніби "розмазується" у часі, і до роботи з додатком можна приступати практично відразу ж після його запуску. А що ж відбудеться, якщо файл упакувати? Він повинен буде зчитуватися з диска цілком (!) і потім – знову-таки, цілком – розпакуватися в оперативну пам'ять.

Але ж нашої оперативної пам’яті явно не вистачить і розпаковані сторінки прийдеться знову скидати на диск. Причому, якщо при проектуванні не упакованого exe-файла оперативна пам'ять не виділяється (у всякому разі, доти, поки в ній не виникне необхідність), розпаковщику без пам'яті ніяк не обійтися. А оскільки оперативної пам'яті ніколи не буває занадто, вона може бути виділена лише за рахунок інших додатків. Відзначимо також, що в силу конструктивних особливостей заліза й архітектури операційної системи, операція записування на диск є помітно повільнішою за операцію зчитування.

Важливо зрозуміти: Windows ніколи не скидає на диск не модифіковані сторінки файла, що проектуються. Навіщо це?

Адже в будь-який момент їх можна знову зчитати з оригінального файла. Але при розпакуванні модифікуються всі сторінки файла! Виходить, система буде змушена "ганяти" їх між диском і пам'яттю, що істотно знизить загальну продуктивність усіх додатків у цілому.

**Стискання динамічних бібліотек**

Ще більші накладні витрати спричиняє стискання динамічних бібліотек. Для економії пам'яті сторінки, зайняті динамічною бібліотекою, спільно використовуються всіма процесами, що завантажили цю DLL. Але як тільки один із процесів намагається щось записати в пам'ять, зайняту DLL, система миттєво створює копію сторінки, що модифікується, і надає її в "монопольне" розпорядження процесу - “письменнику”.

Оскільки розпакування динамічної бібліотеки відбувається в контексті процесу, що завантажив її, система змушена багаторазово дублювати всі сторінки пам'яті, виділені бібліотеці, фактично надаючи кожному процесору свій власний екземпляр DLL. Припустимо, одна DLL розміром у 1 Мегабайт, була завантажена десятьма процесами – порахуємо, скільки пам'яті буде дарма втрачено, якщо вона стиснута!

Таким чином, під Windows 9x\NT стискати виконувані файли недоцільно – ми платимо набагато більше, ніж отримуємо. Що ж стосується захисту від дизасемблювання, то, коли ASPack тільки з'явився, він віднадив від зламу дуже багатьох некваліфікованих хакерів, але ненадовго. Сьогодні в мережі легко можна знайти посібники з так званого ручного зняття ASPack. Існує і маса готового інструментарію – від автоматичних роз пакувальників до плагінів для дизасемблера IDA Pro, що дозволяють йому дизасемблювати стиснуті файли. Тому сподіватися, що ASPack цілком врятує нашу програму від зламу, не слід.

## 3. Продуктивність пакування файлів

Стосовно вимірювання зменшення продуктивності від пакування файлів, то тут, здавалося б, немає нічого складного –

беремо не упакований файл, запускаємо його, заміримо час завантаження, записуємо результат на папірці, упаковуємо, запускаємо ще раз, і...

Перший камінь спотикання – що розуміти під "часом завантаження"? Якщо проектування – так воно виконується практично миттєво, і їм можна взагалі зневажити. Момент часу, починаючи з якого з програмою можна повноцінно працювати? Так це від самої програми залежить більше, ніж від його упакування. До того ж, на час завантаження упакованих файлів дуже сильно впливає кількість вільної на момент запуску фізичної оперативної пам'яті (не слід плутати з загальним обсягом пам'яті, установленої на машині). Якщо перед запуском упакованого файла ми завершимо один-два великих програмних додатки-"монстри", то зайнята ними пам'ять виявиться вільною, і зможе безперешкодно використовуватися розпаковщиком. Але якщо вільної пам'яті немає, її прийдеться по крихтах відривати від інших додатків...

Навіть якщо ми оцінимо зміну часу завантаження (що, до речі, зробити дуже проблематично – серія вимірів на одній і тій самій машині, з тим самим набором додатків дає розкид результатів більш ніж на порядок), як вимірювати падіння продуктивності інших додатків? Адже, при недостачі пам'яті Windows у першу чергу рятується від не модифікованих сторінок, які немає необхідності зберігати на диску! У результаті пакування виконуваного файла можна дещо підвищити продуктивність роботи самого цього файла, але значно погіршити стан інших, не упакованих додатків.

Тому ніяких конкретних цифр навести не можна. Наближені оцінки, виконані "на око", показують, що при наявності практично необмеженої кількості оперативної пам'яті втрати продуктивності складають менше 10%, але при її недостачі швидкість усіх додатків падає від двох до десяти разів! В експерименті брали участь файли, що виконуються, MS Word 2000, Visual Studio 6.0, Free Pascal 1.04, IDA Pro 4.17, Adobe Acrobat Reader 3.4, машина з процесором CLERION-300A, оснащена 256 МБ ОЗУ, для імітації недостачі пам'яті її

обсяг зменшувався до 64 МБ; використовувалися операційні системи Windows 2000 і Windows 98.

Таким чином, можна зробити такі висновки та рекомендації:

1) Файли, що виконуються під Windows, краще не пакувати. У крайньому випадку – використовувати для пакування/розпакування функції операційної системи (LZInit, LZOpenFile, LZRead, LZSeek, LZClose, LZCopy), динамічно розпаковуючи в спеціально виділений буфер тільки ті частини файла, що дійсно потрібні в даний момент для роботи.

2) Динамічні бібліотеки взагалі не слід пакувати, оскільки це веде до дивовижної витрати і фізичної, і віртуальної пам'яті і псує саму концепцію DLL: один модуль – усім процесам.

3) Не слід прагнути базувати свій додаток на безлічі DLL, – сторінки виконуваного файла не вимагають фізичної пам'яті доти, поки до них не відбувається звернення. Тому сміливо можна поміщати весь код програми в один файл.

## 4. Принципи роботи програм-архіваторів

Принцип роботи архіваторів заснований на пошуку у файлі "надлишкової" інформації і наступному її кодуванні з метою одержання мінімального обсягу.

Стискання послідовностей однакових символів – найвідоміший метод архівації файлів. Наприклад, усередині нашого файла знаходяться послідовності байтів, що часто повторюються. Замість того, щоб зберігати кожен байт, фіксується кількість повторюваних символів і їхня позиція. Наприклад, файл, що буде архівуватися, займає 15 байт і складається з наступних символів:

B B B B B L L L L L A A A A A

або у шістнадцятковій системі

42 42 42 42 42 4C 4C 4C 4C 4C 41 41 41 41 41 .

Архіватор може представити цей файл у такому вигляді (шістнадцятковому):

01 05 42 06 05 4C 0A 05 41 .

Це означає: з першої позиції п'ять разів повторюється

символ "B", з позиції 6 п'ять разів повторюється символ "L" і з позиції 11 п'ять разів повторюється символ "A". Для збереження файла в такій формі буде потрібно всього 9 байт, що на 6 байт менше вихідного.

Описаний метод є простим і дуже ефективним способом стискання файлів. Однак він не забезпечує великої економії обсягу, якщо текст містить невелику кількість послідовностей повторюваних символів.

Більш витончений метод стискання даних, використовуваний у тому чи іншому вигляді практично будь-яким архіватором, – це так званий оптимальний префіксний код і, зокрема, кодування символами змінної довжини (алгоритм Хаффмана). Код змінної довжини дозволяє записувати символи і групи символів, що найбільш часто зустрічаються, усього лише декількома бітами, у той час як рідкі символи і фрази будуть записані більш довгими бітовими рядками. Наприклад, у будь-якому англійському тексті буква E зустрічається частіше, ніж Z, а X і Q відносяться до тих, що найменш зустрічаються. Таким чином, використовуючи спеціальну таблицю відповідності, можна закодувати кожну букву E меншим числом біт і використовувати більш довгий код для більш рідких букв.

Популярні архіватори ARJ, PAK, PKZIP працюють на основі алгоритму Лемпела-Зіва. Ці архіватори класифікуються як адаптивні словникові кодувальники, у яких текстові рядки заміняються покажчиками на ідентичні їм рядки, що зустрічаються раніше в тексті. Наприклад, усі слова якої-небудь книги можуть бути представлені у вигляді номерів сторінок і номерів рядків деякого словника. Найважливішою відмінною рисою цього алгоритму є використання граматичного розбору попереднього тексту з розбиттям його на фрази, що записуються у словник. Покажчики дозволяють зробити посилання на будь-яку фразу у вікні встановленого розміру, що передує поточній фразі. Якщо відповідність знайдена, то текст фрази заміняється покажчиком на свого попереднього двійника.

При архівації, як і при компресуванні, ступінь стискання файлів сильно залежить від формату файла. Графічні файли

типу TIFF і GIF уже заздалегідь скомпресовані (хоча існує різновид формату TIFF і без компресії) і тут навіть найкращий архіватор мало що знайде для пакування. Зовсім інша картина спостерігається при архівації текстових файлів, файлів PostScript, файлів .BMP і їм подібних.

**Програми архівації файлів**

Архівний файл являє собою набір з одного або декількох файлів, розміщених у стисненому вигляді в одному файлі, з якого при необхідності їх можна дістати у первісному вигляді. Архівний файл містить зміст, який дозволяє побачити, які саме файли знаходяться в архіві. Для кожного стисненого і поміщеного в архів файла зберігається така інформація:

- ім’я файла;
- відомості про каталог, в якому знаходиться файл;
- дата і час останньої модифікації файла;
- розмір фала на диску і в архіві;
- код циклічного контролю для кожного файла, що використовується для перевірки цілісності архіва.

Найбільш популярні архіватори – WinZip і WinRAR – дозволяють задавати пароль на відкриття архіва. Але існують програми, за допомогою яких можна зламати архівні файли. Ці програми-зламщики архівів можна умовно розбити на дві групи:

- спеціалізовані – програми, які зламують паролі лише одного архіватора (наприклад, програма Azpr зламує пароль архіватора WinZip);
- універсальні – програми, що працюють з двома і більше видами архіваторів (наприклад, програма Archpr працює з усіма видами архівів під Windows). Недоліком таких програм є їх великий об’єм.

Більшість сучасних програм пакування даних мають вбудовану підтримку шифрування. Якщо користувач бажає захистити від чужих очей свою інформацію в архіві, йому необхідно при пакування ввести пароль, і архіватор далі сам виконає необхідні дії. При спробі видобути зашифрований файл архіватор вимагатиме у користувача пароль і розпакує файл лише тоді, коли пароль вказано правильно.

Слід зауважити, що шифрування відбувається завжди після компресування, оскільки зашифровані дані не повинні відрізнятися від випадкової послідовності і, як наслідок, архіватор не зможе знайти в них надлишковість, за рахунок видалення якої і відбувається пакування.

**ZIP**

Одним з найпопулярніших форматів стискання даних серед користувачів операційної системи Windows був і залишається ZIP, розроблений компанією PKWARE, Inc. Широкого розповсюдження цей формат набув зовсім не через технічні особливості – швидке стискання, висока ступень упаковки. Існують архівні формати, що переважають ZIP за багатьма характеристиками. Скоріш за все, формат ZIP завдячує своїй популярності умовно безкоштовній програмі WinZIP.

WinZIP є умовно безкоштовним продуктом. Будь-який користувач має право установити WinZIP і використовувати його на протязі 30 днів у тестових і ознайомлювальних цілях. При цьому програма є повнофункціональною, але іноді з'являється вікно повідомлення з пропозицією купити WinZIP. По завершенні тестового періоду необхідно отримати ліцензію або видалити програму з комп'ютера. Після оплати користувач отримує реєстраційний код, що відповідає його імені. Після введення правильного коду у відповідному вікні WinZIP програма вважається зареєстрованою і припиняє турбувати користувача пропозиціями про купівлю програми.

Для шифрування архівів формату ZIP використовується потоковий алгоритм шифрування, розроблений Роджером Шлафлай. Але у 1994 році Елі Біхем і Пол Кошер опублікували статтю, присвячену атаці на алгоритм шифрування формату ZIP, в якому для знаходження ключа шифрування досить знати 13 послідовних байтів відкритого тексту і виконати певну операцію 238 разів.

Для архівів, що містять 5 і більше файлів і створених на основі бібліотеки InfoZIP, можлива атака, що використовує в якості відкритого тексту дані з заголовків зашифрованих файлів. Цій атаці підлягли архіви, створені за допомогою програми

WinZIP, але в останніх версіях цього архіватора проблема була дещо виправлена.

Не дивлячись на те, що недоліки алгоритму шифрування ZIP давно відомі, він до цих пір лишається одним з найчастіше використовуваних для архівів формату ZIP. Деякий час тому а в архіваторах PKZIP і WinZIP з'явилась підтримка інших, більш стійких алгоритмів шифрування, але нове шифрування не дуже популярне з деяких причин. По-перше, нові формати зашифрованих даних в PKZIP і WinZIP не сумісні між собою, що не дозволяє прочитати одним архіватором те, що створено іншим. По-друге, компанія PKWARE, що створила PKZIP, звинувачує авторів WinZIP у тому, що, реалізувавши своє шифрування, вони порушують патенти, що належать корпорації PKWARE.

У 2002 році компанія PKWARE випустила версію архіватора PKZIP, яка підтримувала більш стійкі алгоритми шифрування. Але через те, що PKZIP був розрахований на корпоративних користувачів, нове шифрування не отримало достатньої популярності.

Отже, алгоритм, що використовується в WinZIP для перевірки відповідності реєстраційного коду імені користувача, давно відкритий, і в Інтернеті можна без особливих зусиль знайти початкові тексти і готові програми для обчислення цього коду. Малоймовірно, що в WinZIP Computing не знають про існування генератора кодів до їх програми, але на протязі багатьох версій схема реєстрації не змінювалась і, схоже, змінюватись не буде. Не зважаючи на порівняну простоту отримання повністю дієздатної копії WinZIP без оплати вартості ліцензії, утруднення схеми реєстрації навряд чи викличе різке збільшення об'ємів продаж. А от витрати на оновлення реєстраційних номерів у всіх легальних користувачів можуть виявитись зовсім не маленькими.

**ARJ**

Популярний з часів DOS, але рідко використовуваний в даний час архіватор, розроблений Робертом Янгом, базується на такому алгоритмі. З пароля, за дуже простим циклічним

алгоритмом отримувалась гама за довжиною рівна паролю. Ця гама накладалась на дані методом додавання за модулем 2 (операція XOR). Таким чином, наявність відкритого тексту, рівного довжині пароля, дозволяла миттєво визначити гаму і використовуваний пароль.

Більш того, на самому початку упакованих даних містилась інформація компресора така, як таблиці Хаффмана, і частина цієї інформації могла бути передбачена, що дозволяло значно підвищити швидкість пошуку пароля перебором.

Починаючи з версії 2.60 (листопад1997 року), ARJ підтримує шифрування за алгоритмом ГОСТ 28147-89.

**RAR**

Архіватор, розроблений Євгеном Рошалем, є непоганим прикладом того, як можна підходити до шифрування даних.

В алгоритмі шифрування, використаному у RAR версії 1.5, є деякі недоліки. Так, ефективна довжина ключа шифрування складає всього 64 біти, тобто перебором 264 варіантів ключа можна гарантовано розшифрувати пароль. Більш того, наявність відкритого тексту дозволяє зменшити кількість варіантів перебору до 240. Отже, атака успішно може бути виконана навіть на одному комп'ютері (швидкість перебору на комп'ютері з процесором Intel Pentium III 333 МГц складає приблизно 600 000 паролів в секунду).

У версії 2.0, вочевидь, була проведена серйозна робота над помилками. Злам нового алгоритму шифрування перебором вимагав вже приблизно 21023 операцій, що більше, ніж це можна було б здійснити на сучасній техніці. Про ефективні атаки, що використовують відкритий текст, офіційно нічого не відомо. Швидкість перебору паролів знизилась приблизно до 2000 штук за секунду (в 300 разів).

Але розробники RAR продовжили роботу. У версії RAR 3.0 (травень 2002 року) для шифрування став використовуватись алгоритм AES з ключем довжиною 128 бітів. Таке рішення було викликано двома причинами. По-перше, безпечніше використовувати перевірений і добре себе зарекомендувавший алгоритм, ніж дещо саморобне, а у AES тут немає конкурентів.

По-друге, у AES швидкість шифрування вища, ніж у алгоритма, використовуваного в RAR 2.0.

Окрім заміни алгоритму шифрування, в RAR 3.0 використовується і інша процедура отримання ключа шифрування з пароля. Ця процедура вимагає обчислення кеш-функції SHA1 262144 рази, що дозволяє перебирати лише до 3-х паролів в секунду, тобто в 600 разів менше, ніж для RAR 2.0.

**Шифрування файлів у програмах Microsoft**

На протязі багатьох років програми, що входять у Microsoft Office, дозволяють шифрувати документи за паролем, який вводить користувач. Але не завжди файли виявляються захищеними надійно.

***Microsoft Word і Mirosoft Excel.*** Шифрування файлів було реалізоване вже в Microsoft Word 2.0. З пароля за допомогою простого циклічного алгоритму отримувалась 16-байтова гама, яка накладалась на вміст документа. Але обчислення гами баз пароля не складало труднощів, оскільки гама накладалась на службові області, які мали фіксоване значення у всіх документах.

У Word 6.0/95 та Excel 5.0/95 алгоритм шифрування не змінився, а змінився лише формат файлів – він став базуватися на OLE Structered Storage. Для відновлення пароля також треба було знайти 16-байтову гаму, використану для шифрування. А знайти цю гаму можна було, базуючись на статистичному аналізі. У будь-якому тексті найчастіше зустрічається символ“ “ (пробіл). Таким чином, досить визначити код найчастіше використовуваного у тексті символа в кожній з 16 позицій, що відповідають різним байтам гами. Виконавши операцію XOR кожного знайденого значення з кодом пробілу (0x20), отримуємо байт гами.

В програмах Word 97/2000 та Excel 97/2000 дані шифруються за допомогою алгоритму RC4 з ключем довжиною 40 байтів. Таке шифрування вже не дозволяє миттєво знайти пароль. Можливості обчислювальної техніки за останні роки виросли на стільки сильно, що єдиний можливий ключ шифрування документів Word (з 240 можливих) може бути

знайдений максимум за чотири доби на комп'ютері з двома процесорами AMD Athlon 2600+.

Починаючи з Office XP, нарешті з'явилась підтримка шифрування документів ключами довжиною більше 40 бітів. Але більшість користувачів до цих пір використовують 40-бітове шифрування, оскільки воно дозволяє відкривати захищені документи у попередніх версіях офісних програм. Та й зміна налаштувань шифрування вимагає додаткових дій з боку користувача (відкриття діалогу налаштувань і вибору потрібного криптопровайдера), тоді як за замовчуванням використовуються 40-бітові ключі.

***Microsoft Access.*** Бази даних Microsoft Access можуть мати два типи паролів: паролі на відкриття і паролі для розмежування прав доступу на рівні користувачів.

Пароль на відкриття, як правило, ніколи не був серйозним захистом, оскільки, починаючи з Access версії 2.0, він зберігався у заголовку бази даних. Правда, сам заголовок був зашифрований алгоритмом RC4, але це не дуже посилювало стійкість, оскільки в рамках однієї версії формату завжди використовувався один і той самий 32-бітовий ключ шифрування, прошитий у динамічно завантажувальну бібліотеку, що відповідає за роботу з файлом бази даних. А враховуючи те, що RC4 – синхронний потоковий шифр, достатньо було один раз знайти гаму, що породжується RC4 з відомим ключем, і після цього пароль можна було визначити, виконавши додавання за модулем 2 гами і потрібних байтів заголовку. Починаючи з Access 2000, звичайне накладання гами вже не дозволяє відразу ж визначити пароль, оскільки необхідно виконати ще декілька додаткових нескладних дій. Але пароль все одно зберігається у заголовку, а отже, може бути звідти прочитаний.

Слід зауважити, що установлення пароля на відкриття бази даних не приводить до шифрування її вмісту. Однак, Access підтримує таку операцію, як шифрування бази даних, але сам пароль у цьому шифруванні ніяк не приймає участі, а ключ шифрування зберігається у заголовку бази.

Інший тип паролів, підтримуваних Microsoft Access, використовується не для забезпечення секретності, а для розмежування доступу. Але виявилось, що при проектуванні було допущено декілька помилок, пов'язаних з цими паролями. Правильнішим було б зберігати не самі паролі, а їх кеш-значення. Але через незрозумілі причини в системній базі даних, що містить імена, паролі та інші атрибути всіх користувачів, можна знайти самі паролі, зашифровані потрійним використанням DES з двома ключами в режимі EDE (Encrypt-Decrypt-Encrypt), коли перший ключ застосовується двічі, на першому і третьому кроці. Ключі зазвичай є константами і зберігаються у динамічно завантажуваній бібліотеці. Такий захист дозволяє швидко визначити пароль будь-якого користувача, хоча Microsoft й стверджує, що втрачені паролі користувачів не можуть бути відновлені.

В системній базі даних для кожного користувача зберігається унікальний ідентифікатор, що є функцією від імені користувача і деякого довільного рядка, що вводиться при створенні облікового запису. Саме цей ідентифікатор і ключем, за яким ідентифікуються користувачі в основній базі даних.

Так, наприклад, у кожної таблиці в основній базі даних є власник, який має максимальні права. Але в основній базі даних зберігається лише ідентифікатор користувача-власника, а ім'я і вся додаткова інформація для аутентифікації користувача зберігається у системній базі. І створюється враження, що якщо системна база даних буде втрачена, то доступ до вмісту основної бази даних отримати не вдасться. Але функція обчислення ідентифікатора користувача є обертаємою, що дозволяє визначити ім'я власника і рядок, введений при створенні облікового запису. Після цього лишається тільки створити нову системну базу даних і додати в неї користувача з відомими атрибутами, але взагалі без пароля.

***Encrypted File System.*** Починаючи з Windows 2000 операційні системи, які базуються на ядрі NT, підтримують Encrypted File System (EFS) – розширення файлової системи NTFS (New Technology File System), що дозволяє зберігати

файли користувачів у зашифрованому вигляді. При цьому шифрування виконується цілком прозоро і не вимагає від користувача додаткових зусиль, окрім одноразової вказівки на те, що файл має бути зашифрований.

Навіть якщо зловмисник зможе отримати фізичний доступ до файлової системи і прочитати захищений файл, йому не вистачатиме ключа шифрування для доступу до вмісту файла.

Симетричний ключ, яким зашифровується файл (File Encryption Key, FEK), сам зашифрований на відкритому ключі, що належить користувачу, який має право доступу до файла. Ключ зберігається разом із зашифрованим файлом, і для його розшифрування використовується секретний ключ користувача.

З кожним файлом може бути асоційовано декілька копій ключа, зашифрованих на відкритих ключах так званих агентів відновлення даних (Recovery Agents).

Процедура отримання усієї необхідної для розшифрування інформації включає в себе багато етапів. Але у Windows 2000 реалізація EFS є такою, що в більшості випадків усі зашифровані файли можуть бути добуті без знання пароля власника або агента відновлення.

### Контрольні питання до Лекції 5

1. Для чого існують програми-архіватори та пакувальники?
2. Для чого програми даного виду існували на початку свого створення?
3. У чому полягають переваги і недоліки стискання виконуваних файлів під Windows? У якому випадку слід стискати виконувані файли, а у якому – ні?
4. Охарактеризуйте доцільність стискання файлів динамічних бібліотек.
5. У чому полягає принцип роботи програм-архіваторів?
6. Наведіть основні відомі вам програми архівації файлів і коротко охарактеризуйте принцип їх роботи.
7. Що таке програми-зломщики архівів? Які види цих програм ви знаєте? Наведіть приклади.

8. Наведіть відомі програми-архіватори, охарактеризуйте їх і зробіть порівняльну характеристику.

9. Як здійснюється захист документів Microsoft Office?

10. Які особливості захисту файлів в операційних системах, що базуються на ядрі NT?

## Лекція 6

# НАЙПРОСТІШІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ В ОПЕРАЦІЙНИХ СИСТЕМАХ

**План лекції:**

1.Установки і налаштування системи захисту.

2.Блокування доступу до комп'ютера за допомогою екранної заставки Windows.

3.Використання пароля BIOS.

4.Програмні продукти для найпростішого захисту.

5.Відновлення інформації після збоїв.

### 1. Установки і налаштування системи захисту

Існує багато установок і налаштувань, що забезпечують деякі додаткові можливості системи захисту Windows. Усі вони безкоштовні та легко реалізуються. Не зважаючи на їх простоту, не слід ігнорувати такими методами захисту.

**Періодичне очищення меню Документи (Documents)**

Це меню містить перелік файлів (приблизно 15 найменувань), з якими ми нещодавно працювали. А це означає, що будь-який сторонній користувач може без проблем переглянути результати нашої роботи або наш особистий файл, навіть не використовуючи будь-якого спеціального пошуку.

Для очищення цього меню слід виконати такі дії: *Пуск – Настройка - Панель задач* и меню *«Пуск» - Настройка меню-Очистить.*

Після очистки цього меню зловмисникам прийдеться витратити більше часу для пошуку наших файлів.

**Очищення і установка Кошика (Recycle Bin)**

*Кошик* – улюблене місце для зловмисників, які хочуть добути якусь інформацію. Вона зберігає для використання будь-які файли, видалені за допомогою програми *Проводник* (Windows Explorer). Є три способи розв'язати цю проблему.

Перший спосіб полягає у тому, щоб очищати корзину кожен день або навіть частіше. Зробити це можна за допомогою

послідовності команд:

*Корзина – Очистить корзину.*

Встановити властивості Кошика так, щоб файли видалялися з нього безперервно, для чого виконати таку послідовність команд:

*Корзина – Свойства – Глобальные – Уничтожать файлы сразу*

потім … - *ОК.*

Видалити файли засобами DOS. Для цього виконати команди:

*Пуск – Програми – Сеанс MS-DOS*

Далі перейти в каталог, що містить файли, які ми хочемо видалити і командою DEL видалити їх. Але слід пам'ятати, що такі файли відновити вже не так просто, якщо це взагалі можливо.

Можна заставити зловмисника повірити, що ми не видаляємо файли з корзини, оскільки нам нічого приховувати. Для цього слід видаляти лише найбільш вразливі файли (видаляти з Корзини тільки ці файли або видаляти їх засобами DOS). Складатиметься враження, що ми нічого навмисно не видаляємо, тобто, нам нічого приховувати. Можливо, це зупинить зловмисника від подальших дій.

**Видалення або перейменування ярликів**

Багато хто з користувачів, отримавши доступ до комп'ютерної системи, відразу ж починають клацати по значках ярликів на *Робочому столі (Desktop)* Windows. Можна запобігти цьому, видаляючи і/або перейменовуючи ці значки. Тоді зловмисник може не звернути увагу на програму, ім'я якої в нього не викликає ніякого зацікавлення. Але при цьому самому слід пам'ятати нові назви ярликів.

Для видалення або перейменування значків на Робочому столі треба клацнути правою клавішею миші на вибраному ярлику і вибрати з контекстного меню, що з'явиться при цьому, пункт *Удалить* – для видалення його, пункт *Переименовать* – для перейменування.

**Видалення і перейменування пунктів меню Start**

Пункти меню *Пуск (Start)* легко можна видалити або змінити, і цими можливостями можна скористатися для запобігання деяких дій зловмисників щодо нашої інформації.

Для видалення слід виконати такі команди:

*Пуск – Панель задач и меню "Пуск" – Свойства – Настройка меню – Удалить,*

вибравши зі списку ту програму або папку, яку треба видалити. При цьому компоненти, що видалені, лишаються на вінчестері. Отже, запустити програми все ж таки можна. Крім того, можна запустити програму на виконання, або скориставшись командами *Пуск – Выполнить*, або за допомогою ярликів Робочого столу, або за допомогою програми *Проводник.*

Для перейменування пункту меню *Пуск* слід вибрати на вкладці *Панель задач* кнопку *Дополнительно* і перейменувати потрібний файл або папку за допомогою контекстного меню.

**Приховування Панелі задач**

У *Панелі задач* є один простий параметр – *Автоматически убрать с экрана (Auto Hide),* який може дещо спантеличити недосвідченого зловмисника і не привернути увагу стороннього користувача. Цей параметр робить *Панель задач* практично невидимою (вона ніби виштовхується зі звичайної області перегляду, хоча і буде з'являтися, якщо курсор миші пройде над її областю розташування). Для установки цього параметра слід виконати такі дії:

*Пуск – Настройка – Параметры панели задач – Автоматически убирать с экрана - ОК.*

**Захист від зміни і видалення файлів і папок**

Якщо ми не хочемо, щоб дехто випадково (або навмисно) видалив або змінив наші файли, можна дещо застрахуватись від цього. Для цього використовують програму *Проводник,* де за допомогою контекстного меню можна встановити властивості файла так, що він стане недоступним для зміни.

Атрибут *Только чтение (Read-only)* дозволить запобігти спробу користувачів відредагувати і зберегти файл за допомогою додатків Windows або DOS та видалити за

допомогою DOS. Хоча за допомогою програми *Проводник* видалити його все ж таки можна. Аналогічно поступають і з папками.

Атрибут *Скрытый (Hidden)* приховує файл або папку для перегляду програмами Windows, DOS і командами DOS для роботи з каталогами.

Звичайно, всі наведені вище міри захисту є надзвичайно ненадійними, і більш-менш досвідчений користувач з легкістю їх подолає. Це лише спроба утруднити роботу зловмисника. для більш надійного захисту слід використовувати і більш надійні інструменти.

**Резервне копіювання системи та шифрування дисків**

Для того, щоб захиститись від того, що сторонній користувач навмисно або випадково не знищить нашу інформацію, в систему попаде вірус або станеться повний збій системи, необхідно гарантувати збереження наших даних, тобто час від часу виконувати резервне копіювання системи. При установці програмного забезпечення рекомендується створити аварійний або системний диск. На цьому диску повинна знаходитись інформація, необхідна операційній системі для запуску комп'ютера з зовнішнього носія, а також деяка інформація про конфігурацію комп'ютера.

Звичайно, аварійний диск не допоможе відновити файли даних, створені за допомогою різних програмних додатків – текстових процесорів, редакторів електронних таблиць, графічних редакторів тощо. Для всіх цінних файлів даних слід використовувати резервне копіювання (архівацію) на інших носіях – на гнучких дисках, на CD-дисках, на ZIP-дисках.

Ще з часів DOS існують програми, що дозволяють створювати захищені диски, вміст яких стає доступним лише після того, як користувач введе правильний пароль. Отже, створення зашифрованих дисків – це ще один з найпростіших методів захисту власної інформації, і застосуванням програм, призначених для цього, також не треба нехтувати.

## 2. Блокування доступу до комп'ютера за допомогою екранної заставки Windows

Існує певний клас програм, здатних захистити комп'ютер від мережевих атак; налаштування ОС не дадуть використати "проколи" розробників (так звані "дірки" у програмному забезпеченні). Але всі ці прийоми не врятують від елементарної людської дурості і явного халатного відношення до захисту власного комп'ютера, що в більшості випадків приводить хакера до потрібного йому результату, тобто до успішного зламу. Але сподіватись на вдачу (тобто на відсутність будь-якого інтересу до ПК з боку зловмисників) щонайменше необережно, тому варто знати, як встановити найпростіше блокування у вигляді програм-заставок з паролем (і подібних їм), щоб грамотно організувати захист свого комп'ютера.

Для того щоб захиститись від зламу, потрібно знати, яким чином буде діяти зломщик, спробувати зрозуміти логіку його вчинків і, спираючись на отримані знання, грамотно організувати оборону. Отже, основною функцією програми-заставки є так зване "запирання" монітора, тобто блокування доступу до комп'ютера.

В операційній системі Linux опція "запирання" екрана передбачена за замовчуванням. Тобто перед тим, як користувач відійде від комп'ютера, він може активізувати заставку, що допускає відображення діалогового вікна із пропозицією ввести пароль користувача для продовження роботи (якщо стороння людина спробує забрати заставку рухом миші або натисканням будь-якої клавіші).

В Windows ця функція активізується установкою певної опції. Для активізації заставки з паролем необхідно відкрити "Панель управління" (команди *Пуск – Настройка - Панель управления*), у якій подвійним натисканням клавіші миші вибрати пункт *Екран.* У результаті відобразиться діалогове вікно властивостей екрана. Інший спосіб активізації цього вікна – натисканням правої кнопки миші по вільній поверхні Робочого столу викликати контекстне меню і вибрати пункт *Свойства.* На

закладці *Заставка* варто вибрати бажану заставку, вказати пароль за допомогою кнопки *Изменить* й установити прапорець *Пароль*. Далі потрібно ввести пароль. Для розблокування комп'ютера, що захищається за допомогою заставки з паролем, необхідно перезавантажити комп'ютер. Після цього заставка пропадає, а для обходження вхідного пароля в Windows 95/98 досить натиснути кнопку *Отменить*. Тому для захисту в Windows 95/98 доведеться використати спеціальні програми.

У системах Windows NT/2000/XP разом із програмою-заставкою варто використати вхідний пароль, щоб навіть після перезавантаження зловмисник не зміг одержати доступ до потрібної інформації.

Перезавантаження – не єдиний метод боротьби із заставкою. На жаль, він не позбавлений недоліків. Один з них полягає в тому, що програма запускається тільки через певний інтервал часу, коли з комп'ютером не здійснюють ніяких дій. Цього інтервалу цілком може вистачити зламщику, щоб отримати доступ до комп'ютера, поки заставка не спрацювала. Вирішується ця проблема так: у закладці *Заставка* натискаємо кнопку *Просмотр,* після чого заставка функціонує в нормальному режимі.

Наступний недолік більш істотний. Програма-заставка не відключає Autorun. Зловмисник може вставити в CD-ROM диск зі шкідливою програмою, що запуститься за допомогою Autorun. Для рішення даної проблеми необхідно відключити *автозапуск.* Для цього в Панелі керування подвійним натисканням клавіші мишки розкрити розділ *Система,* де на закладці *Оборудование* у властивостях параметра *Устройство чтения компакт-диска* слід зняти прапорець *Автоматическое распознавание диска.*

### 3. Використання пароля BIOS

Якщо захист за допомогою програм-заставок здається малоефективним через умови, у яких доводиться працювати, має сенс використати більш серйозні засоби безпеки. Розглянемо захист комп'ютера за допомогою BIOS і методи фізичного блокування доступу.

## Захист за допомогою BIOS

На кожному ПК є вбудована в нього система BIOS (Basic Input Output System – базова система введення-виведення), що являє собою декілька низькорівневих процедур, які тестують комп'ютер після включення його живлення і запускають операційну систему. Більшість BIOS підтримують можливість задання так званого пароля включення. Якщо пароль заданий, то комп'ютер виконає будь-яку операцію тільки після введення правильного пароля.

Для того, щоб задати пароль BIOS, необхідно при завантаженні комп'ютера, коли в правому нижньому кутку з'явиться напис Press DEL to enter Setup, натиснути клавішу Delete (або іншу зазначену), після чого додержуватися підказок, що з'являються при виділенні будь-якого пункту меню. Не будемо давати рекомендації з настроювань BIOS, оскільки їх випуском займаються багато виробників: IBM, AWARD, AMI й інші.

Розглянемо механізм задання пароля BIOS для AWARD BIOS і AMI BIOS. При цьому необхідно мати на увазі, що при роботі з BIOS необхідно бути досить обережним, оскільки неправильне настроювання може негативно позначитися на функціонуванні комп'ютера або навіть привести до виходу з ладу деяких його вузлів.

Існує два типи паролів, які можна задавати в BIOS, – пароль на завантаження комп'ютера й на завантаження меню Setup BIOS.

## Встановлення пароля AWARD BIOS

Для початку розглянемо, як задається пароль в BIOS виробництва Award. При завантаженні комп'ютера необхідно натиснути клавішу Delete, після чого можна вибрати один із двох варіантів задання пароля:

Set User Password (встановити пароль користувача). Він може задаватися або на вхід у меню Setup BIOS, або на вхід у Setup і завантаження комп'ютера. Для того, щоб вибрати будь-який з перерахованих вище методів блокування, необхідно в

меню Advanced BIOS Features вибрати рядок Password Check. Потім після натискання клавіші Enter у вікні, що з'явилося, варто виділити або Setup (тобто пароль тільки на Setup BIOS), або System (тобто пароль на завантаження системи й Setup BIOS);

Set Supervisor Password (встановити пароль суперпвізора). Має пріоритет перед паролем користувача. Всі інші функції й настроювання такі самі, як в User Password.

**Встановлення пароля AMI BIOS**

Виклик вікна настроювань AMI BIOS здійснюється натисканням клавіші F2 у процесі тестування обладнання після включення комп'ютера. Тут для задання пароля необхідно вибрати вкладку Security, у якій доступні Set Supervisor Password і Set User Password. Характерною рисою AMI BIOS є те, що користувач, який знає Supervisor Password, може обмежувати можливості користувача, що знає тільки User Password. Для того, щоб задати пароль, необхідно підсвітити бажаний тип пароля. Після натискання клавіші Enter відобразиться вікно із двома полями: у першому полі необхідно ввести пароль, а в другому – продублювати його.

**Обхід пароля BIOS**

Незважаючи на те, що BIOS є досить потужним засобом захисту, існують різні способи обходу встановленого пароля. Користувач може потрапити в досить скрутну ситуацію, якщо забуде встановлений пароль на завантаження системи. Це може призвести до необхідності придбання нової материнської плати. У подібних випадках можна скористатися "чорним ходом" BIOS, що також застосовується хакерами. Принцип його дії такий. Встановлений пароль зберігається в пам'яті CMOS (Complimentary Metal-Oxide-Semiconductor – комплементарний метало-оксидний напівпровідник), що, у свою чергу, повинна постійно підтримуватися батарейкою, встановленою на материнській платі. Отже, якщо батарейку витягти на якийсь час, можна домогтися очищення BIOS, тобто втрати встановлених параметрів і паролів.

Але справа в тому, що витягти батарейку не завжди

можливо без використання допоміжного інструмента, тому має сенс вдатися до способу, описаного в деяких інструкціях до материнської плати. На більшості материнських плат установлені виводи для очищення пам'яті CMOS. Як правило, ці виводи перебувають поруч із батарейкою. Якщо користувач у силу різних причин не може їх знайти, варто звернутися до інструкції на материнську плату, що повинна входити в комплект документів, одержаних користувачем при покупці комп'ютера. Інструкцію до материнської плати також можна скачати з Internet, із сайту фірми-виробника.

Перед початком маніпуляцій по очищенню пам'яті CMOS варто відключити живлення комп'ютера, тобто всі дії при очищенні пам'яті CMOS потрібно робити, тільки коли комп'ютер виключений, інакше це спричинить вихід материнської плати з ладу. Після відключення живлення за допомогою документації на материнську плату треба знайти виводи для очищення пам'яті CMOS і замкнути їх. Далі включаємо комп'ютер, заново виставляємо настройки BIOS, запам'ятовуємо їх і перезавантажуємось.

У випадку, коли документація відсутня, на виключеному комп'ютері варто спробувати по черзі замкнути всі виводи на платі, перевіривши потім, чи був знятий пароль.

Якщо всі перераховані вище маніпуляції не привели до потрібного результату, варто спробувати застосувати так званий інженерний пароль. Але при цьому необхідно пам'ятати про те, що він працює тільки на досить старих версіях BIOS, а також про те, що він може варіюватися від версії до версії. Нижче наведені паролі для BIOS виробництва фірм AWARD і AMI.

**Використання утиліти debug**

На практиці часто буває, що користувач, установлюючи пароль на BIOS Setup, робить його занадто складним і згодом забуває. При цьому витягти батарейку він з якоїсь причини не може, наприклад, через високу ймовірність ушкодити гарантійні пломби. У цьому випадку можна скористатися стандартною утилітою Windows debug.exe, яку необхідно запускати з консольного режиму.

Для запуску програми debug.exe необхідно в меню *Пуск* вибрати пункт *Виконати,* після чого у вікні, що з'явилося, ввести ім'я програми debug.exe. Потім у вікні, що з'явилося, необхідно ввести наступну послідовність команд, виконуючи переведення рядка клавішею Enter:

-o 70 33
-o 71 33
-q

В результаті пароль на Setup буде знятий. Необхідно відзначити, що з виконанням цієї команди знімаються деякі настроювання Windows, тому після перезавантаження для подальшої коректної роботи системи може знадобитися інсталяційний диск Windows.

**4. Програмні продукти для найпростішого захисту**

У наш час, коли недосвідчений користувач або вірус, що потихеньку пробрався на комп'ютер, можуть одним махом знищити всю систему, необхідно обов'язково мати під рукою надійні програми для створення дискового іміджу і його розпаковування, для блокування комп'ютера та приховування інформації за певними критеріями. Більшість таких програм є дорогими західними розробками, але існує багато програм для вільного розповсюдження або умовно-безкоштовних. Розглянемо деякі з них.

***Stacker.*** Це одна з перших програм, що дозволяє захищати диски. Точніше, вона призначена для створення стиснутих дисків, що підтримують пакування і розпакування інформації "на льоту", а одна з опцій дозволяла захищати збережену на диску інформацію паролем.

Але у програмі використана не зовсім надійна кеш-функція, внаслідок чого складність підбору пароля для розшифрування диска дорівнює 28, тобто пароль підбирається досить швидко.

***Diskreet.*** Ця програма входить до складу Norton Utilites і призначена для створення зашифрованих файлів і дисків. Програма підтримує два методи шифрування, один з яких

базується на алгоритмі DES і працює повільно, а другий – саморобний і більш швидкий алгоритм, хоча і є примітивним і нестійким до атак на основі відкритого тексту.

***BootLock.*** Програма дозволяє шифрувати завантажувальний диск, а отже, робить всю обчислювальну систему недоступною для зловмисника. При цьому програма шифрує не всі дані диска, а лише системні області: завантажувальний запис (Boot) і кореневу директорію (Root). А таблиці розміщення файлів (Fat-таблиця) і область даних, в якій знаходяться самі файли, виявляються незашифрованими.

Шифрування виконується шляхом накладання 512-байтової гами на кожний сектор, що підлягає шифруванню, причому для всіх секторів одного диску використовується одна й та сама гама. Через це наявність одного відкритого сектора дає можливість розшифрувати і інші.

***Screen Look.*** Це програма від компанії iJEN Software, яка добре зарекомендувала себе, відноситься до програм-заставок.. Вона підтримує можливості "запирання" комп'ютера під час короткотермінової відсутності користувача. Крім того, вона може бути використана в якості "вхідного контролю" в операційну систему.

***Black Magic.*** Програма Black Magic безкоштовна. Особливістю цієї програми є те, що вона не є програмою-заставкою у чистому вигляді, оскільки під час блокування не використовує екранну заставку. Слід зазначити, що подібні програми здатні забезпечити безпеку лише на досить короткий термін при відсутності користувача, і тому покладати на них великі надії не можна. Але для користувачів Windows 9X дані програми є непоганою альтернативою "рідним" утилітам аналогічного призначення. З іншої сторони, для більш серйозного блокування системи застосовують захист за допомогою BIOS і деякі інші прийоми.

***Backup 2001 v.2.04.*** Програма для резервного копіювання і відновлення. Для цього вибрано ZIP-формат як найпоширеніший. Резервне копіювання здійснюється на дискети, жорсткі диски, по локальній мережі, ZIP-диски, JAZZ-

диски, Sparq, CD-R і ін. Працює під Windows 95(OSR2) /98/NT4/2000.

***Desktop Disguise v.1.0.*** Програма дозволяє приховати робочий стіл разом зі всіма вікнами методом перекриття заданим зображенням.

***PC Security v.4.11.*** Утиліта, призначена для приватного доступу до даних шляхом блокування файлів, каталогів, системи.

***Bluescreen Screen Saver.*** Цікавий зберігач екрану, який емулює на екрані комп'ютера системну помилку або, іншими словами, видає синій екран. За синім екраном слідує спроба перезавантаження комп'ютера з подальшою "системною помилкою". І так по кругу. Виглядає це досить реалістично, і необізнаній людині може дійсно здатися, що наш комп'ютер "висить".

***HDD Password Protection 2.0.*** Якщо ми не хочемо, щоб у нашу відсутність хтось працював на нашому комп'ютері, слід обмежити доступ до нього. Можна, наприклад, поставити пароль на вхід у Windows – це найпростіший, але і найефективніший спосіб захисту. Ще надійніше обмежити доступ ще в BIOS, але, як показує практика, і цей варіант теж небездоганний і достатньо ризикований.

Програма HDD Password пропонує нам золоту середину. Код доступу при її використанні потрібно вводити при старті операційної системи, а не перед входом в її графічну оболонку. Правда, в цьому випадку все одно можна дістати доступ до даних на жорсткому диску шляхом завантаження з дискети або CD-ROM. Вищого рівня захисту розробники обіцяють добитися в комерційній версії HDD Password Professional. Можливо, в цьому варіанті з'явиться у програми і нормальний графічний інтерфейс. Поки ж запускається вона тільки в DOS або в досівському вікні Windows. Втім, ніяких складнощів в її управлінні немає. Переміщаючись за допомогою клавіатури по рядках меню, можна задати пароль, зняти його, повернути колишній варіант завантажувального MBR-сектора, який, до речі, автоматично зберігається.

Важливо згадати, що HDD Password при установці стирає дані boot-менеджерів і дозволяє завантажувати тільки Windows.

***WinGuard Pro***. Програма WinGuard Pro допоможе нам забезпечити свій комп'ютер від несанкціонованого вторгнення або ж від дій невмілого користувача. Програма може заблокувати за допомогою пароля запуск певних файлів, меню *Пуск, Робочий стіл* і т.д. За допомогою її можна також заборонити установку і видалення програм, закрити доступ в Інтернет. По суті, можна заблокувати практично будь-який елемент операційної системи.

WinGuard Pro розповсюджується не безкоштовно, але при першому запуску у нас буде можливість вибрати тип роботи: використовувати програму безкоштовно, але з істотними обмеженнями функціональності, або ж отримати доступ до всіх функцій утиліти на термін 30 днів.

***Acronis True Image 6.0.*** Acronis True Image – російськомовна і недорога програма, здатна створювати образи дисків прямо з операційної системи Windows.

Що стосується ціни, то ця програма коштує на декілька порядків менше, ніж її західні аналоги (всього близько $10), а по можливостях не поступається багатьом з них. Перше – це простий і продуманий інтерфейс, виконаний у вигляді "майстра". Все, що потрібно користувачу – це переходити від вкладки до вкладки, вибираючи по черзі потрібні параметри. Сьогодні це як ніколи актуально, оскільки багато користувачів починають своє комп'ютерне "життя" відразу з Windows XP, і спочатку їм складно розібратися в досить розгалуженому інтерфейсі, наприклад, такої програми, як DriveImage.

Основні характеристики програми такі. Перше – це ступінь компресії, з якою вона стискає дані. Так, логічний диск, заповнений на 3.6 Гбайти, програма стиснула в архів розміром 1,5 Гбайт, що дуже навіть непогане. При цьому можна створити імідж не на жорсткому диску, а на знімних носіях таких як ZIP-накопичувач або компакт-диск. При цьому не варто турбуватися, що розмір іміджу перевищить об'єм носія, оскільки програма уміє розбивати архіви на декілька частин.

У True Image передбачена маса корисних дрібниць, які роблять роботу з нею зручнішою. Так, наприклад, є можливість додати довгий коментар до створюваного образу.

Найголовніше зручність програми полягає у тому, що вона може створювати і відновлювати образи розділів без перезавантаження в DOS. Це дуже істотний плюс, оскільки швидкість роботи при цьому збільшується майже в два рази. Такою можливістю не володіє жодна аналогічна програма, навіть такі як PowerQuest Drive Image і Norton Ghost.

***MAPBackup 1.2.*** Дуже зручна програма для створення резервних копій. Не дивлячись на те, що всі подібні утиліти схожі одна на одну, MAPBackup все ж таки різниться в цьому. А відрізняє програму від аналогів цікавий підхід до архівації даних. MAPBackup може створити копію робочих файлів певної програми відразу після того, як користувач її закрив. Це дуже корисно, наприклад, для створення резервної копії пошти. При настроюванні програми потрібно вказати, який процес повинен відстежуватися (наприклад, thebat.exe), який архіватор повинен бути використаний і які директорії повинні бути збережені. Після того, як процес буде завершений користувачем, MAPBackup створить архівну копію вказаних файлів або директорій.

***SmartBackup.*** Збереження важливої інформації і файлів останнім часом стало задоволене актуальною темою. Втрата документів, настройок і лігв інтернет-пейджерів, поштових повідомлень може стати причиною серйозних неприємностей. А втратити все це багатство дуже легко. Системний збій, вірус або не вміюча працювати з комп'ютером людина можуть дуже скоро поховати результати багатомісячної праці. Врятувати від цього кошмару можуть утиліти для резервного копіювання, і SmartBackup - одна з них. Працює програма дуже просто. У призначений час вона копіює вказані файли або теки в окрему директорію. Копіювати можна не всі файли директорії, а тільки ті, які піддалися модифікації. До всього, SmartBackup володіє такою корисною можливістю, як стиснення резервної копії в архів.

## 5. Відновлення інформації після збоїв

Важлива інформація може бути втрачена як через необмірковані дії користувачів, так і внаслідок збоїв у роботі техніки або внаслідок навмисного пошкодження даних зловмисником. Але існують програмні засоби, які дозволяють відновити інформацію після таких збоїв. Розглянемо деякі з цих програм.

***Програма EasyRecovery Pro.*** Популярна програма для відновлення видалених або пошкоджених файлів. Усі налаштування програми EasyRecovery Pro досить прості і доступні користувачу, хоча роботу може утруднити повністю англомовний інтерфейс.

У своєму складі програма містить утиліти, за допомогою яких можна виконати діагностику жорстких дисків на предмет наявності помилок:

- DriveTests – сканування приводів усіх накопичувачі на предмет потенційних неполадок;
- SMARTTests – моніторинг жорсткого диска для виявлення можливих пошкоджень;
- SizeManeger – надання інформації про використання дискового простору;
- JumperViewer – перевірка правильного підключення пристроїв;
- PartitionTests – аналіз існуючої структури файлової системи;
- DataAdvisor – створення завантажувальної дискети для діагностики системи.

Основу даної програми складають утиліти для безпосереднього відновлення даних:

- AdvancedRecovery – відновлення даних з використанням специфічних налаштувань;
- DeletedRecovery – знаходження і відновлення пошкоджених файлів;
- FormatRecovery – відновлення інформації з відформатованого або видаленого тома;

- RawRecovery – відновлення інформації з директорії;

- ResumeRecovery – перегляд LOG-файлів, створених під час роботи по відновленню даних;

- EmergencyDiskette – створення аварійної завантажувальної дискети.

Утиліти для відновлення конкретних файлів:

- AccessRepair – відновлення бази даних Microsoft Access;

- ExcelRepair – відновлення таблиці Microsoft Excel;

- PowerPointRepair – відновлення презентації Microsoft PowerPoint;

- WordRepair – відновлення документу Microsoft Word;

- ZipRepair – відновлення пошкодженого WinZip-архіва;

- EmailRepair – засоби для відновлення e-mail листів Outlook Repair;

- Software Updates – засоби для оновлення програми;

- Crisis Center – тематичні каталоги з посиланнями, що дозволяють допомогти користувачу у кризисних ситуаціях.

***Програма FinalData.*** Дана програма дозволяє відновлювати інформацію, втрачену в результаті дії вірусів, форматування дисків або випадкового знищення файла.

Діалог програми дозволяє вибрати диск, на якому знищено або пошкоджено інформацію і діапазон кластерів, в якому здійснювати пошук. В результаті сканування програма показує весь вміст диска на даний момент (як існуючі так і видалені файли). Крім того, можна окремо вивести розділ видалених файлів і розділ втрачених файлів. Потім над файлом здійснюються певні операції по його відновленню.

Програма має зручний інтерфейс і розгалужену систему меню, зручну і зрозумілу у роботі.

***Програма GetDataBack***. Це також популярна програма для відновлення пошкоджених та видалених файлів. Дії в ній виконуються за допомогою покрокового Майстра, який на першому кроці пропонує ряд налаштувань для встановлення параметрів:

- вибір каталогу, в який будуть копіюватися відновлені файли;

- вибір каталогу для тимчасових файлів;

- вибір шляху до програми обробки диска DiskExplorer, яка встановлюється у разі необхідності;

- можливість відновлення знищених файлів (навіть не прив'язаних до якогось певного каталогу;

- можливість використовувати або не використовувати Fat-таблицю (якщо знайдена на момент відновлення Fat не відповідає файловій системі).

На другому кроці здійснюється сканування вказаного диску, враховуючи вибрані на першому кроці параметри. При цьому за бажанням користувача пошук і сканування може здійснюватися як по всьому вибраному диску, так і на певній його частині (вказується початковий і кінцевий сектори), в одній чи іншій файловій системі (якщо ця опція не вказана, буде здійснюватись більш детальний пошук, що збільшить час сканування).

Далі, керуючись пропозиціями Майстра, можна відновлювати певні файли і каталоги. Слід зауважити, що не слід копіювати відновлені дані на той диск, з якого виконується відновлення.

**Контрольні питання до Лекції 6:**

1.Перелічіть загальні методи блокування доступу до комп'ютера.

2.Що таке екранна заставка, як її поставити або змінити на комп'ютері?

3.Як блокувати роботу за допомогою екранної заставки?

4.Які методи використовують хакери для обходу блокування за допомогою екранної заставки? Як цьому запобігти?

5.Які інші програми ви знаєте для блокування роботи комп'ютера? Коротко охарактеризуйте їх. Наведіть переваги та недоліки їх застосування.

6.У чому полягає блокування роботи комп'ютера за допомогою BIOS?

7.Як встановити пароль BIOS на різні версії?

8.Які методи використовують хакери для обходу парольного захисту BIOS?

9.Що можна зробити у разі втрати паролю BIOS?

10. Що таке CMOS? Яке функціональне призначення цього об'єкту комп'ютерної системи?

11. Як можна стерти пам'ять CMOS?

12. Що таке інженерний пароль і для чого його використовують?

13. Для чого існує утиліта debug?

## Практична робота 1 (2 год.)

***Тема практичної роботи:*** **«Приховування файлів».**

***Мета роботи****:* Провести огляд можливих способів та методів приховування файлів та практично застосувати метод програми Hide Folder.

### Теоретичні відомості:

У наш час інформація перетворюється на найдорожчий ресурс. Оперативне отримання інформації дає перевагу над конкурентами, які її не мають, конфіденційна інформація про вас та вашу діяльність, що потрапила до зловмисників може серйозно вам нашкодити – про унікальну технологію можуть дізнатися конкуренти і просто скопіювати її, уникнувши витрат на дослідження; або ваші плани стануть відомі і нечесні суперники приймуть упереджувальні заходи. Для полегшення пошуку, обміну, обробки інформації як фізичні особи, так і фірми використовують персональні комп'ютери приєднані до мережі Інтернет. Комп'ютеризації та Інтернет торкнулись майже всі аспекти діяльності людини: від укладення договорів до особистих відносин, від придбання товарів через Інтернет-магазин до навчання. Комп'ютери значно спростили діяльність людини, виконуючи за неї машинну роботу.

Неважко уявити собі, що у вас є інформація, яка призначена лише для вас і вам не хочеться, щоб інший користувач вашого комп'ютера мав до неї доступ. В мережі існує багато програм для утаєння окремих тек і файлів від інших користувачів комп'ютера. Ці програми пропонують простий метод утаєння тек - за допомогою фільтрації запитів до файлової системи. Але це не означає 100% захист даних.

Одним з найбільших бажань користувача – бажання заховати свої особисті документи, програми і інші дані подалі від чужих очей. Найпоширеніший спосіб для цього:

- використовувати програмки для утаєння файлів і тек. Такі зараз досить багато: Folder Lock, Hide Folders XP.

- використовувати стандартні засоби в Windows: помітити

папку, файл як прихований, на NTFS томах - заборонити доступ до теки/файла шляхом вказівки прав для доступу до об'єкту.

Зручність цих методів полягає в тому, що для утаєння інформації не потрібно робити нічого зайвого. Все довільно просто: досить одного натискання на кнопку миші, щоб приховати ту або іншу теку/файл. Але насправді приховані файли і теки можна побачити і проглянути за допомогою інших засобів.

Просте утаєння тек і обмеження доступу до файлів не рятує в наступних випадках:

- завантаження ПК в іншій операційній системі (якщо вона є) або завантаження іншою ОС з CD-ROM. Наприклад Linux Blin.
- у іншій ОС, яка підключиться до вашого HDD диску, всі файли на ньому будуть видні, оскільки жодних обмежень і захисту не буде.
- завантаження Windows в Safemode (безпечний режим).
- після такого завантаження всі теки будуть видні, та ті ж, які були приховані програмно. Адже, при завантаженні в безпечному режимі, Windows завантажує лише драйвера, необхідні для роботи системи, а всі додаткові пропускає (драйвер-фільтри), щоб заздалегідь відкинути потенційно можливі збої.

Зняти HDD диск і підключити його до іншого комп'ютера.

Якщо зняти жорсткий диск (HDD диск) і підключити його до іншого комп'ютера, то можна побачити і відкрити всі приховані теки і файли. Будуть видні навіть ті теки, доступ до яких заборонений (на NTFS томах) .

Є ще одна можливість проглянути файли, приховані різними програмками. Для цього необхідно володіти правами Адміністратора (Обліковий запис з Адміністраторськими повноваженнями). Тому що, якщо Ви є адміністратором системи, ви можете деінсталювати (видалити) програми для утаєння тек, або завантажитися в режимі захисту від збоїв, і тоді всі приховані об’єкти стануть видимими.

Якщо потрібний надійніший захист, слід звернутися до серйозних криптографічних пакетів (тотальне шифрування).

Як показує практика, найбільшу надійність зберігання інформації дає поєднання декількох способів захисту. Саме цей шлях був вибраний при створенні програми Hide Folder 3.0.

За допомогою Hide Folder 3.0 ви можете не лише зашифрувати вашу конфіденційну інформацію, але і приховати сам факт зберігання її на вашому комп'ютері або ноутбуку. Цей диск буде доступний лише по паролю.

Програма працює за наступними принципами.

Все досить просто. Ви вибираєте потрібні дані і для доступу до них призначаєте пароль. Якщо пароль знаєте лише Ви – то і доступ до інформації дістанете лише Ви! Всі дані зберігаються на окремому диску, який програма сама і створює. Цей диск є віртуальним і він може бути створений на різних носіях (HDD, cd-диск, dvd-диск, Usb-flash drive і ін.). Диск шифрується за допомогою швидкісних алгоритмів шифрування, тобто «на льоту», безпосередньо в процесі роботи. Такий диск називається віртуальним зашифрованим диском.

Після закінчення роботи з файлами, Ви відключаєте диск. Сам диск і вся інформація, яка зберігається на ньому, стають недоступними до тих пір, поки диск не буде знову включений. Диском є один файл, який за розміром дорівнює об'єму диска. Цей файл можна побачити, але він зашифрований, тому інформацію з нього витягувати не можна.

Якщо вам не хочеться запам'ятовувати пароль до диска, то ви можете надбудувати свій flash диск, як ключ до вашого Зашифрованого Диска.

**Хід роботи:**

1. Завантажити з мережі Internet програмний продукт Hide Folder.

2. Встановити програму на свій комп'ютер. За замовчуванням директорія C:\Program Files\Hide Folder XP.

3. Ознайомитись з інтерфейсом та перейти до налаштування програми. Процес налаштування зображений на рис. 3.

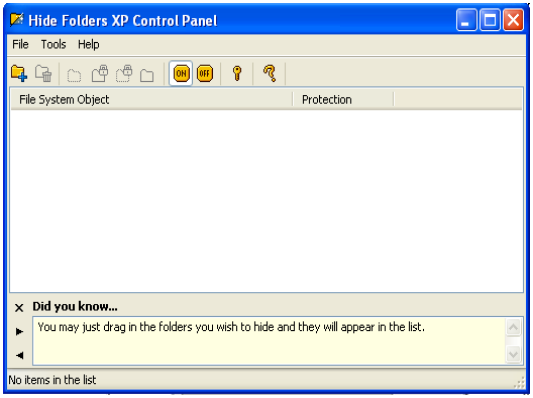



Рис.3. Вигляд вікна головного меню програми

4. Вибрати необхідну папку для приховування (рис. 4).

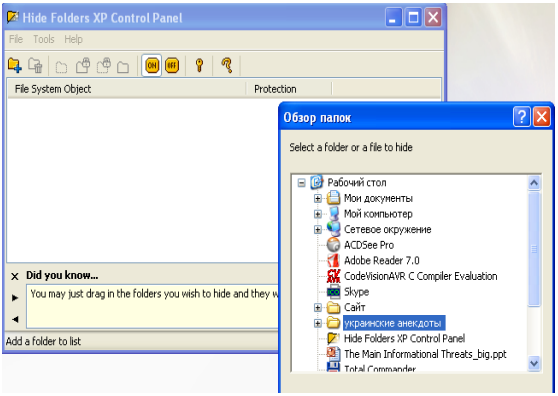



Рис. 4. Вигляд вікна вибору приховуваної папки

5. Приховати необхідну папку чи файл за допомогою ключа та за вибраним Вами певним методом шифрування (рис. 5)

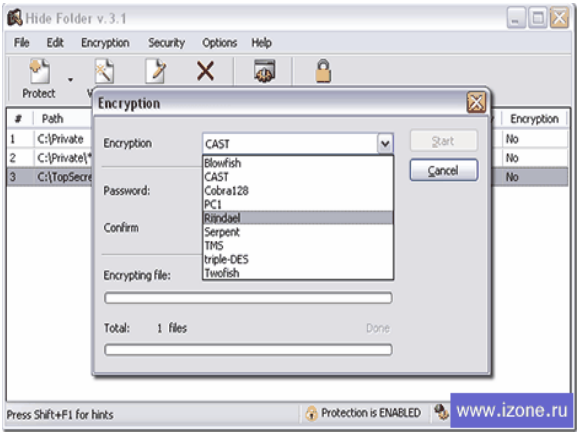



Рис. 5. Вигляд вікна вибору метода шифрування

6. Після вибору методу шифрування ввести пароль (рис. 6).

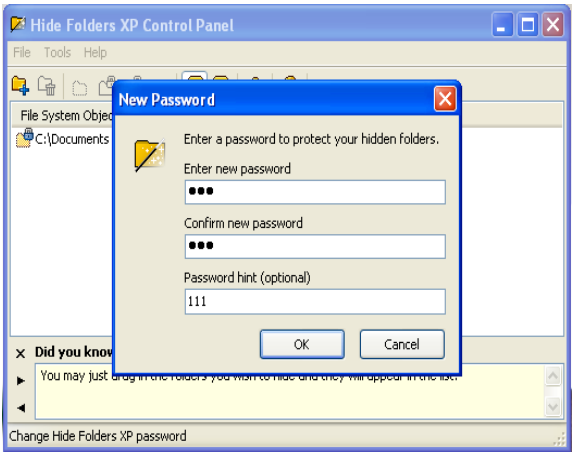



Рис. 6. Вигляд вікна вводу паролю

7. Після приховування дана папка відображається в

головному меню програми.(рис.7)

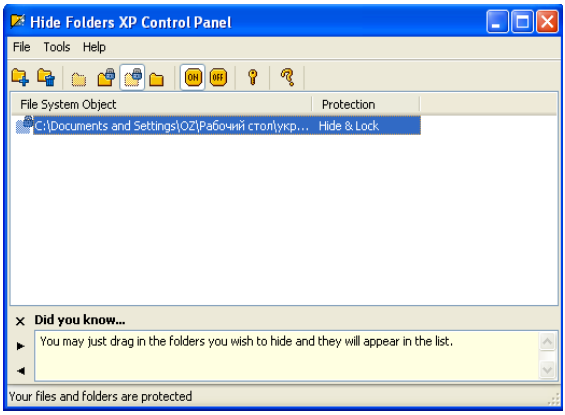



Рис. 7. Вигляд вікна прихованої папки

8. Для відображення прихованої папки вводимо пароль і натискаємо кнопку «OFF» (рис. 8)

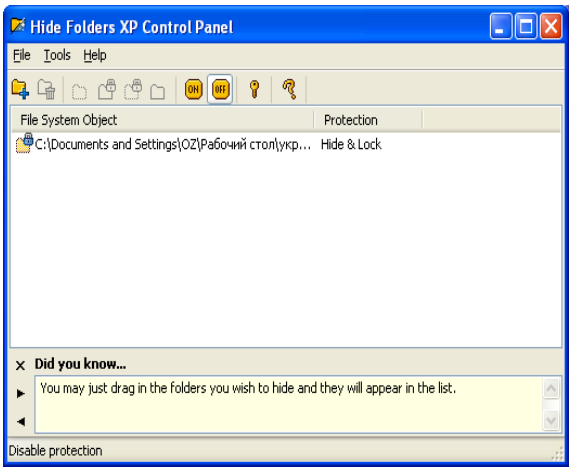



Рис. 8. Відображення прихованої папки

**Контрольні питання:**

1.Які Вам відомі найпоширеніші способи приховування фалів?

2.Назвіть найвідоміші програми для приховування файлів.

3.Принцип роботи програми Hide Folder.

## Практична робота 2 (4 год.)

***Тема практичної роботи*: «Основні ознаки присутності на комп'ютері шкідливих програм».**

***Мета роботи**:* навчитися виявляти шкідливе програмне забезпечення з використанням як явних, опосередкованих так і прихованих ознак.

### Теоретичні відомості:

Види проявів. Прояви шкідливих програм можна умовно розбити на три групи відповідно до того, наскільки легко їх виявити:

- ***явні*** – шкідлива програма самостійно проявляє помітну активність;
- ***непрямі*** – інші програми починають виводити повідомлення про помилки або поводитися нестандартно через присутність на комп'ютері вірусу;
- ***приховані*** – ні явних ні непрямих проявів шкідлива програма не має.

### Явні прояви

Характерні для троянських і рекламних програм. Це і зрозуміло, оскільки основною ознакою вірусів і черв'яків є здатність до зараження, для реалізації якої необхідний час. Якщо мережевий черв'як при проникненні на комп'ютер відразу ж себе виявить, користувач зможе відключити комп'ютер від мережі, перешкодивши подальшому розповсюдженню шкідливої програми.

Навпаки, троянські програми пишуться для виконання якоїсь конкретної шкідливої функції і скритність їм потрібна більшою мірою на етапі проникнення. Але все залежить від типу трояна.

З рекламними модулями просто: їх основна мета – залучення уваги до об'єкту реклами (веб-сайту, програми тощо), а привернути увагу – означає виявити свою присутність.

В даний час явні прояви, як правило, так або інакше пов'язані з мережею Інтернет. До основних явних ознак можна віднести:

**Зміна налаштувань браузера** – зміна стартової сторінки

браузера, зміна стандартної сторінки пошуку, несанкціоноване відкриття нових вікон – все це може бути наслідком присутності в системі шкідливої програми. Іноді до аналогічних ефектів може призводити виконання шкідливого скрипта на одному з відвіданих сайтів. У такому разі нові програми на комп'ютер не проникають, а налаштування браузера можна відновити і повністю вирішити проблему. Якщо ж після відновлення налаштувань вони знову міняються при запуску браузера або після перезавантаження комп'ютера, то причина змін – наявність на комп'ютері шкідливої програми. Подібна поведінка характерна для рекламних модулів, яка примусово відправляє користувачів на сайт, що рекламує яку-небудь продукцію. А також для троянських програм, які направляють користувача на сайти, що містять інші шкідливі програми.

Спливаючі повідомлення – після установки в системі, троянська або рекламна програма виводить на екран повідомлення про те, що на комп'ютері виявлені шкідливі або рекламні програми. Такі повідомлення зазвичай зроблені схожими на стандартні службові повідомлення MS Windows і забезпечені гіперпосиланнями або кнопками для переходу на веб-сайт, з якого нібито можна завантажувати програму для виявлення і видалення небажаних модулів. Не дивлячись на те, що прояви достатньо явні – повідомлення на екрані, через їх маскування під службові повідомлення, користувач не завжди здогадується, що це результат роботи шкідливих програм і в результаті потрапляє на ті ж рекламні або шкідливі сайти.

**Несанкціонований дозвон в Інтернет** – не так давно набули поширень особливі шкідливі програми – утиліти дозвону. Ці утиліти без санкції користувача і ігноруючи налаштування намагаються встановити з'єднання з Інтернет через дорогу телефонну лінію або дорогого провайдера. В результаті власник комп'ютера отримує рахунок на значну суму. Отже, ознакою зараження може бути несанкціоновані спроби комп'ютера з'єднатися з Інтернет.

**Непрямі прояви**

На відміну від явних проявів, непрямі прояви не завжди є

навмисними і нерідко викликані помилками, допущеними автором шкідливої програми.

**Блокування антивірусу** – зазвичай шкідлива програма проникає на захищений антивірусом комп'ютер або якщо антивірус був відключений, або якщо це порівняно нова шкідлива програма, для якої не було запису в антивірусній базі. Зрозуміло, що незабаром антивірус буде включений, або вірус буде внесений до антивірусної бази, і антивірус зможе його виявити і знешкоджувати. Щоб перешкодити цьому, багато шкідливих програм небезуспішно намагаються вивантажити антивірус з пам'яті або навіть видалити файли антивірусу з дисків комп'ютера. Тому раптове завершення роботи антивірусу цілком може бути ознакою зараження.

**Блокування антивірусних сайтів** – оскільки вивантаження або видалення антивірусу все ж таки достатньо помітні, деякі шкідливі програми нейтралізують тільки можливість оновлення антивірусних засобів. Якщо антивірусна база не оновлюватиметься, антивірус не зможе виявляти нові віруси і стане неефективним. При цьому шкідливі програми не блокують доступ в Інтернет цілком – це було б дуже помітно, а тільки доступ до сайтів і серверів оновлень найбільш відомих компаній – виробників антивірусів. В середньому, користувачі не часто заходять на сайти антивірусних компаній, а повідомлення антивірусу про неможливість оновитися можуть списувати на проблеми у провайдера або на самих серверах оновлення.

Збої в системі або в роботі інших програм – дуже часто причиною збоїв в роботі програм користувачі вважають присутність на комп'ютері вірусів. І хоча більшість подібних випадків на перевірку виявляються помилковою тривогою, віруси дійсно іноді можуть бути причиною збоїв. Наприклад, результатом присутності троянської програми Backdoor.NTHack може бути повідомлення про помилку, що виникає при завантаженні комп'ютера:

STOP 0x0000001e KMODE_EXCEPTION_NOT_HANDLED in win32k.sys

**Поштові повідомлення** – якщо комп'ютер заражений і розсилає інфіковані поштові повідомлення, вони можуть бути виявлені на одному з серверів в Інтернеті і антивірус на сервері може відправити повідомлення відправникові зараженого повідомлення. Отже, непрямою ознакою присутності вірусу може бути отримання поштового повідомлення про те, що з поштової адреси користувача комп'ютера був відправлений вірус. Проте, останнім часом багато вірусів підміняють адресу відправника і отримання описаного повідомлення не обов'язково означає, що комп'ютер заражений. Через те, що формальна адреса відправника, вказана в поштовому повідомленні, може не мати ніякого відношення до зараженого комп'ютера, антивірусні програми часто взагалі не посилають повідомлень відправникам заражених повідомлень.

**Приховані прояви**

У відсутність явних або непрямих проявів про присутність вірусу можна судити, наприклад, по незвичайній мережевій активності, коли жодне мережеве застосування не запущено, а значок мережевого з'єднання сигналізує про обмін даними. Іншими ознаками можуть служити незнайомі процеси в пам'яті або файли на диску.

Проте в даний час на комп'ютерах зазвичай встановлено так багато різних програм, що більшість файлів і процесів невідомі звичайному користувачеві. В той же час пошук прихованих проявів – це вже фактично пошук тих самих підозрілих файлів, які потрібно відправити на аналіз в антивірусну компанію.

Де шукати ознаки шкідливого програмного забезпечення. Як видно, ні непрямі, ні навіть явні прояви не можуть служити підставою для впевненості в тому, що комп'ютер заражений. Завжди існує ймовірність, що спостережуваний ефект не є результатом дій вірусу, а викликаний звичайними помилками у використовуваних програмах або ж шкідливими скриптами, які не залишили ніяких файлів на комп'ютері.

Для того, щоб підозри переросли в упевненість, потрібно провести додатковий пошук прихованих проявів шкідливих

програм, маючи кінцевою метою виявлення файлів шкідливої програми.

Приховані прояви включають:

- наявність в пам'яті підозрілих процесів;
- наявність на комп'ютері підозрілих файлів;
- наявність підозрілих джерел в системному реєстрі MS Windows;
- підозріла мережева активність.

Ключовою ознакою у всіх випадках є атрибут "підозрілий". Тобто, користувачеві невідоме призначення даного процесу, файла або ключа, і більш того, інформації про підозрілий об'єкт немає ні в документації до операційної системи, ні у відкритих джерелах мережі Інтернет.

Але перш ніж судити про підозрілість файлів і процесів, потрібно спочатку їх виділити із загального числа.

**Підозрілі процеси** – це фактично запущені виконувані файли. Частина процесів відноситься до операційної системи, частина до запущених програм. Знайти інформацію про невідомий процес можна в мережі Інтернет.

**Автозапуск** – особливою ознакою більшості черв'яків і багатьох троянських програм є зміна параметрів системи так, щоб файл шкідливої програми виконувався автоматично при кожному запуску комп'ютера. Тому наявність незнайомих файлів в списку файлів автозапуску також є приводом для пильного вивчення цих файлів. Залежно від настройок MS Windows і встановлених програм ключі автозапуску можуть містити різні рядки для запуску різних програм. Тому всі, на перший погляд, підозрілі файли потрібно перевіряти ще раз – вони можуть виявитися цілком звичайними програмами. У жодному випадку не слід змінювати налаштування системного реєстру навмання – це може привести до повної непрацездатності комп'ютера і необхідності встановлювати заново операційну систему. Вносити зміни до реєстру можна тільки будучи абсолютно впевненим у своїх діях і повністю усвідомлюючи характер і наслідки модифікацій.

**Мережева активність** – шкідливі програми можуть

виявлятися у вигляді мережевої активності. Черв'яки використовують мережу для розповсюдження, троянські програми – для завантаження додаткових компонентів і відсилання інформації зловмисникові. Деякі типи троянських програм спеціально призначені для забезпечення віддаленого управління зараженим комп'ютером. Для забезпечення доступу до комп'ютера по мережі з боку зловмисника вони відкривають певний порт. Як правило, схожі за призначенням програми використовують одні і ті ж порти для прийому з'єднань. Аналогічно поштовому серверу, шкідливі програми для прийому команд або даних від зловмисника використовують певний порт, постійно чекаючи сигналів на цей порт. У таких випадках прийнято говорити, що програма слухає порт.

Отже, за наслідками аналізу процесів, параметрів автозавантаження і з'єднань отримано список підозрілих з погляду користувача процесів (імен файлів). Для недосвідченого користувача невідомих, а значить підозрілих імен файлів може опинитися дуже багато, тому має сенс виділити найбільш підозрілі з них – ті, які були виявлені в двох і більш джерелах. Наприклад, файли, які присутні в списку процесів і в списку автозавантаження. Ще більш підозрілими є процеси, виявлені в автозавантаженні та прослуховуючи порти.

Для з'ясування природи підозрілих процесів найпростіше використовувати Інтернет. У глобальній мережі є сайти, що збирають інформацію про різні процеси. Даних по всіх процесах шкідливих програм на таких сайтах може і не бути, але в усякому разі на них є інформація про велику кількість безпечних процесів і таким чином можна буде виключити із списку ті процеси, які відносяться до операційної системи або відомих нешкідливих програм.

Після того, як список підозрілих процесів максимально звужений і в нім залишилися тільки ті, про які немає вичерпної і достовірної інформації, залишається останній крок, знайти ці файли на диску і відправити на дослідження.

**Методи захисту від шкідливих програм**

Відомо, що для захисту від шкідливих програм потрібно

використовувати антивіруси. Але в той же час нерідко можна почути про випадки проникнення вірусів на захищені антивірусом комп'ютери. У кожному конкретному випадку причини, з яких антивірус не впорався із своїм завданням можуть бути різні, наприклад:

- антивірус був відключений користувачем;
- антивірусні бази були дуже старі;
- були встановлені слабкі налаштування захисту;
- вірус використовував технологію зараження, проти якої у антивірусу не було засобів захисту;
- вірус потрапив на комп'ютер раніше, ніж був встановлений антивірус, і зміг знешкодити антивірусний засіб;
- це був новий вірус, для якого ще не були випущені антивірусні бази.

Але в цілому можна зробити висновок, що просто наявність встановленого антивірусу може бути недостатньо для повноцінного захисту, і що потрібно використовувати додаткові методи.

Якщо поглянути, на приведені для прикладу причини пропуску вірусу антивірусом, можна побачити, що перші три причини пов'язані з неправильним використанням антивірусу, інші – з недоліками самого антивірусу і роботою виробника антивірусу. Відповідно і методи захисту діляться на два типи – організаційні і технічні.

**Організаційні методи**

Організаційні методи направлені насамперед на користувача комп'ютера. Їх мета полягає в тому, щоб змінити поведінку користувача, адже не секрет, що часто шкідливі програми потрапляють на комп'ютер через необдумані дії користувача. Простий приклад організаційного методу – розробка правил роботи за комп'ютером, яких повинні дотримуватися всі користувачі.

На домашньому комп'ютері користувач сам встановлює собі правила, яким він вважає потрібним слідувати. У міру накопичення знань про роботу комп'ютера і про шкідливі програми, він може свідомо міняти налаштування захисту або

ухвалювати рішення про небезпеку тих або інших файлів і програм.

У великій організації все складніше. Коли колектив об'єднує велика кількість співробітників, що виконують різні функції, складно чекати від усіх розумної поведінки з погляду безпеки. Тому в кожній організації правила роботи з комп'ютером повинні бути загальними для всіх співробітників і затверджені офіційно. Зазвичай, документ, що містить ці правила називається інструкцією користувача. При цьому інструкція користувача в більшості випадків містить тільки правила, дії, що обмежують його. Правила використання програм в інструкцію можуть входити тільки в самому обмеженому вигляді.

Але якщо не користувачі, то хтось інший все-таки повинен відповідати за налаштування засобів захисту і за управління ними. Звичайно, це спеціально призначений співробітник або група співробітників, які зосереджені на виконанні одного завдання – забезпечення безпечної роботи мережі.

Політика безпеки повинна давати відповіді на такі питання:

- які комп'ютери повинні бути захищені антивірусами та іншими програмами;

- які об'єкти повинні перевірятися антивірусом – чи потрібно перевіряти заархівовані файли, мережеві диски, вхідні і вихідні поштові повідомлення тощо;

- які дії повинен виконувати антивірус при виявленні зараженого об'єкту – іноді користувачі не можуть правильно вирішити, що робити з інфікованим файлом, отже антивірус повинен виконувати дії автоматично, не питаючи користувача.

**Технічні методи**

Технічні методи, навпаки, направлені на зміни в комп'ютерній системі. Більшість технічних методів полягають у використанні додаткових засобів захисту, які розширюють і доповнюють можливості антивірусних програм. Такими засобами захисту можуть бути:

- міжмережеві екрани – програми, що захищають від атак по мережі;

- засоби боротьби із спамом;

- виправлення, що знімають "дірки" в операційній системі, через які можуть проникати віруси.

Віруси нерідко проникають на комп'ютери через вразливості в операційній системі або встановлених програмах. Причому, найчастіше шкідливими програмами використовуються вразливості операційної системи MS Windows, пакету додатків MS Office, браузера Internet Explorer і поштової програми Outlook Express.

Для того, щоб не дати вірусам можливості використовувати вразливість, операційну систему і програмне забезпечення потрібно оновлювати.

Хоча більшість шкідливих програм використовують вразливості в продуктах Microsoft, існує немало і таких, які експлуатують дірки в програмах інших виробників. Особливо це стосується широко поширених програм обміну повідомленнями, браузерів і поштових клієнтів. Тому мало просто встановити браузер, відмінний від Internet Explorer, його теж потрібно періодично оновлювати. Найчастіше, в подібних програмах є вбудовані засоби оновлення, але незайвим буде стежити за новинами на веб-сайтах, присвячених питанням безпеки. Інформація про всі критичні вразливості і виправлення до них обов'язково потрапляє в новини.

Для того, щоб віддалено скористатися вразливістю в програмному забезпеченні або операційній системі, потрібно встановити з'єднання і передати спеціально сформований пакет даних. Отже можна захиститися від таких спроб проникнення і зараження, шляхом заборони певних з'єднань. Задачу контролю з'єднань успішно вирішують міжмережеві екрани.

**Міжмережевий екран** – це програма, яка стежить за мережевими з'єднаннями і ухвалює рішення про дозвіл або заборону нових з'єднань на підставі заданого набору правил.

Правило міжмережевого екрану задається декількома атрибутами:

- **Додаток** – визначає програму, до якої відноситься правило, так що одні і ті ж дії можуть бути дозволені одним

програмам і заборонені іншим.

- **Протокол** – визначає протокол, який використовується 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** – задає номери портів, на які розповсюджується правило.
- **Напрям** – дозволяє контролювати окремо вхідні та вихідні з'єднання.
- **Дія** – визначає реакцію на виявлення з'єднання, відповідно до перерахованих вище параметрів. Реакція може бути такою: дозволити, заборонити або запитати у користувача.

Не обов'язково давати конкретні значення всім атрибутам правила. Можна створити правило, яке заборонятиме вхідні з'єднання на порт 111 для всіх програмних додатків або дозволяти будь-які вихідні з'єднання для програми Internet Explorer.

Для боротьби з вірусами міжмережеві екрани можуть застосовуватися по-різному.

По-перше, міжмережевий екран можна успішно використовувати для захисту від шкідливих програм, які розповсюджуються безпосередньо по мережі, використовуючи вразливості в операційних системах. Міжмережеві екрани можна використовувати і для захисту від атак невідомих вірусів. У випадку домашнього комп'ютера, що використовує мережу тільки для доступу в Інтернет, можна заборонити взагалі всі вхідні з'єднання, і тим самим захистити себе від будь-яких атак ззовні.

Другий аспект застосування міжмережевих екранів для захисту від шкідливих програм полягає в контролі вихідних з'єднань. Багато троянських програм та черв'яків після виконання шкідливої функції прагнути подати сигнал авторові вірусу. Для того, щоб перешкодити цьому, можна побудувати міжмережевий екран таким чином, щоб він блокував всі невідомі з'єднання: дозволити тільки з'єднання від довірених

програм, таких як використовуваний браузер, поштовий клієнт, а решту з'єднань заборонити. У такому разі, шкідлива програма, навіть потрапивши на комп'ютер непоміченою, не зможе відправляти дані.

Деякі шкідливі програми не намагаються активно пересилати дані, а пасивно чекають з'єднання на якомусь з портів. Якщо вхідні з'єднання 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 відбулося, міжмережевий екран повинен бути налаштований на заборону вхідних з'єднань або на всі порти взагалі, або на всі, окрім фіксованого переліку портів, які використовують відомі програми або операційна система.

Останнім часом широко поширені універсальні захисні програми, які об'єднують можливості міжмережевого екрану та антивірусу.

**Хід роботи:**

Перед початком лабораторної роботи переконайтеся, що вхід у систему виконано під обліковим записом, що маєте права адміністратора.

**Завдання 1.** Вивчити налаштовування браузера.

У цьому завданні пропонується досліджувати явні прояви вірусної активності на прикладі несанкціонованої зміни настроювань браузера.

1. Відкрийте браузер Internet Explorer, скориставшись однойменним ярликом на робочому столі або в системному меню *Пуск/Программы*.

Повинна відкритися сторінка за замовчуванням.

2. Перевірте значення параметрів, що відповідають за стартову сторінку. Для цього потрібно скористатися меню *Сервис*. Відкрийте його й виберіть пункт *Свойства обозревателя*.

3. Адреса стартової сторінки зазначена у першому ж полі вікна *Свойства обозревателя*, на закладці *Общие*.

4. Змініть це поле, увівши адресу http://www.vspu.edu.ua.

5. Закрийте й знову відкрийте браузер. Переконаєтеся, що тепер першою була завантажена сторінка http://www.vspu.edu.ua.

Таким чином, якщо браузер почав самостійно завантажувати сторонній сайт, у першу чергу потрібно вивчити настроювання браузера: яка адреса виставлена в поле домашньої сторінки.

Ряд шкідливих програм обмежуються зміною цього параметра й для усунення наслідків зараження потрібно лише виправити адресу домашньої сторінки. Однак, це може бути тільки частиною шкідливого навантаження. Тому, якщо Ви виявили несанкціоновану зміну адреси домашньої сторінки, варто негайно встановити антивірусне програмне забезпечення й перевірити весь жорсткий диск на наявність вірусів.

**Завдання 2.** Визначення підозрілих процесів.

Одним з основних проявів шкідливих програм є наявність у списку запущених процесів підозрілих програм. Це часто допомагає при виявленні шкідливих програм, що мають лише приховані або непрямі прояви.

Однак необхідно чітко розуміти й уміти відрізняти легальні процеси від підозрілих.

1. Перейдіть до вікна *Диспетчер задач Windows*, натиснувши одночасно клавіші [Ctrl], [Shift] та [Esc].

2. Відкрите вікно містить чотири закладки, що відповідають чотирьом видам активності, які відслідковує Диспетчер: *Приложения, Процессы, Быстродействие* (використання системних ресурсів), *Сеть* та *Пользователи.* За замовчуванням повинна відкритися друга закладка, *Процессы.*

3. Уважно вивчіть представлений у вікні список процесів. Якщо на комп'ютері не запущені жодні користувацькі програми, він повинен містити тільки службові процеси операційної системи. Опис більшості процесів можна знайти в Інтернет.

4. Для кожного процесу виводяться його параметри: ім'я образа (може не збігатися з ім'ям файла, що його запускає), ім'я користувача, від імені якого був запущений процес, завантаження цим процесом процесора та обсяг займаної ним

оперативної пам'яті.

5. Відсортуйте всі процеси за використанням ресурсів процесора. Для цього натисніть на заголовок поля ЦП. Оскільки в цей момент не повинна бути запущена жодна користувацька програма, процесор повинен бути вільний.

6. У ряді випадків може знадобитися вручну завершити якийсь процес. Це можна зробити за допомогою кнопки *Завершить процесс* у контекстному меню.

7. Випишіть усі запущені процеси на аркуш паперу або в текстовий файл і перейдіть до закладки *Приложения.* Оскільки в цей момент не запущений жодний додаток, список запущених додатків порожній.

8. Не закриваючи вікна *Диспетчер задач Windows*, відкрийте програму *Paint.*

9. Не закриваючи додаток Paint, поверніться до вікна Диспетчера задач Windows і простежте за змінами на закладці *Приложения.* Список запущених додатків повинен містити рядок, який відповідає Paint. Оскільки він зараз працює, це записано в рядку *Состояние.*

10. Перейдіть до закладки *Процессы.* Порівняєте список запущених зараз процесів з переліком, складеним у п. 5 цього завдання. Знайдіть відмінність. З’ясуйте, який процес відповідає програмі Paint.

11. Перейдіть до закладки *Быстродействие.*

12. Уважно вивчите розташовані тут графіки. Будь-які сплески на них повинні за часом відповідати якимсь діям, наприклад запуску вимогливої до ресурсів програми. Якщо нічого схожого свідомо не запускалося, це може бути причиною для більш детального дослідження комп'ютера

13. Закрийте вікно *Диспетчер задач Windows.*

**Завдання 3**. Дослідити елементи автозапуску.

Для того, щоб прикладна програма почала виконуватися, її потрібно запустити. Отже, і вірус має потребу в тому, щоб його запустили. Для цього можна використати два сценарії: або зробити так, щоб користувач сам його стартував (використовуються обманні методи), або «залізти» в

конфігураційні файли й запускати одночасно з іншою, корисною програмою. Оптимальним з погляду вірусу варіантом є запуск одночасно з операційною системою – у цьому випадку запуск практично гарантований.

Найпростіший спосіб додати яку-небудь програму в автозавантаження – це помістити її ярлик у розділ *Автозагрузка* системного меню *Пуск->Программы*. За замовчуванням, відразу після установки операційної системи цей розділ порожній, оскільки ні однієї прикладної програми ще не встановлено.

1. Перевірте папку *Автозагрузка* на Вашому комп'ютері. Вона повинна бути порожня.

2. Додайте в список автозавантаження свою програму. Для цього двічі клацніть лівою клавішею миші за назвою групи *Автозагрузка.* Все, що потрібно зробити, щоб якась програма запускалася автоматично при старті операційної системи – це помістити в цю папку її ярлик.

3. Повторіть дії пункту 2, але тільки для папки *Пуск->Программы->Стандартные.* У вікні, що відкрилося, знайдіть ярлик *Блокнот.* Скопіюйте його до вікна *Автозагрузка.*

4. Закрийте вікно й переконаєтеся, що тепер розділ *Авто загрузка* в системному меню *Пуск->Программы* не порожній.

5. Перезавантажите комп'ютер (*Пуск->Завершение работы*) і ввійдіть у систему під своїм обліковим записом. Переконайтеся, що по завершенню завантаження автоматично запустилася програма *Блокнот.*

При обстеженні комп'ютера потрібно пам'ятати, що відсутність підозрілих ярликів у розділі *Автозагрузка* системного меню *Пуск->Программы* не гарантує, що жоден додаток не запускається автоматично. Технічно для автозавантаження потрібно додати відповідний запис до системного реєстру операційної системи.

6. Для більшості ситуацій, пов'язаних з автозавантаженням, досить використати системну утиліту *Настройка системы.*

7. Запустіть утиліту *Настройка системы*. Для цього відкрийте системне меню *Пуск* і перейдіть до пункту *Выполнить.*

8. У вікні *Запуск программы* наберіть *msconfig* і натисніть *ОК*.

9. На першій закладці *Общие* можна вибрати варіант запуску операційної системи. За замовчуванням відзначений *Обычный запуск*. Він забезпечує максимальну функціональність системи. Інші два варіанти запуску призначені для діагностування.

Другий режим, *Диагностичный запуск*, рекомендується використати також при вірусному інциденті, що підтвердився, – якщо комп'ютер уже заражений, відразу встановити антивірус у ряді випадків не можна, наприклад, якщо вірус свідомо блокує запуск ряду антивірусних програм. Тоді, якщо немає можливості видалити або хоча б тимчасово знешкодити вірус вручну, рекомендується запустити операційну систему в безпечному режимі, встановити антивірус і відразу ж перевірити весь жорсткий диск на наявність вірусів.

Для одержання додаткової інформації про цю закладку й інші, можна скористатися кнопкою *Справка.*

10. Перейдіть на закладку *Службы.* Кожна служба являє собою якийсь додаток, що працює у фоновому режимі. Однак, зараз ніяких сторонніх служб, крім системних, установлено бути не повинно. Переконайтеся в цьому, встановивши прапорець: *Не отображать службы Microsoft.* Якщо сторонніх додатків дійсно немає, список повинен стати порожнім.

11. Перейдіть до останньої закладки *Автозагрузка* і переконайтеся, що в списку додатків, що запускають автоматично при завантаженні системи, є *Блокнот*. Відзначимо, що список у вікні *Настройка системы* може містити додаткові елементи, не відображувані в розділі *Пуск->Программы->Автозагрузка.*

12. Відключіть автоматичне завантаження *Блокнота,* знявши прапорець у стовпці *Элемент автозагрузки* й натисніть *ОК*. У відкритому вікні дозвольте провести перезавантаження.

13. Дочекайтеся закінчення перезавантаження й увійдіть у систему під своїм обліковим записом.

14. Перейдіть до закладки *Автозагрузка* й переконаєтеся,

що її вид не змінився – *Блокнот* все так само присутній у списку, але відключений.

15. Перевірте, що *Блокнот* автоматично не запустився.

16. Поверніться до закладки *Общие* вікна *Настройка системы* й виберіть сценарій *Обычный запуск*. Натисніть *ОК* й у такому вікні виберіть *Перезагрузка.*

17. Дочекайтеся закінчення перезавантаження, увійдіть у систему під своїм обліковим записом і переконайтеся, що повідомлення про вибірковий запуск не з'являється.

18. Однак, оскільки прапорець, знятий в пункті 12, при перемиканні в режим *Обычный запуск* повернувся (*Обычный запуск* припускає завантаження всіх зареєстрованих компонентів), додаток *Блокнот* знову автоматично запускається по завершенні перезавантаження операційної системи. Переконаєтеся, що в *Пуск->Программы->Автозагрузка* повернувся ярлик *Блокнота.*

19. Видаліть його. Тепер автозавантаження чисте. Переконайтеся в цьому, виконавши перезавантаження й увійшовши в систему під своїм обліковим записом.

**Завдання 4.** Дослідити мережеву активність.

Зненацька підвищена мережева активність може служити яскравим свідченням роботи на комп'ютері підозрілих програм, які роблять несанкціоноване розсилання листів, зв'язується зі своїм автором і передає йому конфіденційну інформацію, просто завантажує свої додаткові модулі або атакує сусідній комп'ютер. Але при цьому потрібно не забувати, що ряд цілком легальних додатків також мають властивість іноді зв'язуватися із сайтом фірми-виробника, наприклад для перевірок наявності оновлень або більше нових версій.

1. Відкрийте вікно *Диспетчер задач Windows*, натиснувши одночасно клавіші [Ctrl], [Shift] та [Esc], і перейдіть до закладки *Сеть.* Оскільки зараз не ініціюється жодне мережеве з'єднання, графік повинен бути порожній, вірніше являти собою пряму на рівні 0 %.

2. Ініціюйте яке-небудь мережеве з'єднання. Наприклад, відкрийте браузер і завантажите сайт http://www.vspu.edu.ua.

Простежте за змінами на графіку *Диспетчер задач*: всі Ваші дії відобразяться на графіку у вигляді піків мережної активності, а значення поля *Использование сети* на деякий час перестане бути рівним нулю.

Диспетчер задач Windows показує тільки загальну інформацію. Для одержання більш докладних даних можна скористатися утилітою netstat.

4. Закрийте вікно *Диспетчер задач Windows* і перейдіть до системного меню *Пуск->Программы->Стандартные->Командный рядок*. Наберіть

netstat -a

і натисніть [Enter].

Результатом виконання команди є список активних підключень, у який входять установлені з'єднання й відкриті порти.

Частина портів пов'язана із системними службами Windows і відображається за назвою. Порти, що не стосуються стандартних служб, відображаються за номерами.

UDP-порти позначаються рядком UDP у колонці «Ім'я». Вони не можуть перебувати в різних станах, тому спеціальна позначка LISTENING у їхньому відношенні не використовується. Як і TCP-порти, вони можуть відображатися за іменами або номерами.

Порти, що використовуються шкідливими програмами, найчастіше є нестандартними і тому відображаються відповідно до їхніх номерів. Втім, можуть зустрічатися троянські програми, що використовують для маскування стандартні для інших додатків порти, наприклад 80, 21, 443 – порти, використовувані на файлових та веб-серверах.

5. Перевірте, як зміниться статистика, відображувана netstat при ініціюванні нових з'єднань. Команда netstat, на відміну від *Диспетчер задач Windows*, не працює в режимі реального часу, а відображає миттєву статистику.

6. Досліджуйте отриману статистику. Закрийте браузер, повторіть команду netstat -a і натисніть [Enter]. Переконайтеся, що всі викликані раніше мережні з'єднання закриті, а перелік

активних з'єднань не відрізняється від даних

7. Закрийте вікно командного рядка. Для цього введіть команду

exit і натисніть [Enter].

8. Іноді зібрані дані дозволяють визначити ім'я вірусу, тоді можна звернутися наприклад, до вірусної енциклопедії www.viruslist.com, щоб вручну ліквідувати наслідки зараження. Якщо однозначної відповіді одержати не вдається, необхідно зібрати всі підозрілі прояви й звернутися до Інтернет. На сьогоднішній день існує досить багато сайтів, що містять описи безпечних процесів, наприклад, www.processlibrary.com. Зрівнявши отримані в результаті аналізу дані із представленими в бібліотеці описами, потрібно залишити не заявлені як легальні процеси й об'єкти й простежити їхнє розташування на диску.

Подальші дії залежать від того, чи використовується на комп'ютері антивірусна програма чи ні. Якщо ні, то отримані файли потрібно досліджувати за допомогою антивірусної програми, наприклад онлайн сканера http://www.kaspersky.ru /virusscanner, що дозволяє безкоштовно перевіряти окремі об'єкти.

Якщо на комп'ютері антивірус уже встановлений, після виявлення підозрілих файлів варто звернутися в службу технічної підтримки антивірусної компанії, чий продукт використовується на комп'ютері, прикріпивши до повідомлення виявлені підозрілі об'єкти. Цілком можливо, вони містять новий, ще не відомий вірус.

Складіть звіт по роботі.

**Контрольні питання:**

1. Які види шкідливого програмного забезпечення Ви знаєте?

2. Що таке явні прояви шкідливого програмного забезпечення? Яким чином їх перевірити?

3. Перерахуйте непрямі ознаки наявності шкідливого програмного забезпечення? Яким чином їх перевірити?

4. У чому основна небезпека шкідливого програмного забезпечення з прихованими проявами. Як його виявити? За допомогою яких засобів?

5. Як перевірити процеси Windows та виділити з них підозрілі? Яким чином перевірити чи процес є легальним?

6. Що таке автозапуск? Навіщо його використовують шкідливі програми?

7. Перерахуйте всі місця де можна перевірити, чи знаходяться в автозапуску шкідливі програми.

8. Як перевірити мережеву активність? Перерахуйте всі відомі Вам засоби.

## Практична робота 3 (2 год.)

***Тема практичної роботи*: «Захист інформації шляхом чистки комп'ютера, перешкоджання збиранню "сміття" та відновлення після збоїв.**

***Мета роботи**:* Засвоїти роботу з програмами, що захищають комп'ютер користувача від можливостей відновлення файлів після їх знищення та програм, що унеможливлюють перегляд результатів роботи користувача під час роботи на комп'ютері і в мережі.

### Теоретичні відомості:

***"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"** (disk scavenging, garbage collection).

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 (erasure pattern)** – 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 (highewater marking)** – робить неможливим читання ділянок пам'яті, відведених для процесу записування, але не використаних ним. Верхня межа пам'яті, яка використовується, і є міткою повноти. Цей спосіб використовується для захисту послідовних файлів виняткового доступу (результуючі файли редакторів, компіляторів,

компоновщиків та їм подібних). Для індексних файлів і послідовних розділюваних файлів цей метод носить назву “стирання при розміщенні”, тобто пам’ять очищується при видаленні її процесу.

**Відновлення інформації після збоїв**

Важлива інформація може бути втрачена як через необмірковані дії користувачів, так і внаслідок збоїв у роботі техніки або внаслідок навмисного пошкодження даних зловмисником. Але існують програмні засоби, які дозволяють відновити інформацію після таких збоїв. Розглянемо деякі з цих програм.

**Програма EasyRecovery Pro**

Популярна програма для відновлення видалених або пошкоджених файлів. Усі налаштування програми EasyRecovery Pro (рис. 1**)** досить прості і доступні користувачу, хоча роботу може затруднити повністю англомовний інтерфейс.

У своєму складі програма містить утиліти, за допомогою яких можна виконати діагностику жорстких дисків на предмет наявності помилок:

- DriveTests – сканування приводів усіх накопичувачі на предмет потенційних не поладок;
- SMARTTests – моніторинг жорсткого диска для виявлення можливих пошкоджень;
- SizeManeger – надання інформації про використання дискового простору;
- JumperViewer – перевірка правильного підключення пристроїв;
- PartitionTests – аналіз існуючої структури файлової системи;
- DataAdvisor – створення завантажувальної дискети для діагностики системи.


Рис.1. Інтерфейс програми

Основу даної програми складають утиліти для безпосереднього відновлення даних:

- AdvancedRecovery – відновлення даних з використанням специфічних налаштувань;
- DeletedRecovery – знаходження і відновлення пошкоджених файлів;
- FormatRecovery – відновлення інформації з відформатованого або видаленого тома;
- RawRecovery – відновлення інформації з директорії;
- ResumeRecovery – перегляд LOG-файлів, створених під час роботи по відновленню даних;
- EmergencyDiskette – створення аварійної завантажувальної дискети.

Утиліти для відновлення конкретних файлів:

- AccessRepair – відновлення бази даних Microsoft Access;
- ExcelRepair – відновлення таблиці Microsoft Excel;
- PowerPointRepair – відновлення презентації Microsoft PowerPoint;
- WordRepair – відновлення документу Microsoft Word;
- ZipRepair – відновлення пошкодженого WinZip-архіва;
- EmailRepair – засоби для відновлення e-mail листів

Outlook Repair;

- Software Updates – засоби для оновлення програми;
- Crisis Center – тематичні каталоги з посиланнями, що дозволяють допомогти користувачу у кризисних ситуаціях.

## Програма FinalData

Дана програма дозволяє відновлювати інформацію, втрачену в результаті дії вірусів, форматування дисків або випадкового знищення файла.

Діалог програми (рис.2) дозволяє вибрати диск, на якому знищено всю пошкоджену інформацію і діапазон кластерів, в якому здійснювати пошук. У результаті сканування програма показує весь вміст диска на даний момент (як існуючі, так і видалені файли). Крім того, можна окремо вивести розділ видалених файлів і розділ втрачених файлів. Потім над файлом здійснюються певні операції по його відновленню.

Програма має зручний інтерфейс і розгалужену систему меню, зручну і зрозумілу у роботі.

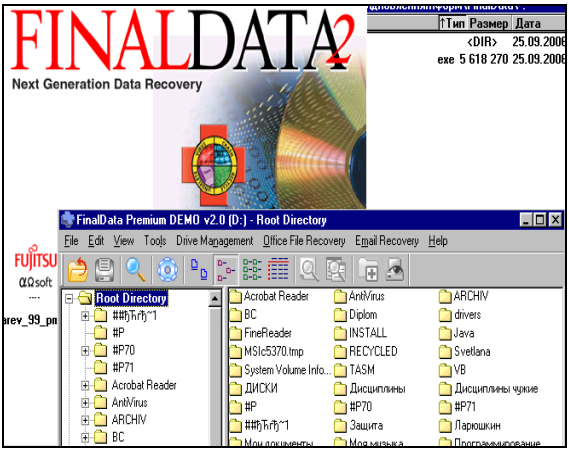



Рис. 2. Інтерфейс програми

### Програма GetDataBack

Це також популярна програма для відновлення пошкоджених та видалених файлів. Дії в ній виконуються за допомогою покрокового Майстра, який на першому кроці пропонує ряд налаштувань для встановлення параметрів:

- вибір каталогу, в який будуть копіюватися відновлені файли;
- вибір каталогу для тимчасових файлів;
- вибір шляху до програми обробки диска DiskExplorer, яка встановлюється у разі необхідності;
- можливість відновлення знищених файлів (навіть не прив'язаних до якогось певного каталогу;
- можливість використовувати або не використовувати Fat-таблицю (якщо знайдена на момент відновлення Fat не відповідає файловій системі).

На другому кроці здійснюється сканування вказаного диску, враховуючи вибрані на першому кроці параметри. При цьому за бажанням користувача пошук і сканування може здійснюватися як по всьому вибраному диску, так і на певній його частині (вказується початковий і кінцевий сектори), в одній чи іншій файловій системі (якщо ця опція не вказана, буде здійснюватись більш детальний пошук, що збільшить час сканування).

Далі, керуючись пропозиціями Майстра, можна відновлювати певні файли і каталоги. Слід зауважити, що не слід копіювати відновлені дані на той диск, з якого виконується відновлення.

### HDDLife (ChipCD 11-05)

Завдяки HDDLife можна визначити збої в роботі жорстких дисків і, якщо виникли неполадки, встигнути вчасно замінити вінчестер на новий, зберігши інформацію. Вона оцінює «здоров'я» жорсткого диска по стобальній системі і візуально представляє одержані дані. HDDLife може працювати у фоновому режимі, контролюючи також температуру жорсткого диска. Якщо вона зросте більш ніж на 10 градусів, програма

обов'язково повідомить вас про це. Працює HDDLife з вінчестерами IDE і Serial ATA.


Рис.3. Інтерфейс програми

**Saver (ChipCD 08-05)**

Програма служить для збереження і завантаження настройок програм і ігор. Програму зручно використовувати перед перестановкою\зміною операційної системи або для резервування налаштувань. При збереженні налаштувань економиться багато часу, терпіння і сил, не треба кожного разу забивати одні і ті ж дані, достатньо запустити цю програму і вибрати завантаження налаштувань для виділеної програми або всі доступні збереження. Також програма допоможе, якщо раптом ви видалите якусь програму і забудете зберегти ваші налаштування до неї (навіщо десь шукати, якщо можна натиснути 1 кнопку?) або якщо полетить система.

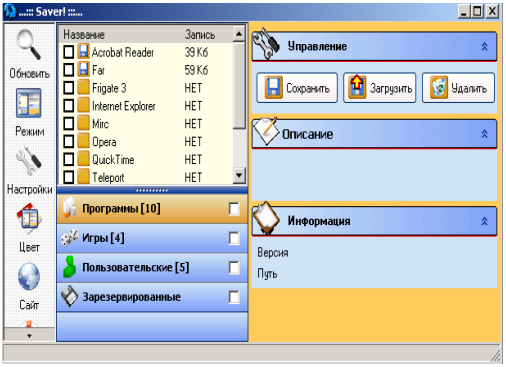



Рис.4. Інтерфейс програми

**Your Uninstaller (ChipCD 07-05)**

Іноді не вдається видалити яку-небудь непотрібну більш утиліту, оскільки містичним чином пропав її деінсталятор. У цьому випадку на допомогу можна призвати програму Your Uninstaller. Ця утиліта може не просто видалити будь-який додаток, але і очистити систему від всіх слідів його перебування (файли, ключі реєстру). Цим можливості програми не обмежуються. Your Uninstaller може також попереджати про програми, що додають свої ключі в гілці реєстру, що відповідають за автозапуск.

Рис.5. Інтерфейс програми

## Absolute Uninstaller (ChipCD 09-05)

Бувають програми, які встановлюються на комп'ютер нормально, але видалятися не хочуть, або ж їх взагалі не видно в списку сервісу "Установка і видалення програм". Саме в таких випадках стане в нагоді програма Absolute Uninstaller. Вона знайде все встановлене програмне забезпечення і видалить будь-яку не видалену повністю програму.

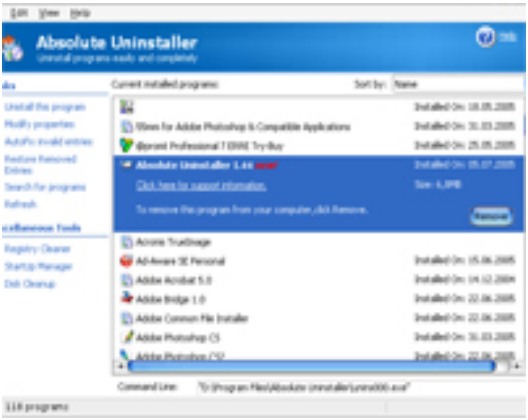



Рис.6. Інтерфейс програми

## Rapid File Defragmentor (ChipCD 07-05)

Дефрагментація жорсткого диска необхідна, але достатньо тривала процедура.

За допомогою програми Rapid File Defragmentor ви зможете дефрагментувати тільки найбільш фрагментовані файли. Цей шлях набагато простіший і швидший, ніж дефрагментація цілого диска. За декілька секунд програма знайде на вашому комп'ютері файли і каталоги, яким необхідна дефрагментація, і також швидко проведе її. А щоб прискорити роботу конкретного додатку, наприклад, Outlook, необхідно вказати Rapid File Defragmentor файли, в яких зберігаються архіви листів (це основний масив, який найбільш триваліше обробляється програмою).

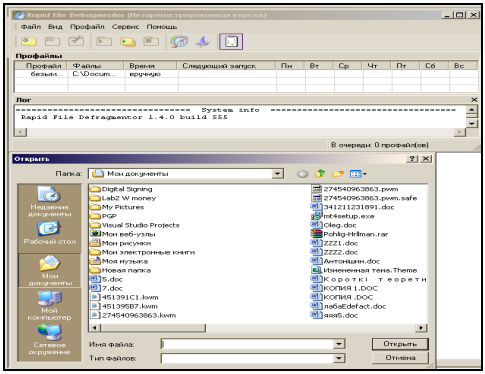


Рис.7. Інтерфейс програми

## Програми для “чистки” комп’ютера після роботи користувача

### Privacy Inspector (Shareware, сайт розробника: http://www.privacyinspector.com)

Утиліта для видалення інформації про те, чим займався користувач, працюючи за комп'ютером. Видаляє cookies браузера, очищає листи із списками документів, що недавно відкрилися, листи пошукових запитів, історії пункту «Виконати» меню «Пуск», а також звільняє місце на жорсткому диску. Крім того, натисненням «гарячої» клавіші підтримується виконання вказаних операцій і закриття всіх активних вікон.

### MyPrivacy 3.0.2 ( Freeware, сайт розробника: http://www.omniquad.com)

Бажаючим зберегти конфіденційність достатньо, звичайно, кожного разу після роботи вручну очищати тимчасові теки, але це дуже довгий і, прямо скажемо, не кращий вихід. Набагато простіше скористатися утилітою з вельми показовою назвою MyPrivacy. Вона істотно спрощує і автоматизує процес знищення конфіденційної інформації користувачем. На двох її основних закладках Internet Privacy і Windows Privacy представлені практично всі уразливі з погляду приватності роботи компоненти системи взагалі і Internet Explorer зокрема. Зайшовши на першу, можна не тільки очистити кеш браузера і

його історію, але і видалити взагалі всі згадки про відвідані сторінки (Clean All Typed Urls), а також відфільтрувати одержані cookies (Manage Cookies).

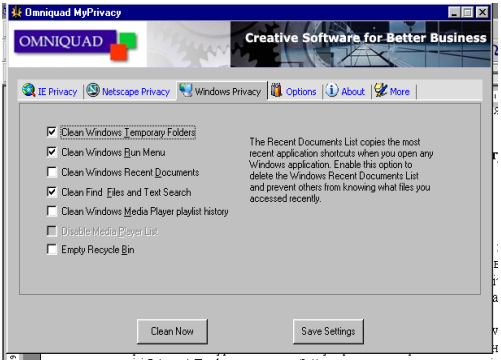



Рис.8. Інтерфейс програми

**Easy Cleaner 1.7f (Freeware, сайт розробника: http://www.toniarts.com/ecleane.htm)**

Часто можна помітити, що комп'ютер став дещо пригальмовувати, а місця на диску з кожним днем стає все менше. Отже, пора прийматися за “прибирання”. Роль пилососа виконає утиліта Easy Cleaner.

Відкривши, наприклад, перший розділ в її головному вікні і натиснувши кнопку "Знайти", ми запустимо процес сканування ключів реєстру. Помилкові записи з'являтимуться у вигляді списку. Процес сканування, в порівнянні з деякими аналогічними програмами, багато часу не займе. Річ у тому, що Easy Cleaner шукає не все підряд, а винятково тільки ті посилання, що вказують на відсутні в системі файли і теки. Їх можна тут же виділити і видалити. У разі виникнення якої-небудь помилки або збою можна звернутися до автоматично створюваних програмою REG-файлів відкату, а якщо і це не допоможе, до наперед скопійованих копій реєстру.

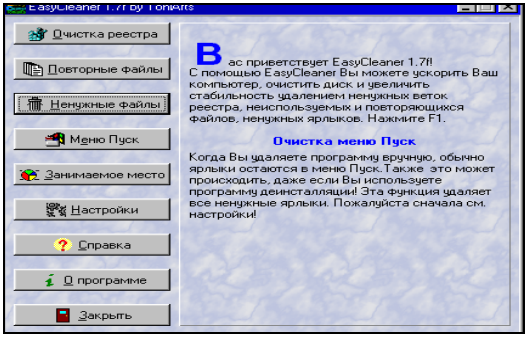



Рис.9. Інтерфейс програми

Слід бути обережними при зверненні до другого розділу Easy Cleaner "Повторні файли". У жодному випадку не слід видаляти нічого з системних директорій, переконатися, що в настройках програми прописано видалення знайденого в "Корзину". Та і для сторінки "Непотрібні файли" бажано задати ігнорування системних файлів. А ось з меню "Пуск" за допомогою однойменного розділу все зайве видаляти можна сміливо.

**Програми для «видалення сміття»**

CleanCenter
CleanDiskSecurity
(Захист 2)
BlackBoard
(Захист 2)
Evidance Eliminator
(Захист 2)
WinXP Manager (закладка Cleaner)
(ChipCD 11-05)
BC Wipe
(ChipCD 08-05)
(ChipCD 01-05)

**Хід роботи:**

1. Вивчити програми описані у теоретичній частині.
2. Встановити описані програми на віртуальній машині.
3. Зробити копію всіх даних на флешці.
4. Знищити всі файли з флешки.
5. Спробувати відновити дані, що були знищені.
6. Відформатувати флешку.
7. Спробувати відновити дані, що були знищені.
8. Результати помістити у звіт.

**Контрольні запитання:**

1. Що означає «Збір сміття»?
2. Як відбувається відновлення інформації після збоїв?
3. Що таке «Стираючий зразок»?
4. Що означає «Мітка повноти»?
5. Назвіть популярні програми для відновлення видалених або пошкоджених файлів.

## Важливість і складність проблеми інформаційної безпеки

Інформаційна безпека є одним з найважливіших аспектів інтегральної безпеки, на якому б рівні ми не розглядали останню – національному, галузевому, корпоративному або персональному.

Для ілюстрації цього положення наведемо кілька прикладів.

З'явилась інформація про те, що планується терористична атака Нью-Йорської Біржі. Ціллю терористів є комп'ютерні системи, що зберігають і працюють з інформацією про торгові операції в США та Європі. Наслідки такої операції можуть призвести до криз світового масштабу. (З інтерв'ю з М. Дюре, директором Центру інформації та документації НАТО в Україні).

Американський ракетний крейсер "Йорктаун" був вимушений повернутися в порт через численні проблеми з програмним забезпеченням, що функціонувало на платформі Windows NT. Таким виявився побічний ефект програми ВМФ США з максимально широким використанням комерційного програмного забезпечення з метою зниження вартості військової техніки.

У лютому 2001 року двоє колишні співробітники компанії Commerce One, скориставшись паролем адміністратора, видалили з сервера файли, що складали крупний (на декілька мільйонів доларів) проект для іноземного замовника. На щастя, була резервна копія проекту, так що реальні втрати обмежилися витратами на слідство і засоби захисту від подібних інцидентів у майбутньому. У серпні 2002 року злочинці постали перед судом.

Британський спеціаліст з інформаційних технологій Максвелл Парсонс отримав 2,5 роки ув'язнення за злом банкоматів за допомогою MP3-плеєра і спеціального програмного забезпечення. Таким чином, він отримував конфіденційну інформацію про банківські рахунки клієнтів для клонування кредитних карток.

Американські військові оголосили про створення Командного центру кіберпростору ВВС США (U.S. Air Force Cyberspace Command) для захисту країни від онлайнових загроз з Інтернету.

Невідомі "жартівники" скористалися принципами роботи онлайнової енциклопедії Wikipedia для розповсюдження шкідливого програмного забезпечення – нової модифікації вірусу Blaster.

Одна студентка втратила стипендію в 18 тисяч доларів в Мічиганському університеті через те, що її сусідка по кімнаті скористалася їх загальним системним входом і відправила від імені своєї жертви електронний лист з відмовою від стипендії.

Зрозуміло, що подібних прикладів багато, можна пригадати і інші випадки - недоліку в порушеннях ІБ немає і не передбачається.

При аналізі проблематики, пов'язаної з інформаційною безпекою, необхідно зважати на специфіку даного аспекту безпеки, що полягає в тому, що ІБ є складова частина інформаційних технологій, - області, що розвивається безпрецедентно високими темпами.

На жаль, сучасна технологія програмування не дозволяє створювати безпомилкові програми, що не сприяє швидкому розвитку засобів забезпечення ІБ. Слід виходити з того, що необхідно конструювати надійні системи ІБ із залученням ненадійних компонентів (програм). У принципі, це можливо, але вимагає дотримання певних архітектурних принципів і контролю стану захищеності на всьому протязі життєвого циклу ІС.

Приведемо ще декілька цифр. У березні 1999 року був опублікований черговий, четвертий річний звіт "Комп'ютерна злочинність і безпека-1999: проблеми і тенденції" (Issues and Trends: 1999 CSI/FBI Computer Crime and Security Survey). У звіті наголошується різке зростання числа звернень в правоохоронні органи з приводу комп'ютерних злочинів (32% з числа опитаних); 30% респондентів повідомили про те, що їх інформаційні системи були зламані зовнішніми зловмисниками;

атакам через Internet піддавалися 57% опитаних; у 55% випадках наголошувалися порушення з боку власних співробітників. Примітно, що 33% респондентів на питання чи "були зламані ваші Web-сервери і системи електронної комерції за останні 12 місяців?" відповіли "не знаю".

У аналогічному звіті, опублікованому в квітні 2002 року, цифри змінилися, але тенденція залишилася такою ж: 90% респондентів (переважно з крупних компаній і урядових структур) повідомили, що за останні 12 місяців в їх організаціях мали місце порушення інформаційної безпеки; 80% констатували фінансові втрати від цих порушень; 44% (223 респонденти) змогли та/або захотіли оцінити втрати кількісно, загальна сума склала більше 455 млн. доларів. Найбільшого збитку завдали крадіжки і фальсифікації (більше 170 і 115 млн. доларів відповідно).

Збільшення числа атак – ще не найбільша неприємність. Гірше те, що постійно виявляються нові вразливі місця в програмному забезпеченні і, як наслідок, з'являється новий вигляд атак.

У таких умовах системи ІБ повинні уміти протистояти різноманітним атакам, як зовнішнім, так і внутрішнім, атакам автоматизованим і скоординованим. Іноді напад триває долі секунди; деколи прощупування вразливих місць ведеться поволі і розтягується на години, так що підозріла активність практично непомітна. Метою зловмисників може бути порушення всіх складових ІБ – доступності, цілісності або конфіденційності.

## Огляд зарубіжного законодавства в області інформаційної безпеки

Переглянемо деякі закони кількох країн (в першу чергу – США), оскільки тільки в США таких законодавчих актів близько 500.

Ключову роль відіграє американський «Закон про інформаційну безпеку» (Computer Security Act of 1987, Public Law 100-235 (H.R. 145), January 8, 1988). Його мета – реалізація мінімально достатніх дій із забезпечення безпеки інформації у федеральних комп'ютерних системах, без обмежень всього спектру можливих дій.

Характерно, що вже на початку Закону називається конкретний виконавець – Національний інститут стандартів і технологій (НІСТ), що відповідає за випуск стандартів, направлених на захист від знищення і несанкціонованого доступу до інформації, а також від крадіжок і фальсифікацій, що виконуються за допомогою комп'ютерів. Таким чином, йдеться про регламентацію дій фахівців і підвищення інформованості всього суспільства.

Згідно Закону, всі оператори федеральних ІС, що містять конфіденційну інформацію, повинні сформувати плани забезпечення ІБ. Обов'язковим є і періодичне навчання всього персоналу таких ІС. НІСТ, у свою чергу, зобов'язаний проводити дослідження природи і масштабу вразливих місць, виробляти економічно виправдані заходи захисту. Результати досліджень розраховані на застосування не тільки в державних системах, але і в приватному секторі.

Закон зобов'язує НІСТ координувати свою діяльність з іншими міністерствами і відомствами, включаючи Міністерство оборони, Міністерство енергетики, Агентство національної безпеки (АНБ) і так далі, щоб уникнути дублювання і несумісності.

Крім регламентації додаткових функцій НІСТ, Закон наказує створити при Міністерстві торгівлі комісію з інформаційної безпеки, яка повинна:

- виявляти перспективні управлінські, технічні,

адміністративні і фізичні заходи, що сприяють підвищенню ІБ;

- видавати рекомендації Національному інституту стандартів і технологій, доводити їх до всіх зацікавлених відомств.

З практичної точки зору важливий розділ 6 Закону, що зобов'язує всі урядові відомства сформувати план забезпечення інформаційної безпеки, направлений на те, щоб компенсувати ризики і запобігти можливому збитку від втрати, неправильного використання, несанкціонованого доступу або модифікації інформації у федеральних системах. Копії плану пересилаються в НІСТ і АНБ.

У 1997 році з'явилося продовження описаного закону – законопроект «Про вдосконалення інформаційної безпеки» (Computer Security Enhancement Act of 1997, H.R. 1903), направлений на посилення ролі Національного інституту стандартів і технологій і спрощення операцій з криптозасобами.

У законопроекті констатується, що приватний сектор готовий надати криптозасоби для забезпечення конфіденційності і цілісності (зокрема автентичності) даних, що розробка і використання шифрувальних технологій повинні відбуватися на підставі вимог ринку, а не розпоряджень уряду. Крім того, тут наголошується, що за межами США є зіставні і загальнодоступні криптографічні технології, і це слід враховувати при виробленні експортних обмежень, щоб не знижувати конкурентоспроможність американських виробників апаратного і програмного забезпечення.

Для захисту федеральних ІС рекомендується ширше застосовувати технологічні рішення, засновані на розробках приватного сектора. Крім того, пропонується оцінити можливості загальнодоступних зарубіжних розробок.

Дуже важливий розділ 3, в якому від НІСТ потрібно по запитах приватного сектора готувати добровільні стандарти, інструкції, засоби і методи для інфраструктури відкритих ключів (див. вище Закон РФ про ЕЦП), що дозволяють сформувати недержавну інфраструктуру, придатну для взаємодії з федеральними ІС.

У розділі 4 особлива увага звертається на необхідність аналізу засобів і методів оцінки вразливих місць інших продуктів приватного сектора в області ІБ.

Вітається розробка правил безпеки, нейтральних по відношенню до конкретних технічних рішень, використання у федеральних ІС комерційних продуктів, участь у реалізації шифрувальних технологій, що дозволяє зрештою сформувати інфраструктуру, яку можна розглядати як резервну для федеральних ІС.

Важливо, що відповідно до розділів 10 і далі передбачається виділення конкретних (і чималих) сум, називаються точні терміни реалізації програм партнерства і проведення досліджень інфраструктури з відкритими ключами, національної інфраструктури цифрових підписів. Зокрема, передбачається, що для засвідчуючих центрів повинні бути розроблені типові правила і процедури, порядок ліцензування, стандарти аудиту.

У 2001 році був схвалений Палатою представників і переданий в Сенат новий варіант розглянутого законопроекту – Computer Security Enhancement Act of 2001 (H.R. 1259 RFS). У цьому варіанті примітно як те, що, в порівнянні з попередньою редакцією, було прибрано, так і те, що додалося.

За чотири роки (1997-2001 рр.) на законодавчому і інших рівнях інформаційної безпеки США було зроблено багато що. Пом'якшені експортні обмеження на криптозасоби (у січні 2000 р.). Сформована інфраструктура з відкритими ключами. Розроблено велике число стандартів (наприклад, новий стандарт електронного цифрового підпису — FIPS 186-2, січень 2000 р.). Все це дозволило не загострювати більше увагу на криптографії як такій, а зосередитися на одному з її найважливіших застосувань — аутентифікації, розглядаючи її по відпрацьованій на криптозасобах методиці. Очевидно, що, незалежно від долі законопроекту, в США буде сформована національна інфраструктура електронної аутентифікації. В даному випадку законотворча діяльність йде в ногу з прогресом інформаційних технологій.

Програма безпеки, що передбачає економічно виправдані захисні заходи і синхронізована з життєвим циклом ІС, згадується в законодавстві США неодноразово. Згідно пункту 3534 («Обов'язки федеральних відомств») підрозділи II («Інформаційна безпека») розділу 35 («Координація федеральної інформаційної політики») рубрик 44 («Суспільні видання і документи»), така програма повинна включати:

- періодичну оцінку характеристик з розглядом внутрішніх і зовнішніх погроз цілісності, конфіденційності і доступності систем, а також даних, що асоціюються з критично важливими операціями і ресурсами;
- правила і процедури, що дозволяють, спираючись на проведений аналіз, економічно виправданим чином зменшити ризики до прийнятного рівня;
- навчання персоналу з метою інформування про існуючі ризики і про обов'язки, виконання яких необхідне для їх (ризиків) нейтралізації;
- періодичну перевірку і оцінку ефективності правил і процедур;
- дії при внесенні істотних змін до системи;
- процедури виявлення порушень інформаційної безпеки і реагування на них; ці процедури повинні допомогти зменшити ризики, уникнути крупних втрат; організувати взаємодію з правоохоронними органами.

Звичайно, в законодавстві США є в достатній кількості і положення обмежувальної спрямованості, і директиви, що захищають інтереси таких відомств, як Міністерство оборони, АНБ, ФБР, ЦРУ, але ми на них не зупинятимемося. Про це можна прочитати розділ «Законодавча база в області захисту інформації» у статті О. Беззубцева і А. Ковальова «Про ліцензування і сертифікацію в області захисту інформації» (Jet Info, 1997, 4).

У законодавстві ФРН виділимо вельми розгорнений (44 розділи) Закон про захист даних (Federal Data Protection Act of December 20, 1990 (BGB1.I 1990 S.2954), amended by law of September 14, 1994 (BGB1. I S. 2325)). Він цілком присвячений

захисту персональних даних.

Ймовірно, що і у всіх інших законах аналогічної спрямованості, в даному випадку встановлюється пріоритет інтересів національної безпеки над збереженням таємниці приватного життя. У останньому права особи захищені вельми ретельно. Наприклад, якщо співробітник фірми обробляє персональні дані на користь приватних компаній, він дає підписку про не розголошування, яка діє і після переходу на іншу роботу.

Державні установи, зберігаючи і обробляючи персональні дані, несуть відповідальність за порушення таємниці приватного життя «суб'єкта даних», як мовиться в Законі. У матеріальному відношенні відповідальність обмежена верхньою межею в 250 тисяч німецьких марок.

У сучасному світі глобальних мереж законодавча база повинна бути узгоджена з міжнародною практикою. У цьому плані – приклад Аргентини. В кінці березня 1996 року компетентними органами Аргентини був арештований Хуліо Цезар Ардіта, 21 року, житель Буенос- Айреса, системний оператор електронної дошки оголошень «Крик», відомий в комп'ютерному підпіллі під псевдонімом «E1 Griton». Йому ставилися в провину систематичні вторгнення в комп'ютерні системи ВМС США, НАСА, багатьох найбільших американських університетів, а також в комп'ютерні системи Бразилії, Чилі, Кореї, Мексики і Тайваню. Проте, не дивлячись на тісну співпрацю компетентних органів Аргентини і США, Ардіта був відпущений без офіційного пред'явлення звинувачень, оскільки за аргентинським законодавством вторгнення в комп'ютерні системи не вважається злочином. Крім того, через принцип «подвійної кримінальності», що діє в міжнародних правових відносинах, Аргентина не може видати хакера американській владі. Справа Ардіта показує, яким може бути майбутнє міжнародних комп'ютерних вторгнень за відсутності загальних або хоч би двосторонніх угод про боротьбу з комп'ютерною злочинністю.

**Література:**

1.Антонюк А.О. Основи захисту інформації в автоматизованих системах. Навчальний посібник / А.О. Антонюк. – Київ. Видавничий дім «КМ Академія», 2003.– 342 с.

2.Баранов А. П. Безопасность информационных технологий / А. П.Баранов, Д. П.Зегжда, П. Д. Зегжда –СПб.: DiaSoft, 2002. – 688 с.

3.Белкин Н. Ю. Программно-аппаратные средства обеспечения информационной безопасности. Защита программ и данных / Н. Ю. Белкин, О. О. Міальський, А.С. Першаков – М.: Радио и связь, 1999. –168 с.

4.Галатенко В.А. Информационная безопасность: практический подход / В.А. Галатенко. Под ред. Бетелина В.Б.- М.: Наука, 1998.

5.Герасименко В. А. Основы защиты информации / В. А. Герасименко. – М: Инкомбук, 1997. – 537 с.

6.Домарев В. В. Безопасность информационных технологий / В. В. Домарев – СПб.: DiaSoft, 2002. – 688 с.

7.Домарев В. В. Безопасность информационных технологий. Методы создания систем защиты / В. В. Домарев – К.: ООО ТИД ДС, 2001. – 688 с.

8.Захист інформації. Технічний захист інформації. Терміни і визначення. – К.: Держстандарт України, 1998.

9.Защита компьютерных систем от разрушающих программных воздействий: Руководство к практическим занятиям / Под ред. проф. П. Д. Зегжды. — СПб., 1998. — 128 с.

10. Зегжда Д. П. Теоретические основы информационной безопасности. Защищенные операционные системы: Руководство к практическим занятиям / Д. П. Зегжда, М. О. Калинин, П. Г. Степанов. – СПб., 1998. – 69 с.

11. Касперский Е. “Дыры” в MS-DOS и программы защиты информации / Е. Касперский – М.: Компьютер-Пресс, 1991.

12. Конев И. Информационная безопасность предприятия / И. Конев, А. Беляев – СПб.: БХВ Петербург, 2003. — 752 с.

13. Кудрявцева С.П. Міжнародна інформація:

Навчальний посібник для студентів вищих навчальних закладів / С.П Кудрявцева, В.В. Колос. – К.: Видавничий Дім «Слово», 2005. – 400 с.

14. Медведовский И.Д. Атака на Интернет / И.Д. Медведовский, П.В., Семьянов, Д.Г. Леонов. –2-е изд. перераб. и доп. – М.: ДМК, 1999. – 336 с.

15. Методы и средства защиты информации / Под ред. Ю. С. Ковтанюка. – К.: ЮНИОР, 2003. — 501 с.

16. Методы и средства защиты информации / Под ред. Ю. С. Ковтанюка. – К.: ЮНИОР, 2003. — 501 с.

17. О вирусах, червях, троянцах и бомбах. Защита информации: Переводы. – М.: Знание, 1990. — Новое в жизни, науке и технике. Сер. “Вычислительная техника и ее применение”.

18. Петров А. А. Компьютерная безопасность. Криптографические методы защиты / А. А. Петров. – М: ДМК, 2000. – 448 с.

19. Трубачев А.П. и др. Оценка безопасности информационных технологий. Под общ. ред. Галатенко В.А. - М.: СИП РИА, 2001.

20. Ходаков В.Є. Вступ до комп’ютерних систем. Навчальний посібник / В.Є. Ходаков, М.В. Пилипенко, Н.А. Соколова. За ред. д.т.н., проф., заслуж. діяча науки й техніки Ходакова В.Є. К. : Центр навчальної літератури, 2005. – 496 с.

**Сайти з теми ІБ:**

1. Библиотека Конгресу США: http://thomas.loc.gov/.

2. Домашня сторінка відомого спеціаліста в області інформаційної безпеки Дороті Денінга: http://www.cs.georgetown.edu/denning/.

3. Журнал з інформаційної безпеки: http://www.securitymagazine.com/.

4. Класичний хакерський журнал 2600: http://www.2600.com/.

5. Підбірка відповідей на часті запитання з інформаційної безпеки: http://www.faqs.org/faqs/computer-security/.

6. Подбірка матеріалів з інформаційної безпеки: http://www.linuxsecuri ty. com/.

7. Сервер з матеріалами по законодавству Великобританії: http://www.hmso.gov.uk/.

8. Сервер Інституту інформаційної безпеки: http://www.gocsi.com/.

9. Сервер координаційної ради групи реагування на порушення ІБ: http://www.cert.org/.

10. Сервер новин з ІБ: http://www.infosecnews.com/.

11. Сервер оперативної інформації UA/Security Line: http://hack.com.ua/.

12. Сервер с архівами повідомлень і з можливістю підписки на списки розсилок з ІБ: http://www.securityfocus.com/.

13. Сервер з хакерським іміджем: http://www.insecure.org/.

14. Сервер Тематичної групи по технології Internet: http://www.ietf.org/.

15. Web-сервер Інституту національної безпеки США: http: //www. nsi. org/.

16. Web-сервер Національного інституту стандартів США: http://www.nist.gov/.

17. Web-сервер підрозділу Міністерства юстиції США, призначений для освітлення питань кіберзлочинності: http://www. cybercri me .gov/.

**Зміст**

Передмова……. 3
Лекція 1. Поняття інформаційної безпеки…… 4
Лекція 2. Проблема захисту операційних систем……. 23
Лекція 3. Характеристика найпоширеніших загроз безпеці комп'ютерних систем……. 40
Лекція 4. Віруси як шкідливе програмне забезпечення….. 61
Лекція 5. Пакування, архівація і шифрування даних в операційних системах…….. 132
Лекція 6. Найпростіші методи захисту інформації в операційних системах…….. 148
Практична робота 1. Приховування файлів…… 166
Практична робота 2,3. Основні ознаки присутності на комп'ютері шкідливих програм……... 173
Практична робота 4. Захист інформації шляхом чистки комп'ютера, перешкоджання збиранню "сміття" та відновлення після збоїв…..……. 192
Важливість і складність проблеми інформаційної безпеки…… 204
Огляд зарубіжного законодавства в області інформаційної безпеки…… 207
Література…….212
Сайти з теми ІБ…….213

Навчальне видання

**КИРИЛЕНКО Неля Михайлівна**

**ІНФОРМАЦІЙНА БЕЗПЕКА**
**навчально-методичний посібник**

*Редактор А.М. Мельниченко*
*Комп'ютерна верстка О.В. Ступак*
*Коректор О.В. Петрова*

Здано до набору 20.10.2011.
Підписано до друку 01.11.2011.
Формат 60x84/16. Папір офсетний.
Гарнітура BALTICA. Друк офсетний.
Ум.друк арк.5, 2. Зам. №27-01
Наклад 100 прим.

Віддруковано з готових діапозитивів ЧТ «Эдельвейс»