

МІНІСТЕРСТВО ВНУТРІШНІХ СПРАВ УКРАЇНИ

Харківський національний університет внутрішніх справ

Факультет № 4

Кафедра інформаційних технологій

ТЕКСТ ЛЕКЦІЇ

**з дисципліни «Операційні системи та комп'ютерні мережі»
за темою «Основні вбудовані механізми захисту ОС та їх недоліки»**

Галузь знань: 12 "Інформаційні технології "

Спеціальність: 125 "Кібербезпека"

Ступінь вищої освіти - бакалавр

**м. Харків
2017 р.**

Передмова

СХВАЛЕНО

Науково-методичною радою ХНУВС

_____ Протокол № _____

(дата, місяць, рік)

ЗАТВЕРДЖЕНО

Вченою радою факультету № 4

ХНУВС

_____ Протокол № _____

(дата, місяць, рік)

_____ (підпис)

_____ (П.І.Б.)

ПОГОДЖЕНО

Секцією Науково-методичної ради
ХНУВС з технічних дисциплін

_____ Протокол № _____

(дата, місяць, рік)

_____ (підпис)

_____ (П.І.Б.)

ЗАТВЕРДЖЕНО

На засіданні кафедри інформаційних
технологій

_____ Протокол № _____

(дата, місяць, рік)

_____ (підпис)

_____ (П.І.Б.)

Рецензент:

Зацеркляний М.М., доктор технічних наук, професор;

Розробники: Можєв О.О. – Харків: Харківський національний університет
внутрішніх справ, 2017

© Можєв О.О., 2017

© Харківський національний
університет внутрішніх справ

План лекції

1. Аналіз захищеності сучасних операційних систем.
2. Основні захисні механізми ОС ряду UNIX.
3. Основні захисні механізми ОС ряду WINDOWS
4. Аналіз існуючих статистик загроз для сучасних універсальних ОС
5. Огляд і статистика методів, що знаходяться в основі атак на сучасні ОС.

Література:

Основна:

1. Барановская Т.П. Архитектура компьютерных систем и сетей: Учеб. пособие / Т.П. Барановская, В.И. Лойко и др.; под ред. В.И. Лойко. – М.: Финансы и статистика, 2003. – 256с.
2. Пятибратов А. П. Вычислительные системы, сети и телекоммуникации: Учебник. – 2-е изд., перераб. и доп. / А. П. Пятибратов, Л. П. Гудыно, А. А. Кириченко; Под ред. А. П. Пятибратова – М.: Финансы и статистика, 2004. – 512с.
3. Струков В.М. Комп'ютерні основи систем кібербезпеки/ Зацеркляний М.М., Струков В.М. – Харків, ХНУВС, 2017. – 274с.
4. Олифер В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – СПб.: Питер, 2001. – 672 с.
5. Цилькер Б. Я. Организация ЭВМ и систем / Б. Я. Цилькер, С.А. Орлов. СПб.: Питер, 2006. - 668 с.
6. Столлингс В. Структурная организация и архитектура компьютерных систем. / В. Столлингс - М.: Издательский дом "Вильямс", 2002. - 896с.

Текст лекції

1. Аналіз захищеності сучасних операційних систем.

Під механізмами захисту операційної системи розуміються всі засоби і механізми захисту даних, що функціонують у складі операційної системи. Операційні системи, у складі яких функціонують засоби і механізми захисту даних, часто називаються захищеними системами.

Під безпекою операційної системи розуміється такий її стан, при якому неможливе випадкове чи навмисне порушення функціонування операційної системи, а також порушення безпеки, при яких ресурси комп'ютерної системи знаходяться під управлінням операційної системи.

Вкажемо особливості операційної системи, які дозволяють виділити питання забезпечення її безпеки в особливу категорію:

- управління всіма ресурсами системи;

- наявність вбудованих механізмів, які прямо чи опосередковано впливають на безпеку програм і даних, що працюють у середовищі операційної системи;

- забезпечення інтерфейсу користувача з ресурсами системи;

- розміри і складність операційної системи.

Більшість операційних систем мають дефекти з погляду забезпечення безпеки даних у системі, що обумовлено виконанням завдання забезпечення максимальної доступності системи для користувача.

Розглянемо типові функціональні дефекти операційної системи, які можуть призвести до створення каналів витоку даних.

1. Ідентифікація. Кожному ресурсу в системі має бути присвоєно унікальне ім'я - ідентифікатор. У багатьох системах користувачі не мають можливості упевнитися в тому, що використовувані ними ресурси дійсно належать системі.

2. Паролі. Більшість користувачів вибирають найпростіші паролі, які легко підібрати або вгадати.

3. Список паролів. Зберігання списку паролів у незашифрованому вигляді дає можливість його компрометації з подальшим несанкціонованим доступом до даних.

4. Граничні значення. Для запобігання спроб несанкціонованого входу в систему за допомогою підбору пароля необхідно обмежити число таких спроб, що в деяких операційних системах не передбачено.

5. Довіра. У багатьох випадках програми операційної системи вважають, що інші програми працюють правильно.

6. Спільна пам'ять. При використанні спільної пам'яті не завжди після виконання програм очищаються ділянки оперативної пам'яті (ОП).

7. Розрив зв'язку. У разі розриву зв'язку операційна система повинна негайно закінчити сеанс роботи з користувачем або повторно встановити справжність суб'єкта.

8. Передача параметрів за посиланням, а не за значенням (при передачі параметрів за посиланням можливе збереження параметрів у зовнішній пам'яті після перевірки їх коректності, порушник може змінити ці дані до їх використання).

9. Система може містити чимало елементів (наприклад, програм), що мають різні привілеї. Основною проблемою забезпечення безпеки операційної системи є проблема створення механізмів контролю доступу до ресурсів системи. Процедура контролю доступу полягає у перевірці відповідності запиту суб'єкта наданим йому правам доступу до ресурсів. Крім того, операційна система містить допоміжні засоби захисту, такі як засоби моніторингу, профілактичного контролю та аудиту. У сукупності механізми контролю доступу та допоміжні засоби захисту утворюють механізми управління доступом.

2. Основні захисні механізми ОС ряду UNIX.

Захист ОС ряду Unix у загальному випадку базується на трьох основних механізмах:

- 1) ідентифікації та аутентифікація користувача при вході в систему;
- 2) розмежуванні прав доступу до файлової системи, в основі якої знаходиться реалізація дискреційної моделі доступу;
- 3) аудит, тобто реєстрація подій.

При цьому відзначимо, що для різних клонів ОС ряду Unix можливості механізмів захисту можуть незначно відрізнятися, проте будемо розглядати ОС Unix у загальному випадку, без урахування деяких незначних особливостей окремих ОС цього ряду.

Побудова файлової системи і розмежування доступу до файлових об'єктів має особливості, притаманні даному ряду ОС. Розглянемо коротко ці особливості. Всі дискові накопичувачі (томи) об'єднуються в єдину віртуальну файлову систему шляхом операції монтування тому. При цьому вміст тому проектується на вибраний каталог файлової системи. Елементами файлової системи є також всі пристрої, що вмикаються до комп'ютера, який захищається (монтовані до файлової системи). Тому розмежування доступу до них здійснюється через файлову систему.

Кожний файловий об'єкт має індексний дескриптор, в якому серед іншого зберігається інформація про розмежування доступу до даного файлового об'єкту. Права доступу діляться на три категорії: доступ для власника, доступ для групи і доступ для інших користувачів. У кожній категорії визначаються права на читання, запис і виконання (у випадку каталогу - перегляд).

Користувач має унікальний символічний ідентифікатор (ім'я) і числовий ідентифікатор (UID). Символьний ідентифікатор пред'являється користувачем при вході в систему, числовий використовується операційною системою для визначення прав користувача у системі (доступ до файлів тощо).

Принципові *недоліки* захисних механізмів ОС ряду Unix. Розглянемо в загальному випадку недоліки реалізації системи захисту ОС ряду Unix у частині невиконання вимог до захисту конфіденційної інформації, безпосередньо пов'язаних із можливістю несанкціонованого доступу до інформації.

Для початку зазначимо, що в ОС ряду Unix внаслідок реалізованої нею концепції адміністрування (не централізованої) неможливо забезпечити замкнутість (або цілісність) програмного середовища. Це пов'язано з неможливістю встановлення атрибуту "виконання" на каталог (для каталогу даний атрибут обмежує можливість "огляду" вмісту каталогу). Тому при розмежуванні адміністратором доступу користувачів до каталогів користувач, як "власник" створюваного ним файлу, може занести у свій каталог виконуваний файл і, як його "власник", встановити на файл атрибут "виконання", після чого завантажити записану ним програму. Ця проблема безпосередньо пов'язана з реалізованою в ОС концепцією захисту інформації.

Не в повному обсязі реалізується дискреційна модель доступу, зокрема,

не можуть розмежовуватися права доступу для користувача "root" (UID=0), тобто даний суб'єкт доступу виключається зі схеми управління доступом до ресурсів. Відповідно всі процеси, які запускаються, мають необмежений доступ до ресурсів, що захищаються. З цим недоліком системи захисту пов'язана множина атак, зокрема:

- несанкціоноване одержання прав root;
- запуск із правами root власного виконуваного файлу (локально чи віддалено впровадженого), при цьому несанкціонована програма одержує повний доступ до ресурсів, що захищаються, тощо.

Крім того, в ОС ряду Unix неможливо вбудованими засобами гарантовано видаляти залишкову інформацію. Для цього в системі абсолютно відсутні відповідні механізми.

Необхідно також відзначити, що більшість ОС даного ряду не мають можливості контролю цілісності файлової системи, тобто не містять відповідних вбудованих засобів. У кращому випадку додатковими утилітами може бути реалізований контроль конфігураційних файлів ОС за розкладом у той час, як найважливішою можливістю даного механізму можна вважати контроль цілісності програм (застосувань) перед їх запуском, контроль файлів даних користувача і т.д.

Що стосується реєстрації (аудиту), то в ОС ряду Unix не забезпечується реєстрація видачі документів на "тверду копію", а також деякі інші вимоги до реєстрації подій.

Якщо ж трактувати вимоги до управління доступом у загальному випадку, то при захисті комп'ютера в складі ЛОМ необхідне управління доступом до вузлів мережі. Проте вбудованими засобами в деяких ОС ряду Unix управління доступом до вузлів не реалізується.

З наведеного аналізу видно, що чимало механізмів, необхідних із точки зору виконання формалізованих вимог, більшістю ОС ряду Unix не реалізується в принципі, або реалізується лише частково.

3. Основні захисні механізми ОС ряду WINDOWS

Тепер коротко зупинимось на основних механізмах захисту, реалізованих в ОС ряду Windows, і проведемо аналіз захищеності цих операційних систем. Відзначимо, що тут ряд об'єктів доступу (зокрема, пристрої, реєстр ОС тощо) не є об'єктами файлової системи. Тому виникає питання, як варто трактувати таку вимогу "Система захисту повинна контролювати доступ іменованих суб'єктів (користувачів) до іменованих об'єктів (файлів, програмам, томів тощо)". Не зрозуміло, чи є об'єктами доступу, до яких, слідуючи формальним вимогам, необхідно розмежовувати доступ користувачів, наприклад, реєстр ОС і т.д.

На відміну від ряду ОС Unix, де всі завдання розмежувальної політики доступу до ресурсів вирішуються засобами управління доступом до об'єктів файлової системи, доступ у даних ОС розмежовується власним механізмом для кожного ресурсу. Іншими словами, при розгляді механізмів захисту ОС Windows постає завдання визначення та задання вимог до повноти

розмежувань (це визначається тим, що вважати об'єктом доступу).

Так, як і для ряду ОС Unix, тут основними механізмами захисту є:

- 1) ідентифікація та аутентифікація користувача при вході в систему;
- 2) розмежування прав доступу до ресурсів, в основі якого знаходиться реалізація дискреційної моделі доступу (окремо до об'єктів файлової системи, до пристроїв, до реєстру ОС, до принтерів та ін.);
- 3) аудит, тобто реєстрація подій.

Тут явно виділяються (в кращу сторону) можливості розмежувань прав доступу до файлових об'єктів (для NTFS) - істотно розширені атрибути доступу, що встановлюються на різні ієрархічні об'єкти файлової системи (логічні диски, каталоги, файли). Зокрема, атрибут "виконання" може встановлюватися і на каталог, тоді він успадковується відповідними файлами.

При цьому істотно обмежені можливості управління доступом до інших ресурсів, що захищаються, зокрема, до пристроїв введення. Наприклад, тут відсутній атрибут "виконання", тобто неможливо заборонити запуск несанкціонованої програми з пристроїв введення.

Принципові *недоліки* захисних механізмів ОС ряду Windows. Насамперед розглянемо принципові недоліки захисту ОС ряду Windows, безпосередньо пов'язані з можливістю несанкціонованого доступу до інформації.

При цьому на відміну від ОС ряду Unix в ОС Windows неможлива в загальному випадку реалізація централізованої схеми адміністрування механізмів захисту або відповідних формалізованих вимог. Згадаймо, що в ОС Unix це поширювалося лише на завантаження процесів. Пов'язано це з тим, що в ОС Windows прийнята інша концепція реалізації розмежувальної політики доступу до ресурсів (для NTFS).

В рамках цієї концепції розмежування для файлу пріоритетніше, ніж для каталогу, а в загальному випадку - розмежування для файлового об'єкта, який включається, пріоритетніше, ніж для об'єкта, який включає. Це призводить до того, що користувач, створюючи файл і будучи його "власником", може призначити будь-які атрибути доступу до такого файлу (тобто дозволити до нього доступ будь-якому іншому користувачеві). Звернутися до цього файлу може користувач (якому призначив права доступу "власник") незалежно від встановлених адміністратором атрибутів доступу на каталог, в якому користувач створює файл. Ця проблема безпосередньо пов'язана з реалізованою в ОС Windows концепцією захисту інформації.

Далі, в ОС ряду Windows не в повному обсязі реалізується дискреційна модель доступу, зокрема, не можуть розмежовуватися права доступу для користувача "Система". В ОС присутні не тільки користувацькі, а й системні процеси, які завантажуються безпосередньо системою. При цьому доступ системних процесів не може бути розмежований. Відповідно, всі завантажені системні процеси мають необмежений доступ до ресурсів, які захищаються. З цим недоліком системи захисту пов'язана множина атак, зокрема, несанкціоноване завантаження власного процесу з правами системного. До

речі, це можливо і внаслідок некоректної реалізації механізму забезпечення замкнутості програмного середовища.

В ОС ряду Windows неможливо в загальному випадку забезпечити замкнутість (або цілісність) програмного середовища. Це пов'язано з іншими проблемами, ніж в ОС ряду Unix, в яких неможливо встановити атрибут "виконання" на каталог. Для з'ясування складності даного питання розглянемо два способи, якими в загальному випадку можна реалізувати даний механізм, причому обидва способи неспроможні. Отже, механізм замкнутості програмного середовища в загальному випадку може бути забезпечений:

- заданням списку дозволених до запуску процесів із наданням можливості користувачам запускати процеси тільки з цього списку. При цьому процеси задаються повношляховими іменами, причому засобами розмежування доступу забезпечується неможливість їх модернізації користувачем. Такий підхід не реалізується вбудованими в ОС механізмами;

- дозволом запуску користувачами програм тільки із заданих каталогів при неможливості модернізації цих каталогів. Однією з умов коректної реалізації даного підходу є заборона користувачам запуску програм інакше, ніж із відповідних каталогів. Некоректність реалізації ОС Windows даного підходу пов'язана з неможливістю встановлення атрибуту "виконання" на пристрої введення (дисковод або CD-ROM). У зв'язку з цим при розмежуванні доступу користувач може завантажити несанкціоновану програму з дискети, або з диска CD-ROM (досить поширена атака на ОС даного ряду).

Тут же варто відзначити, що з точки зору забезпечення замкнутості програмного середовища [тобто реалізації механізму, який забезпечує можливість користувачам запускати тільки санкціоновані процеси (програми)] дії користувача із запуску процесу можуть бути як явними, так і прихованими.

Явні дії передбачають запуск процесів (виконуваних файлів), які однозначно ідентифікуються своїм іменем. Приховані дії дозволяють здійснювати вбудовані в застосування інтерпретатори команд. Прикладом таких є офісні застосування. При цьому прихованою діє користувача буде запуск макросу.

У даному випадку ідентифікації підлягає лише власне застосування, наприклад, процес winword.exe. При цьому процес може крім своїх регламентованих дій виконувати ті приховані дії, які задаються макросом (відповідно, ті, які допускаються інтерпретатором), що зберігається у відкритому документі. Те ж стосується і будь-якої віртуальної машини, що містить вбудований інтерпретатор команд. При цьому відзначимо, що при використанні застосувань, які мають вбудовані інтерпретатори команд (у тому числі офісних застосувань), не в повному обсязі забезпечується виконання вимоги щодо ідентифікації програм.

Повертаючись до обговорення недоліків, відзначимо, що в ОС ряду Windows неможливо вбудованими засобами гарантовано видаляти залишкову

інформацію. В системі відсутні відповідні механізми.

Крім того, ОС ряду Windows не має в повному обсязі можливість контролю цілісності файлової системи. Вбудовані механізми системи дозволяють контролювати лише власні системні файли, не забезпечуючи контроль цілісності файлів користувача. Крім того, вони не вирішують найважливіше завдання даних механізмів - контроль цілісності програм (застосовань) перед їх запуском, контроль файлів даних користувача та ін.

Що стосується реєстрації (аудиту), то в ОС ряду Windows не забезпечується реєстрація видачі документів на "тверду копію", а також деякі інші вимоги до реєстрації подій. Знову ж таки, якщо трактувати вимоги до управління доступом у загальному випадку, то при захисті комп'ютера в складі ЛОМ необхідне управління доступом до вузлів мережі (розподілений пакетний фільтр). В ОС ряду Windows механізм управління доступу до вузлів у повному обсязі не реалізується.

Що стосується мережевих ресурсів, то фільтрації піддається тільки доступ до вхідного ресурсу, а запит доступу на комп'ютері, з якого він здійснюється, фільтрації не підлягає. Це принципово, оскільки не можуть підлягати фільтрації застосування, якими користувач здійснює доступ до ресурсів. Завдяки цьому, досить поширеними є атаки на протокол NETBIOS.

Крім того, в повному обсязі управляти доступом до ресурсів можна тільки при встановленій на всіх комп'ютерах ЛОМ файлової системі NTFS. В іншому випадку неможливо заборонити запуск несанкціонованої програми з віддаленого комп'ютера, тобто забезпечити замкнутість програмного середовища в цій частині.

З наведеного аналізу можна зробити висновок, що чимало механізмів, необхідних із точки зору виконання формалізованих вимог із захисту, ОС ряду Windows не реалізують у принципі, або реалізують лише частково.

З урахуванням сказаного можемо зробити важливий висновок щодо того, що більшістю сучасних універсальних ОС не виконуються в повному обсязі вимоги до захисту автоматизованих систем. Це означає, що вони не можуть без використання додаткових засобів захисту застосовуватися для захисту навіть конфіденційної інформації. При цьому слід зазначити, що основні проблеми захисту тут викликані не неможливістю ОС до вимог окремих механізмів захисту, а принциповими причинами, зумовленими реалізованою в ОС концепцією захисту. Концепція ця ґрунтується на реалізації розподіленої схеми адміністрування механізмів захисту, що само по собі є невиконанням формалізованих вимог до основних механізмів захисту.

4. Аналіз існуючих статистик загроз для сучасних універсальних ОС

На сьогоднішній день існує досить велика статистика загроз ОС, спрямованих на подолання вбудованих в ОС механізмів захисту, які дозволяють змінити налаштування механізмів безпеки, обійти розмежування

доступу і т.д.

Таким чином, статистика фактів несанкціонованого доступу до інформації показує, що більшість поширених систем (універсального призначення) досить вразливі з точки зору безпеки. І це незважаючи на виразну тенденцію до підвищення рівня захищеності цих систем.

Тут необхідно зазначити, що на практиці сучасні інформаційні системи, призначені для обробки конфіденційної інформації, будуються вже з урахуванням додаткових заходів безпеки, що також побічно підтверджує початкову уразливість сучасних ОС.

Розглянемо операційні системи, які фігурують у опублікованих списках системних і прикладних помилок, які дозволяють одержати несанкціонований доступ до системи, знизити міру її захищеності або добитися відмови в обслуговуванні (системного збою).

Внаслідок того, що більшість атак для операційних систем, побудованих на базі Unix, досить схожі, доцільно об'єднати їх в одну групу. Теж саме можна сказати і про ОС ряду Windows. Таким чином, далі будемо розглядати тільки ряди ОС: Unix, MS Windows, Novell NetWare.

Щодо ОС Novell варто зауважити, що дана ОС спочатку створювалася як захищена (не універсального призначення) ОС, основною функцією якої був захищений файловий сервіс. Це, з одного боку, повинно було забезпечити їй більш високий рівень захищеності, з іншого боку, накладало певні обмеження щодо використання.

Проте, починаючи з п'ятої версії, дана ОС почала набувати властивості універсальності (з точки зору застосовуваних протоколів і застосувань), що якоюсь мірою позначилося і на рівні її захищеності.

5.Огляд і статистика методів, що знаходяться в основі атак на сучасні ОС.

Аналізуючи розглянуті атаки, всі методи, які дозволяють не санкціоновано втрутитися в роботу системи, можна розділити на такі групи:

- 1) які дозволяють не санкціоновано завантажити виконуваний код;
- 2) які дозволяють здійснити несанкціоновані операції читання/запису файлових чи інших об'єктів;
- 3) які дозволяють обійти встановлені розмежування прав доступу;
- 4) які призводять до відмови (Denial of Service) в обслуговуванні (системний збій);
- 5) які використовують вбудовані не документовані можливості (помилки і закладки);
- 6) які використовують недоліки системи зберігання або вибору (недостатня довжина) даних про аутентифікації (паролі) і дозволяють шляхом реверсування, підбору або повного перебору всіх варіантів одержати ці дані;
- 7) троянські програми;
- 8) інші.

Аналізуючи відому статистику загроз, можна зробити висновок, що більша їх частина пов'язана саме з недоліками засобів захисту ОС, тобто недоліками, пов'язаними з невиконанням (повним, чи частковим) формалізованих вимог до захисту, серед яких, в першу чергу, можуть бути виділені:

1) некоректна реалізація механізму управління доступом, перш за все, при розмежуванні доступу до захищених об'єктів системних процесів і користувачів, що мають права адміністратора;

2) відсутність забезпечення замкнутості (цілісності) програмного середовища.

Як видно, більшість атак здійснювалося або з використанням деяких прикладних програм, або із застосуванням вбудованих у віртуальні машини засобів програмування, тобто можливість більшості атак безпосередньо пов'язана з можливістю запуску зловмисником відповідної програми. При цьому запуск може бути здійснений як явно, так і приховано, в рамках можливостей вбудованих у застосування інтерпретаторів команд.

Проведений аналіз відомих загроз сучасним універсальним ОС повністю підтверджує, що більша їх частина обумовлена саме реалізованим в ОС концептуальним підходом, що складається в реалізації схеми розподіленого адміністрування механізмів захисту. В рамках цієї схеми користувач розглядається як довірена особа, яка є елементом схеми адміністрування і має можливість призначати/змінювати правила розмежування доступу. При цьому він не сприймається як потенційний зловмисник, який може свідомо чи несвідомо здійснити НСД до інформації, отже призначення механізмів додаткового захисту ОС полягає в реалізації централізованої схеми адміністрування механізмів захисту, в рамках якої буде здійснюватися протидія НСД користувача до інформації