ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

УДК 004.891

к.т.н., доц. **Джулій В.М.** (ХмНУ)
к.т.н., доц. **Бойчук В.О.** (ХмНУ)
к.т.н., доц. **Тітова В.Ю.** (ХмНУ)
д.т.н., с.н.с. **Сєлюков О.В.** («Укрспецконсалтинг»)
к.т.н., с.н.с. **Мірошніченко О.В.** (ВІКНУ)

DOI: https://doi.org/10.17721/2519-481X/2020/67-08

МОДЕЛІ І МЕТОДИ ЗАХИСТУ ВІД ЗАГРОЗЛИВИХ ПРОГРАМ ІНФОРМАЦІЙНИХ СИСТЕМ

У статті запропоновано підхід до розвитку методів захисту від загрозливих програм в сучасних інформаційних системах, що складається в розробці методів захисту, заснованих на реалізації контролю доступу до файлів по їх типам, які можуть бути ідентифіковані розширеннями файлів, що істотно перевершують відомі методи антивірусного захисту, як по ефективності захисту, так і в міру впливу на завантаження обчислювальних ресурсів інформаційної системи.

Показано, що найбільш актуальними для захисту є виконувані бінарні і скриптові файли і про те, що дані класи шкідливих програм передбачають обов'язкове збереження загрозливого файлу на жорсткому диску перед його виконанням (читанням). Це дозволило зробити висновок щодо того, що захист від загрозливих програм може будуватися реалізацією контролю (розмежування прав) доступу до файлів.

Запропоновано загальний підхід до реалізації захисту від загрозливих програм, заснований на реалізації контролю доступу до файлів по їх типам, які можуть бути ідентифіковані розширеннями файлів. Можливість використання подібного підходу обґрунтовано проведеним дослідженням засобів захисту.

Методи захисту від загрозливих програм дозволяють захистити інформаційну систему, як від завантаження, так і від виконання бінарних і скриптових загрозливих файлів, що відрізняються можливістю врахування розташування виконуваних файлів, можливістю адміністрування при працюючій системі захисту, можливістю контролю модифікації об'єктів доступу, перейменування об'єктів доступу, можливістю захисту від скриптових загрозливих програм, в тому числі з урахуванням можливості наділення загрозливими властивостями інтерпретаторів (віртуальних машин).

Розроблено моделі контролю доступу, що дозволили на побудованих матрицях доступу сформулювати вимоги до побудови безпечної системи, виконання яких запобігає витоку заданих прав доступу суб'єктів до об'єктів.

Ключові слова: загрозливі програми, комп'ютерна безпека, інформаційна система, контроль доступу, об'єкт доступу, суб'єкт доступу.

Вступ. Однією з ключових сучасних проблем забезпечення комп'ютерної безпеки є необхідність ефективної протидії загрозливим програмам. При цьому необхідно враховувати, що це можуть бути, як самостійні програми, покликані здійснювати відповідні несанкціоновані дії, так і цілком легальні, санкціоновано використовувані додатки, що наділяються в процесі роботи загрозливими властивостями. На сьогоднішній день для вирішення розглянутих задач використовуються різноманітні способи сигнатурного і поведінкового аналізів, покликаних запобігти можливості несанкціонованого впровадження та запуску загрозливих програм на ресурси, що захищаються. Однак існуюча статистика зростання загрозливих програм дозволяє припустити про низьку ефективність методів вирішення найбільш актуальних сучасних завдань захисту інформації. Більшість сучасних

методів захисту від загрозливих програм зводиться до виявлення таких, або до спроби запобігання атаки з боку присутньої на комп'ютері загрозливої програми.

Технологія виявлення загрозливих програм була заснована на використанні сигнатур - ділянок коду, однозначно ідентифікують ту чи іншу загрозливу програму. У міру того, як еволюціонували загрозливі програми, ускладнювалися і розвивалися технології їх детектування. Умовно роботу антивірусних програм можна розділити на збір даних для виявлення загрозливих програм і якийсь механізм, що визначає на основі зібраної інформації загрозлива ця програма чи ні. Узагальнюючи, можна виділити наступні способи: робота з файлом як з масивом байтів; емуляція коду програми; запуск програми в «пісочниці» (sandbox2) (а також використання близьких за змістом технологій віртуалізації); моніторинг системних подій; пошук системних аномалій.

Сучасні антивірусні засоби захисту, як правило, одночасно реалізують і технічний, і аналітичний компоненти, засновані на сукупності методів захисту. При цьому, незважаючи на те, що сигнатурний метод застарів, він на сьогодні залишається в захисних продуктах і рекламується, як частина стійкої бізнес моделі. При цьому не варто забувати, що створення сигнатур - ручний процес, тому розробка нових сигнатур і випуск оновлень для користувача може займати від декількох годин до доби. Таким чином, сигнатурні механізми не рятують від нових, що не потрапили в базу, погроз. У зв'язку з ручним складанням сигнатур вони дають помилкові спрацьовування. База сигнатур зростає, і, в зв'язку з цим, робота з сигнатурами накладає все більші витрати на ресурси комп'ютера.

Як показує практика, евристичний метод справляється лише в невеликій кількості випадків, тому цей підхід також не вирішує завдання захисту від запуску загрозливих програм. При сучасних методах приховування і шифрування виконуваних файлів виявити нові загрози такими методами практично неможливо.

Постановка задачі. На підставі проведених досліджень можемо зробити наступний важливий висновок - всі дослідження ефективності антивірусних засобів захисту зводяться до оцінювання ефективності детектування загрозливих програм, причому на деяких відомих кінцевих наборах, що вже ставить під сумнів коректність подібних оцінок. Разом з тим, систему створення і виявлення загрозливих програм можна розглядати, як стохастичну систему, яка характеризується відповідними інтенсивностями і можливостями. Саме застосування такого підходу дозволить виявити і кількісно коректно оцінити основні характеристики системи антивірусного захисту - системи захисту від загрозливих програм. Виходячи з вище викладеного, в рамках проведеного дослідження ставиться задача моделювання системи антивірусного захисту з використанням математичного апарату теорії масового обслуговування, визначення і розрахунку основних характеристик з подальшою оцінкою реальної ефективності відомих методів захисту від загрозливих програм. Інший висновок, зроблений в результаті проведених досліджень, полягає в тому, що завдання захисту від загрозливих програм апріорі розглядається експертами з безпеки в якості актуальної. В рамках проведеного дослідження ставиться задача розробки математичної моделі, що дозволяє кількісно оцінювати актуальність загрози в інформаційній системі, що може служити обґрунтуванням рішення задачі захисту від тієї чи іншої загрози.

Проведене дослідження існуючих підходів до оцінки ефективності методів і засобів захисту від загрозливих програм, в результаті якого зроблені висновки про неможливість з використанням відомих підходів ні кількісно оцінити актуальність окремої загрози для інформаційної системи в цілому (з урахуванням безлічі інших потенційних загроз), в тому числі загрози занесення і запуску загрозливих програм, ні кількісно оцінити основні стохастичні характеристики безпеки системи від загрозливих програм. В результаті чого сформульована задача розробки відповідних математичних моделей і наступного проведення на них досліджень, що дозволять отримати необхідні кількісні оцінки.

Основна частина. Розглянемо ефективність антивірусного програмного забезпечення (ПЗ), що використовує бази сигнатур, евристик або поведінки загрозливих програм.

Представимо виявлення нової загрозливої програми як систему масового обслуговування, де прилад - аналітик, заявка - нова загрозлива програма. В результаті отримуємо багатоканальну СМО з необмеженою чергою M/M/C. Математична модель для розрахунку ефективності, яка може будуватися в залежності від розв'язуваних завдань:

$$p_0 = f(\lambda, \mu), \quad (1)$$

де μ - інтенсивність випуску нових сигнатур і евристик; λ - інтенсивність виявлення нових загрозливих програм. Дані величини задаються виходячи з існуючих статистик. Розглянемо стаціонарну роботу системи:

$$\frac{\rho}{C} < 1, \quad (2)$$

де ρ - інтенсивність навантаження, що дорівнює відношенню інтенсивності надходження заявок до інтенсивності обслуговування:

$$\rho = \frac{\lambda}{\mu}, \quad (3)$$

C - кількість обслуговуючих приладів. Із формул (2) і (3), отримуємо нерівність:

$$\frac{\lambda}{\mu \cdot C} < 1. \quad (4)$$

Інтенсивність обслуговування і кількість приладів ми задаємо виходячи із загальних відомостей про роботу антивірусних компаній. Знаючи ці параметри, з формули (4) отримуємо інтервал, яким обмежується інтенсивність надходження заявок в умовах стаціонарної роботи системи:

$$\lambda < \mu \cdot C. \quad (5)$$

Для початку розрахуємо ймовірність (p_0) того, що всі аналітики виявляться вільними і в системі не виявиться жодної заявки. Імовірність p_0 - характеристика актуальності загрози, чим ймовірність менше, тим загроза актуальніша.

Імовірність p_0 того, що система готова до експлуатації в будь-який момент часу і відсутні не виявлені сигнатури, розраховується за формулою:

$$p_0 = \left(\sum_{n=0}^{C} \frac{\left(\lambda/\mu\right)^n}{n!} + \frac{\left(\lambda/\mu\right)^{C+1}}{C!\left(C - \frac{\lambda}{\mu}\right)} \right)^{-1}. \quad (6)$$

Розрахуємо, яка буде довжина черги виходячи з статистичної інтенсивності надходження заявок. Розглянемо випадок стаціонарної роботи системи:

$$\frac{\rho}{C} < 1. \quad (7)$$

З формули (7) і (3), отримуємо:

$$\mu > \frac{\lambda}{C}. \quad (8)$$

Виходячи, що $\mu = 1/ T_{обс}$, отримуємо з формули (8) обмеження на середній час обслуговування заявки:

$$T_{обс} < \frac{C}{\lambda}. \quad (9)$$

Виходячи з формули (5), будемо задавати $T_{обс}$ і розрахуємо середню довжину черги:

$$L_{дч} < \frac{\rho^{C+1} p_0}{C \cdot C!\left(1 - \rho/C\right)^2}. \quad (10)$$

Середній час обслуговування заявки розраховується за формулою:

$$T_{обс} = \frac{L_{дч}}{\lambda} . \quad (11)$$

Припускаючи, що система працює в стаціонарному режимі, задаючи інтенсивність надходження заявок і кількість аналітиків, отримали систему близьку до нестаціонарної, так як в крайніх точках черга різко зростає. Це свідчить про низьку ефективність існуючих антивірусних програм і про відсутність необхідного рівня захисту.

На підставі введеної класифікації загрозливих програм, зроблено висновок про потенційну можливість реалізації захисту від загрозливих програм на основі контролю (розмежування) доступу до ресурсів, зокрема, до файлових об'єктів. Пропонується будувати захист від загрозливих програм на основі дискреційної моделі розмежування доступу до об'єктів файлової системи. Дискреційний принцип управління доступом будемо розглядати як примусове управління потоком інформації, виключаючи власника об'єкта доступу. Вихідною вимогою для заданої моделі є те, що ніякий користувач не зможе вивести систему з безпечного стану. В рамках цієї моделі система обробки інформації представляється у вигляді сукупності активних сутностей - суб'єктів (множини *S*), які здійснюють доступ до інформації, пасивних сутностей - об'єктів (множина *O*), що містять інформацію яка захищається, і кінцевої множини прав доступу $R-\{r_1, \dots, r_n\}$, що означають повноваження на виконання відповідних дій (наприклад, читання, запис, виконання). Рядки матриці відповідають суб'єктам, а стовпці - об'єктам. Будь-яка комірка матриці $M[S,O]$ містить набір прав суб'єкта *S* до об'єкта *O*, що належать множині прав доступу *R*. Множина режимів доступу залежить від типу аналізуємих об'єктів. До режимів доступу відносяться: читання, запис, виконання.

Формальний опис моделі складається з наступних елементів: кінцевий набір вихідних суб'єктів $S=\{S_1, \dots, S_n\}$; кінцевий набір вихідних об'єктів $O=\{O_1, \dots, O_n\}$; кінцевий набір прав доступу $R-\{r_1, \dots, r_n\}$ вихідна матриця доступу *M*, що містить права доступу суб'єктів до об'єктів.

Для аналізу безпеки системи захисту, що реалізує дискреціонну політику безпеки, будемо використовувати модель Харрісона - Руззо - Ульмана. Для заданої моделі початковий стан $Q_0=(S_0,O_0,M_0)$ є безпечним щодо права *r*, якщо не існує застосовної до Q_0 послідовності команд, в результаті якої право *r* буде занесено в комірку матриці *M*, в якій воно було відсутнє в стані Q_0. Іншими словами це означає, що суб'єкт ніколи не отримає право доступу *r* до об'єкта, якщо він не мав його на початку. Якщо ж право *r* виявилося в комірці матриці *M*, в якій воно на початку було відсутнє, то стався витік критичного права *r* по об'єкту доступу.

Для запобігання несанкціонованого завантаження і створення загрозливої програми необхідно заборонити модифікацію, видалення і створення подібних існуючих файлів, які підпадають під зазначені розширення. Для скорочення списку контрольованих об'єктів доступу слід дозволяти виконувати (бінарні файли) і читати (скриптові) тільки обмежений набір існуючих об'єктів, які теж задаються виходячи з їх розширень. В результаті основна суть пропонованого підходу захисту від загрозливого ПЗ полягає в тому, що: об'єкти доступу визначаються по їх розширенням (виконувані, системні або інформаційні); виключається будь-яка можливість несанкціонованої модифікації заданих об'єктів; виключається будь-яка можливість несанкціонованого видалення заданих об'єктів; виключається будь-яка можливість створення заданих об'єктів; тільки задані об'єкти доступу дозволяється виконувати.

Згідно класифікації загрозливих програм (ЗП), вони поділяються на бінарні і скриптові виконані файли і макро-програми. Перший і другий типи ЗП є файлами з певними розширеннями, відповідно можуть виступати в якості об'єктів доступу в пропонованому

підході захисту. Розглянемо варіант роботи в інформаційній системі, коли у всіх користувачів однакові права доступу, включаючи системних користувачів і адміністратора. В якості суб'єктів доступу виступатимуть користувачі системи $S_i : S = \{S_1, \ldots, S_k\}$. В якості суб'єкта доступу може бути будь-який користувач, у тому числі з правами системи - користувач System. Об'єкти доступу діляться на виконувані, системні та інформаційні: $O = \{ O_{вик1}, \ldots, O_{викq}, O_{сист1}, \ldots, O_{систm}, O_{інф1}, \ldots, O_{інфn} \}$. Це дозволяє захищати як від вже відомих загрозливих програм, так і від нових. Кінцевий набір прав доступу включає читання, виконання і запис: $R = \{Чт, В, Зп\}$.

Права доступу призначаються суб'єктам доступу, а не як атрибути об'єктів доступу. Використовується дозвільна політика доступу, тобто все що явно не дозволено, то заборонено. Згідно з основною ідеєю пропонованого підходу захисту від загрозливого ПЗ, пропонований метод полягає в наступному: для виконуваних об'єктів доступу (ОД) дозволяється читання і виконання; для системних ОД дозволяється тільки читання; для інформаційних ОД дозволяється читання і запис; все інше забороняється.

Виходячи із заданих параметрів, отримуємо наступну матрицю доступу (МД):

$$M = \begin{array}{c} \\ S_1 \\ \ldots \\ \ldots \\ S_k \end{array} \begin{array}{c} \begin{array}{ccc} O_{вик1}, \ldots, O_{викq} & O_{сист1}, \ldots, O_{систm} & O_{інф1}, \ldots, O_{інфn} \end{array} \\ \begin{bmatrix} Чт, В & Чт & Чт, Зп \\ & \ldots & \\ & \ldots & \\ Чт, В & Чт & Чт, Зп \end{bmatrix} \end{array}.$$

У даній МД критичний витік права доступу «Запис» і «Виконання». Оскільки послідовність дій, в результаті яких відбудеться витік права, генерує суб'єкт доступу (користувач), то його дії не зможуть привести до витоку права доступу. В тому числі при даній політиці виключена сутність «Власник», відповідно користувач, який створив новий об'єкт, не зможе наділити правом доступу «Запис» інших користувачів. При створенні нового СД також неможливий витік права доступу, так як розмежування застосовні до будь-якого користувачева і не важливі ні його ім'я, ні його SID.

Перевагою даного методу є те, що неможливо зберегти або запустити загрозливе ПЗ.

Недоліком даних розмежувань є той факт, що дозволено виконувати всі виконувані файли, незалежно від їх розташування. У даній ситуації не можемо розмежовувати доступ до тих ресурсів, зміну яких ми не можемо проконтролювати. Другим недоліком є те, що Адміністратор системи не зможе встановлювати нове програмне забезпечення або оновлювати існуюче.

Для виключення недоліку, пов'язаного з неможливістю розмежовувати доступ до тих ресурсів, зміну яких ми не можемо проконтролювати, введемо додаткову вимогу для розташування виконуваних файлів. Можливі два способи обмеження: системні каталоги «%ProgramFiles%» і «%windir%»; системний диск, наприклад, «C:\». В якості суб'єктів доступу виступатимуть користувачі системи $S_i : S = \{S_1, \ldots, S_k\}$. ОД залишаються незмінними: $O = \{ O_{вик1}, \ldots, O_{викq}, O_{сист1}, \ldots, O_{систm}, O_{інф1}, \ldots, O_{інфn} \}$. Кінцевий набір прав доступу включає в себе читання, виконання і запис: $R = \{Чт, В, Зп\}$. Введемо зміни для завдання об'єкта доступу: дозволити читання і виконання тільки тих виконуваних ОД, які вже знаходяться на системному диску; дозволити тільки читання для системних ОД; дозволити читання і запис для інформаційних ОД; все інше забороняється.

Виходячи із заданих параметрів, отримуємо наступну матрицю доступу:

$$M=\begin{array}{c} \\ S_1 \\ \dots \\ \dots \\ S_k \end{array}\begin{array}{c} \begin{array}{ccc} O_{вик1}, \dots, O_{викq} & O_{сист1}, \dots, O_{систm} & O_{інф1}, \dots, O_{інфn} \end{array} \\ \begin{bmatrix} Чт, В & Чт & Чт, Зп \\ \dots & \dots & \\ \dots & \dots & \\ Чт, В & Чт & Чт, Зп \end{bmatrix} \end{array}.$$

Перевагою даного методу є контроль всіх виконуваних файлів, в тому числі на зовнішніх пристроях і в мережевих каталогах. Недоліком залишається той факт, що Адміністратор системи не зможе встановлювати нове програмне забезпечення або оновлювати існуюче.

Для усунення недоліку, пов'язаного з відсутністю можливості установки нового програмного забезпечення або відновлення існуючого розглянемо варіант, коли Адміністратору інформаційної системи дозволено додатково встановлювати програмне забезпечення. Вводимо додатковий суб'єкт доступу Адміністратор і отримуємо $S_i : S = \{S_1, \dots, S_k, A\}$. Об'єкти доступу залишаються незмінними: $O = \{ O_{вик1}, \dots, O_{викq}, O_{сист1}, \dots, O_{систm}, O_{інф1}, \dots, O_{інфn} \}$. Кінцевий набір прав доступу включає в себе читання, виконання і запис: $R = \{Чт, В, Зп\}$.

Даний метод захисту полягає в наступному:

- для суб'єкта доступу Адміністратор: дозволити читання, запис (установка) і виконання виконуваних ОД; дозволити читання, запис (установка) системних ОД; дозволити читання і запис інформаційних ОД; все інше забороняти.

- для всіх інших суб'єктів доступу: дозволити читання і виконання тільки тих виконуваних ОД, які вже знаходяться на системному диску; дозволити читання системних ОД; дозволити читання і запис інформаційних ОД; все інше забороняти.

У даній МД критична витік права доступу «Запис». Оскільки послідовність дій, в результаті яких право «Запис» буде занесено в комірку матриці, що відноситься до виконуваних ОД, генерує суб'єкт доступу (користувач), то його дії не зможуть привести до витоку права доступу.

Виходячи із заданих параметрів, отримуємо наступну матрицю доступу:

$$M=\begin{array}{c} \\ S_1 \\ \dots \\ \cdots \\ S_k \\ A \end{array}\begin{array}{c} \begin{array}{ccc} O_{вик1}, \dots, O_{викq} & O_{сист1}, \dots, O_{систm} & O_{інф1}, \dots, O_{інфn} \end{array} \\ \begin{bmatrix} Чт, В & Чт & Чт, Зп \\ \dots & \dots & \\ \cdots & \dots & \\ Чт, В & Чт & Чт, Зп \\ Чт, Зп, В & Чт, Зп & Чт, Зп \end{bmatrix} \end{array}.$$

У даній ситуації при аналізі змін стану системи не розглядаємо дії адміністратора так як він просто може відключити засіб захисту. В тому числі при даній політиці виключена сутність «Власник», відповідно користувач, який створив новий об'єкт, не зможе наділити правом доступу «Запис» інших користувачів. При створенні нового ОД також неможливий витік права доступу «Запис», так як розмежування застосовні до будь-якого користувача і не важливі ні його ім'я, ні його SID. У разі підвищення привілею звичайного користувача до Адміністратора можливий витік права щодо суб'єкта доступу.

Перевагою даного методу є можливість адміністратора встановлювати нове ПЗ або оновлювати існуюче. Недоліком даного методу є можливість отримання адміністративних прав і внаслідок чого необхідність додаткового захисту від підвищення привілею до адміністративних прав. Дана технологія реалізована для надання можливості окремому потоку

виконуватися від імені іншого користувача, тобто в цьому випадку буде наданий інший маркер доступу.

Для усунення недоліку даного методу і можливості підвищення привілею і для контролю зміни прав доступу необхідно СД задавати виходячи з двох сутностей: ефективного та первинного користувача, і забороняти уособлення будь-якого користувача з Адміністратором.

Одним з варіантів проникнення загрозливого ПЗ є його впровадження під виглядом санкціонованого типу файлів (інформаційного), видалення санкціонованого виконуваного файлу і після цього його перейменування. Для вирішення цієї проблеми пропонується розділяти способи модифікації об'єктів. Даний метод робить акцент на способах модифікації ОД. Відмінними моментами від попереднього методу є: заборона будь-якої модифікації об'єктів (санкціонованих програм), дозволених до запуску шляхом перейменування; заборона будь-якої модифікації об'єктів (санкціонованих програм), дозволених до запуску шляхом видалення виконуваного файлу.

Суб'єкти доступу - $S_i : S = \{S_1, \dots, S_k, A\}$. Об'єкти доступу залишаються незмінними: $O = \{ O_{вик1}, \dots, O_{викq}, O_{сист1}, \dots, O_{системm}, O_{інф1}, \dots, O_{інфn} \}$. Кінцевий набір прав доступу включає в себе читання, виконання і запис: $R = \{Чт, В, Зп\}$. Крім звичайного набору прав доступу, введемо додаткові методи доступу. Заборона запису - «~~Зп~~». Заборона перейменування - «~~П~~». Перейменування виділяємо, щоб уникнути підміну санкціонованого виконуваного файлу. У набір прав доступу додамо заборону видалення - «~~Д~~». В результаті отримаємо наступний набір прав: $R = \{Чт, В, Зп, ~~Зп~~, ~~П~~, ~~Д~~\}$.

Даний метод захисту полягає в наступному:

- для суб'єкта доступу Адміністратор: дозволити читання, запис (установка) і виконання виконуваних ОД; дозволити читання, запис (установка) системних ОД; дозволити читання і запис інформаційних ОД; все інше заборонити.

- для всіх інших суб'єктів доступу: дозволити читання і виконання тільки тих виконуваних ОД, які вже знаходяться на системному диску; заборонити запис, перейменування і видалення існуючих виконуваних ОД на системному диску; дозволити читання для системних ОД; заборонити для системних ОД запис, перейменування і видалення; дозволити для інформаційних ОД читання і запис; заборонити для інформаційних ОД перейменування і видалення; все інше заборонити.

Виходячи із заданих параметрів, отримуємо наступну матрицю доступу:

$$M = \begin{array}{c} \\ S_1 \\ \dots \\ \dots \\ S_k \\ A \end{array} \begin{array}{c} \begin{array}{ccc} O_{вик1}, \dots, O_{викq} & O_{сист1}, \dots, O_{системm} & O_{інф1}, \dots, O_{інфn} \end{array} \\ \begin{bmatrix} Чт, В, ~~Зп~~, ~~П~~, ~~Д~~ & Чт, ~~Зп~~, ~~П~~, ~~Д~~ & Чт, Зп, ~~П~~, ~~Д~~ \\ \dots & \dots & \\ \dots & \dots & \\ Чт, В, ~~Зп~~, ~~П~~, ~~Д~~ & Чт, ~~Зп~~, ~~П~~, ~~Д~~ & Чт, Зп, ~~П~~, ~~Д~~ \\ Чт, Зп, В & Чт, Зп & Чт, Зп \end{bmatrix}. \end{array}$$

У даній МД критичний витік права доступу «Запис». Оскільки послідовність дій, в результаті яких право «Запис» буде занесено в комірку матриці, що відноситься до виконуваних ОД, генерує суб'єкт доступу (користувач), то його дії не зможуть привести до витоку права доступу. У даній ситуації при аналізі змін стану системи не розглядаємо дії адміністратора так як він просто може відключити засіб захисту. В тому числі при даній політиці виключена сутність «Власник», відповідно користувач, який створив новий об'єкт, не зможе наділити правом доступу «Запис» інших користувачів. При створенні нового СД також неможливий витік права доступу «Запис», так як розмежування застосовні до будь-якого користувачева і не важливі ні його ім'я, ні його SID.

Перевагою даного методу є захист від впровадження шкідливої програми під виглядом санкціонованої. Недоліком даного методу є можливість отримання адміністративних прав і внаслідок чого необхідність додаткового захисту від підвищення привілею до адміністративних прав. Відповідно для усунення можливості підвищення привілею і для контролю зміни прав доступу необхідно суб'єкт доступу задавати виходячи з двох сутностей: ефективного та первинного користувача, і забороняти уособлення будь-якого користувача з Адміністратором. Іншим недоліком даного методу є той факт, що не враховується різниця між перейменуванням існуючого виконуваного файлу від перейменування, наприклад, інформаційного файлу, запис якого дозволена, в виконуваний файл.

Для усунення даного недоліку в даному методі додатково додається заборона перейменування існуючого виконуваного файлу (позначається «~~Пв~~»), заборона перейменування в виконуваний файл (позначається «~~ПвВ~~»). Суб'єкти доступу - $S_i : S = \{S_1, \ldots, S_k, A\}$. Об'єкти доступу залишаються незмінними: $O = \{ O_{вик1}, \ldots, O_{викq}, O_{сист1}, \ldots, O_{системт}, O_{інф1}, \ldots, O_{інфn} \}$. Кінцевий набір прав доступу включає в себе читання, виконання і запис, заборона запису, заборона перейменування існуючого виконуваного файлу, заборона перейменування в виконуваний файл, заборона видалення. В результаті отримаємо наступний набір прав: *R* = {*Чт, В, Зп, ~~Зп~~, ~~Пв~~, ~~ПвВ~~, ~~Д~~*}.

Даний метод захисту полягає в наступному:

- для суб'єкта доступу Адміністратор: дозволити читання, запис (установка) і виконання виконуваних ОД; дозволити читання, запис (установка) системних ОД; дозволити читання і запис інформаційних ОД; все інше заборонити.

- для всіх інших суб'єктів доступу: дозволити читання і виконання тільки тих ОД, які вже знаходяться на системному диску; заборонити запис, перейменування існуючого виконуваного ОД, видалення існуючих виконуваних ОД на системному диску; дозволити читання для системних ОД; заборонити для системних ОД запис, перейменування існуючого ОД, видалення; дозволити для інформаційних ОД читання і запис; заборонити для інформаційних ОД перейменування існуючого ОД, видалення; все інше заборонити.

Виходячи із заданих параметрів, отримуємо наступну матрицю доступу:

$$M = \begin{array}{c} \\ S_1 \\ \ldots \\ \ldots \\ S_k \\ A \end{array} \begin{array}{c} \begin{array}{ccc} O_{вик1}, \ldots, O_{викq} & O_{сист1}, \ldots, O_{системт} & O_{інф1}, \ldots, O_{інфn} \end{array} \\ \begin{bmatrix} Чт,В,Зп,\sout{Пв},\sout{ПвВ},Д & Чт,Зп,\sout{Пв},\sout{ПвВ},Д & Чт,Зп,\sout{Пв},\sout{ПвВ},Д \\ & \ldots & \\ & \ldots & \\ Чт,В,Зп,\sout{Пв},\sout{ПвВ},Д & Чт,Зп,\sout{Пв},\sout{ПвВ},Д & Чт,Зп,\sout{Пв},\sout{ПвВ},Д \\ Чт,Зп,\ В & Чт,Зп & Чт,\ Зп \end{bmatrix} \end{array}.$$

У даній МД критичний витік права доступу «Запис». Оскільки послідовність дій, в результаті яких право «Запис» буде занесено в комірку матриці, що відноситься до виконуваних ОД, генерує суб'єкт доступу (користувач), то його дії не зможуть привести до витоку права доступу. В тому числі при даній політиці виключена сутність «Власник», відповідно користувач, який створив новий об'єкт, не зможе наділити правом доступу «Запис» інших користувачів. При створенні нового СД також неможливий витік права доступу «Запис», так як розмежування застосовні до будь-якого користувачева і не важливі ні його ім'я, ні його SID.

Перевагою даного методу є додатковий контроль перейменування об'єктів доступу.

Недоліком даного методу є можливість отримання адміністративних прав і внаслідок чого необхідність додаткового захисту від підвищення привілею до адміністративних прав. Для усунення можливості підвищення привілею і для контролю зміни прав доступу необхідно

суб'єкт доступу задавати виходячи з двох сутностей: ефективного та первинного користувача, і забороняти уособлення будь-якого користувача з Адміністратором. Іншим недоліком даного методу є той факт, що не враховується особливість впровадження загрозливого скриптового виконуваного файлу.

Додамо захист від скриптових виконуваних файлів. Їх головна відмінність в тому, що для їх виконання необхідно встановити інший інтерпретатор (середовище виконання). В даному випадку не користувач запускає виконуваний файл, а інтерпретатор. Таким чином, слід розглядати процес в якості самостійного суб'єкта доступу. До нього також слід застосовувати розмежувальні політики для управління доступом до ресурсів. Метод захисту враховує особливості скриптових виконуваних загрозливих файлів. У разі скриптових виконуваних файлів головне не дати записати, а потім не дати інтерпретатору запустити скрипт. Але проблема в тому, що інтерпретатор при доступі до даного файлу використовує метод доступу читання, а воно дозволено для всіх користувачів. В даному випадку можна розглянути всі інтерпретатори (процеси), які запускаються окремо і для кожного призначити свої права доступу. Будемо захищати безпосередньо від запису скриптового виконуваного файлу, а не від подальшого його читання. Отримаємо, що суб'єкт доступу слід задавати трьома сутностями: первинний користувач, ефективний користувач і процес. Основою моделі захисту від скриптових виконуваних файлів є захист від витоку права модифікації функцій санкціонованої програми. Особливістю методу є розбиття методу доступу «Запис» на створення нового об'єкту доступу і зміна існуючого об'єкта доступу шляхом перейменування. Заборона створення нового будемо позначати «~~ЗпН~~», а зміна існуючого - «~~ЗпІ~~». Суб'єкти доступу - $S_i : S = \{S_1, \ldots, S_k, A\}$. Об'єкти доступу залишаються незмінними: $O = \{ O_{вик1}, \ldots, O_{викq}, O_{сист1}, \ldots, O_{систm}, O_{інф1}, \ldots, O_{інфn} \}$. Кінцевий набір прав доступу включає в себе читання, виконання і запис, заборона створення нового ОД, заборона зміни існуючого ОД, заборона перейменування існуючого виконуваного файлу, заборона перейменування в виконуваний файл, заборона видалення. В результаті отримаємо наступний набір прав: *R* = {*Чт, В, Зп, ~~ЗпН~~, ~~ЗпІ~~, ~~Пв~~, ~~ПвВ~~, ~~Д~~*}.

Метод полягає в наступному:

- для суб'єкта доступу Адміністратор: дозволити читання, запис (установка) і виконання виконуваних ОД; дозволити читання, запис (установка) системних ОД; дозволити читання і запис інформаційних ОД; все інше заборонити.

- для всіх інших суб'єктів доступу: дозволити читання і виконання тільки тих виконуваних ОД, які вже знаходяться на системному диску; заборонити створення нового ОД, зміни існуючого ОД, перейменування існуючого виконуваного ОД, видалення існуючих виконуваних ОД на системному диску; дозволити читання для системних ОД; заборонити для системних ОД створення нового ОД, зміна існуючого ОД, перейменування існуючого ОД; дозволити для інформаційних ОД читання і запис; заборонити для інформаційних ОД перейменування існуючого ОД видалення; все інше заборонити.

Виходячи із заданих параметрів, отримуємо наступну матрицю доступу:

$$M = \begin{array}{c} \\ S_1 \\ \cdots \\ \cdots \\ S_k \\ \\ A \end{array} \begin{array}{c} \begin{array}{ccc} O_{вик1}, \ldots, O_{викq} & O_{сист1}, \ldots, O_{систm} & O_{інф1}, \ldots, O_{інфn} \end{array} \\ \left[\begin{array}{ccc} \text{Чт, В, ~~ЗпН~~, ~~ЗпІ~~,} & \text{Чт, ЗпН, ЗпІ,} & \text{Чт, ЗпН, ЗпІ,} \\ \text{~~Пв~~, ~~ПвВ~~, ~~Д~~} & \text{~~Пв~~, ~~ПвВ~~, ~~Д~~} & \text{~~Пв~~, ~~ПвВ~~, ~~Д~~} \\ & \cdots & \\ \text{Чт, В, ~~ЗпН~~, ~~ЗпІ~~,} & \text{Чт, ЗпН, ЗпІ,} & \text{Чт, ЗпН, ЗпІ,} \\ \text{~~Пв~~, ~~ПвВ~~, ~~Д~~} & \text{~~Пв~~, ~~ПвВ~~, ~~Д~~} & \text{~~Пв~~, ~~ПвВ~~, ~~Д~~} \\ \text{Чт,Зп, В} & \text{Чт,Зп} & \text{Чт, Зп} \end{array} \right]. \end{array}$$

У даній МД критичний витік права модифікації функцій в процесі виконання невиконуваних файлів. Іншими словами санкціонована програма наділяється загрозливими можливостями. Також критичний витік права доступу «*~~ЗпН~~*» і «*~~ЗпІ~~*». Оскільки послідовність дій, в результаті яких право «*~~ЗпН~~*» або «*~~ЗпІ~~*» буде занесено в комірку матриці, що відноситься до виконуваних ОД, генерує суб'єкт доступу (користувач), то його дії не зможуть привести до витоку права доступу. В тому числі при даній політиці виключена сутність «Власник», відповідно користувач, який створив новий об'єкт, не зможе наділити правом доступу «Запис» інших користувачів. При створенні нового СД також неможливий витік права доступу «Запис», так як розмежування застосовні до будь-якого користувача і не важливі ні його ім'я, ні його SID.

Перевагою даного методу є контроль запису скриптових виконуваних програм, тобто протидія їх впровадження. Недоліком даного методу є можливість отримання адміністративних прав і внаслідок чого необхідність додаткового захисту від підвищення привілею до адміністративних прав. Відповідно для усунення можливості підвищення привілею і для контролю зміни прав доступу необхідно суб'єкт доступу задавати виходячи з двох сутностей: ефективного і первинного користувача, і забороняти уособлення будь-якого користувача з Адміністратором. Іншим недоліком даного методу є відсутність контролю роботи браузера з інтерпретаторами.

Модель і метод захисту з додатковим захистом від наділення загрозливими властивостями інтерпретаторів (віртуальних машин). Тепер усунемо недолік попереднього методу: відсутність контролю, коли браузер, використовуючи інтерпретатор, відкриває файл для читання. Виходячи з матриці доступу попереднього методу, процес може читати виконуваний файл, і тим самим зможе його запустити. В такому випадку виділимо окремий суб'єкт доступу, який є інтерпретатором (віртуальна машина), наприклад, java.exe (JVM). Таким чином, в якості суб'єктів доступу виступатимуть користувачі системи і віртуальні машини $S_i : S = \{S_1, \ldots, S_k, A, VM_1, \ldots, VM_j\}$. ОД залишаються незмінними: $O = \{O_{вик1}, \ldots, O_{викq}, O_{сист1}, \ldots, O_{системm}, O_{інф1}, \ldots, O_{інфn}\}$. Кінцевий набір прав доступу включає в себе читання, виконання і запис, заборона створення нового ОД, заборона зміни існуючого ОД, заборона перейменування існуючого виконуваного файлу, заборона перейменування в виконуваний файл, заборона видалення. Введемо додатково право доступу - заборона читання (позначимо заборона читання як «*~~Чт~~*»). В результаті отримаємо наступний набір прав: $R = \{Чт, \sout{Чт}, В, Зп, \sout{ЗпН}, \sout{ЗпІ}, \sout{Пв}, \sout{ПвВ}, \sout{Д}\}$.

Метод передбачає захист від наділення загрозливими властивостями інтерпретаторів і полягає в наступному:

- для суб'єкта доступу Адміністратор: дозволити читання, запис (установка) і виконання виконуваних ОД; дозволити читання, запис (установка) системних ОД; дозволяти читання і запис інформаційних ОД; все інше забороняти.

- для суб'єктів доступу віртуальна машина (інтерпретатор): дозволити виконання тільки тих виконуваних ОД, які вже знаходяться на системному диску; заборонити читання, запис, перейменування існуючого виконуваного ОД, видалення існуючих виконуваних ОД на системному диску; дозволити читання для системних ОД; заборонити для системних ОД запис, перейменування існуючого ОД, видалення; дозволити для інформаційних ОД читання і запис; заборонити для інформаційних ОД перейменування існуючого ОД, видалення; все інше заборонити.

- для всіх інших суб'єктів доступу: дозволити читання і виконання тільки тих виконуваних ОД, які вже знаходяться на системному диску; заборонити запис, перейменування існуючого виконуваного ОД, видалення існуючих виконуваних ОД на системному диску; дозволити читання для системних ОД; заборонити для системних ОД запис, перейменування існуючого ОД, видалення; дозволити для інформаційних ОД читання і

запис; заборонити для інформаційних ОД перейменування існуючого ОД, видалення; все інше заборонити. Виходячи із заданих параметрів, отримуємо наступну матрицю доступу:

Перевагою методу є те, що після установки даних прав доступу ні бінарні, ні скриптові виконувані загрозливі програми не будуть загрожувати безпеці системи.

Недоліком даного методу є можливість отримання адміністративних прав і внаслідок чого необхідність додаткового захисту від підвищення привілею до адміністративних прав. Відповідно для усунення можливості підвищення привілею і для контролю зміни прав доступу необхідно суб'єкт доступу задавати виходячи з двох сутностей: ефективного та первинного користувача, і забороняти уособлення будь-якого користувача з Адміністратором.

	$O_{вик1}, \dots, O_{викq}$	$O_{сист1}, \dots, O_{систm}$	$O_{інф1}, \dots, O_{інфn}$
S_1	*Чт, В, ~~ЗпН~~, ~~ЗпІ~~,*	*Чт, ~~ЗпН~~, ~~ЗпІ~~,*	*Чт, Зп,*
…	*~~Пв~~, ~~ПвВ~~, ~~Д~~*	*~~Пв~~, ~~ПвВ~~, ~~Д~~*	*~~Пв~~, ~~ПвВ~~, ~~Д~~*
…		…	
S_k	*Чт, В, ~~ЗпН~~, ~~ЗпІ~~,*	*Чт, ~~ЗпН~~, ~~ЗпІ~~,*	*Чт, Зп,*
	~~Пв~~, ~~ПвВ~~, ~~Д~~	*~~Пв~~, ~~ПвВ~~, ~~Д~~*	*~~Пв~~, ~~ПвВ~~, ~~Д~~*
M = VM_1	*~~Чт~~, В, ~~ЗпН~~, ~~ЗпІ~~,*	*Чт, ~~ЗпН~~, ~~ЗпІ~~,*	*Чт, Зп,*
…	*~~Пв~~, ~~ПвВ~~, ~~Д~~*	*~~Пв~~, ~~ПвВ~~, ~~Д~~*	*~~Пв~~, ~~ПвВ~~, ~~Д~~*
…		…	
VM_j	*~~Чт~~, В, ~~ЗпН~~, ~~ЗпІ~~,*	*Чт, ~~ЗпН~~, ~~ЗпІ~~,*	*Чт, Зп,*
	~~Пв~~, ~~ПвВ~~, ~~Д~~	*~~Пв~~, ~~ПвВ~~, ~~Д~~*	*~~Пв~~, ~~ПвВ~~, ~~Д~~*
A	*Чт,Зп, В*	*Чт,Зп*	*Чт, Зп*

Висновки. Проведено дослідження основних типів загрозливих програм, на підставі якого запропоновано класифікацію загрозливих програм за способом їх виконання. На підставі існуючої статистики зроблено висновок, що найбільш актуальними для захисту є виконувані бінарні і скриптові файли.

Проведено дослідження способів впровадження загрозливих програм, в результаті якого зроблено висновок - класи загрозливих програм, що розглядаються передбачають обов'язкове збереження файлу на жорсткому диску перед виконанням (читанням). Для захисту від найбільш актуальних загрозливих програм потенційно може бути реалізований контроль доступу (розмежувальна політика доступу) до файлових об'єктів.

Результатом дослідження існуючих підходів до оцінки ефективності методів і засобів захисту від загрозливих програм, зроблені висновки про неможливість з використанням відомих підходів ні кількісно оцінити актуальність окремої загрози для інформаційної системи в цілому (з урахуванням безлічі інших потенційних загроз), в тому числі загрози занесення і запуску загрозливих програм, ні кількісно оцінити основні стохастичні характеристики безпеки системи від загрозливих програм. В результаті чого сформульована задача розробки відповідних математичних моделей і наступного проведення на них досліджень, що дозволяють отримати необхідні кількісні оцінки.

Запропоновано загальний підхід до захисту від загрозливих програм, заснований на контролі доступу до ресурсів по розширенням і типам файлів. Розроблені методи, що дозволяють захищати від бінарних і скриптових загрозливих файлів.

Розроблені моделі безпечної системи, що дозволяють сформулювати вимоги до побудови безпечної системи в частині запобігання витоку прав доступу. Сформульовані вимоги до механізмів захисту, реалізація яких дозволить побудувати безпечну систему, що обґрунтовано на побудованих моделях.

ЛІТЕРАТУРА:

1. Основы программно-аппаратной защиты информации. / М. А. Борисов, И. В. Заводцев, И. В. Чижов. – М.: УРСС: Либроком, 2013. – 370 с.

2. Михайлов А.В. Компьютерные вирусы и борьба с ними. / А.В. Михайлов. – М.: Диалог-МИФИ, 2012. – 148 с.

3. Касперский Е. В. «Компьютерное зловредство» / Е. В. Касперский. – Санкт-Петербург: Питер, 2009. – 208 с.

4. Партыка Т. Л. Информационная безопасность учебное пособие / Т. Л. Партыка, И. И. Попов. – М.: ФОРУМ, 2011. – 432 с.

5. Сердюк В. А. Организация и технологии защиты информации / В. А. Сердюк. – М.: Издательский дом Государственного университета – Высшей школы экономики, 2011. – 571 с.

6. Шаньгин В. Ф. Защита информации в компьютерных системах и сетях. / В. Ф. Шаньгин. – М.: ДМК Пресс, 2012. – 576 с.

7. Гольдштейн, Б.С. Сети связи пост-NGN/Б.С. Гольдштейн, А.Е. Кучерявый. – СПб.:БХВ-Петербург,2014. – 160с.: ил.

8. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы /В. Г. Олифер, Н. А.Олифер - СПб.: Питер, 2017. – 992 с.

9. Рыжиков, Ю.И. Имитационное моделирование. Теория и технология / Ю.И. Рыжиков - СПб: КОРОНА принт, 2015. – 384 с.

10. Советов, Б. Я. Моделирование систем : учебник для бакалавров / Б. Я. Советов, С. А. Яковлев. – 7-е изд. – М. : Издательство Юрайт, 2015. – 343 с.

11. Тюрликов, А. М. Методы случайного множественного доступа [Текст] : монография / А. М. Тюрликов – Санкт-Петербург : ГУАП, 2014. - 299 с. : ил.

REFERENCES:

1. Borisov, M.A., Zavodcev, I.V. and CHizhov, I.V. (2013). ”Osnovy programmno-apparatnoj zashchity informacii”, M.: URSS: Librokom, 370 p.

2. Mihajlov, A.V. (2012), ”Komp'yuternye virusy i bor'ba s nimi.”, M.: Dialog-MIFI, 148 p.

3. Kasperskij, E.V. (2009), ”Komp'yuternoe zlovredstvo”, Sankt-peterburg: Piter,. 208 p.

4. Partyka, T.L. and Popov, I.I. (2011). ”Informacionnaya bezopasnost' uchebnoe posobie” , M.: FORUM, 432 p.

5. Serdyuk, V. A. (2011). ”Organizaciya i tekhnologii zashchity informacii ”, M.: Izdatel'skij dom Gosudarstvennogo universiteta – Vysshej shkoly ekonomiki,. 571 p.

6. SHan'gin V. F. (2012). ”Zashchita informacii v komp'yuternyh sistemah i setyah ”. / V. F. SHan'gin. - M.: DMK Press, 576 p.

7. Goldshteyn, B.S. and Kucheryavy, A.E. (2014). ”Seti svyazi post-NGN”[Post-NGN communication networks], SPb.:BHV-Peterburg, 160p.: il.

8. Olifer, V. G. and Olifer, N. A. (2017). ”Kompyuternyie seti. Printsipyi, tehnologii, protokolyi ” [Computer networks. Principles, technologies, protocols], SPb.: Piter, 992p.

9. Ryizhikov, Yu.I. (2015). ”Imitatsionnoe modelirovanie. Teoriya i tehnologiya.” [Imitation modeling. Theory and technology], SPb: KORONA print, 384p.

10. Sovetov, B. Ya. and Yakovlev, S.A. (2015). ”Modelirovanie sistem : uchebnik dlya bakalavrov” [System modeling: a textbook for bachelors] ,7-e izd. M. : Izdatelstvo Yurayt, 343p.

11. Tyurlikov, A. M. (2014). ”Metodyi sluchaynogo mnozhestvennogo dostupa”[Random Multiple Access Methods], [Tekst] : monografiya , Sankt-Peterburg : GUAP, 299p.

к.т.н., доц. Джулий В.Н., к.т.н., доц. Бойчук В.А., к.т.н.Титова В.Ю.
д.т.н., с.н.с. Селюков А.В., к.т.н., с.н.с. Мирошниченко О.В.

МОДЕЛИ И МЕТОДЫ ЗАЩИТЫ ОТ ВРЕДОНОСНЫХ ПРОГРАММ ИНФОРМАЦИОННЫХ СИСТЕМ

В статье предложен подход к развитию методов защиты от вредоносных программ в современных информационных системах, состоит в разработке методов защиты, основанных на реализации контроля доступа к файлам по их типам, которые могут быть идентифицированы расширениями файлов, существенно превосходят известные методы антивирусной защиты, как

по эффективности защиты, так и по мере влияния на загрузку вычислительных ресурсов информационной системы.

Показано, что наиболее актуальными для защиты являются выполняемые бинарные и скриптовые файлы и о том, что данные классы вредоносных программ предусматривают обязательное сохранение вредоносного файла на жестком диске перед его выполнением (чтением). Это позволило сделать вывод о том, что защита от вредоносных программ может строиться реализацией контроля (разграничение прав) доступа к файлам.

Предложен общий подход к реализации защиты от вредоносных программ, основанный на реализации контроля доступа к файлам по их типам, которые могут быть идентифицированы расширениями файлов. Возможность использования подобного подхода обоснованно проведенным исследованиям средств защиты.

Методы защиты от вредоносных программ позволяют защитить информационную систему, как от загрузки, так и от выполнения бинарных и скриптовых вредоносных файлов, отличающиеся возможностью учета расположения исполняемых файлов, возможностью администрирования при работающей системе защиты, возможностью контроля модификации объектов доступа, переименование объектов доступа, возможностью защиты от скриптовых вредоносных программ, в том числе с учетом возможности наделения вредоносными свойствами интерпретаторов (виртуальных машин).

Разработаны модели контроля доступа, позволили на построенных матрицах доступа сформулировать требования к построению безопасной системы, выполнение которых предотвращает утечку заданных прав доступа субъектов к объектам.

Ключевые слова: вредоносные программы, компьютерная безопасность, информационная система, контроль доступа, объект доступа, субъект доступа.

Ph.D. Dzhulij V.M., Ph.D. Boychuk V.A., Ph.D. Titova V.Y.,
Doctor of Technical Sciences Selyukov A.V., Ph.D. Miroshnichenko O.V.

PROTECTION MODELS AND METHODS AGAINST THREATED PROGRAMS INFORMATION SYSTEMS

The article proposes an approach to the development of protection methods against threatening programs in modern information systems, which consists in the development of security methods based on the implementation of access control to files by their types, which can be identified by file extensions that significantly exceed the known methods of antivirus protection, such as on the effectiveness of protection, as well as the impact on the load of computing resources of the information system.

It is shown that the most important for protection are executable binary and script files, and that these classes of malware require mandatory storage of the threatening file on the hard disk before its execution (read). This led to the conclusion that protection against threatening programs can be built by implementing control (delineation) of access to files.

A general approach to the implementation of protection against threatening programs is proposed, based on the implementation of control of access to files by their types, which can be identified by file extensions. The possibility of using such an approach is substantiated by a study of remedies.

Methods of protection against threatening programs allow to protect the information system, both from loading, and from execution of binary and scripted threat files, differing in the possibility of taking into account the location of executable files, the possibility of administration with a working security system, the ability to control the modification of access objects, renaming access features, the ability to protect against scripted threat programs, including the ability to give threatening properties to interpreters (virtual x machines).

Models of access control have been developed, which allowed the built-in access matrices to formulate requirements for building a secure system, the implementation of which prevents the leakage of given access rights of subjects to objects.

Keywords: threatening programs, computer security, information system, access control, object of access, subject of access.