КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (13), 2021
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2021.13.145157
УДК 004.056.53

Гнатюк Сергій Олександрович
д.т.н., доцент, заступник декана факультету кібербезпеки, комп'ютерної та програмної інженерії
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0003-4992-0564
s.gnatyuk@nau.edu.ua

Верховець Олексій Сергійович
заступник начальника науково-дослідного центру
Державний науково-дослідний інститут технологій кібербезпеки та захисту інформації, Київ, Україна
ORCID ID: 0000-0002-3897-106X
o.s.verhts@gmail.com

Толбатов Андрій Володимирович
к.т.н., доцент, докторант факультету кібербезпеки, комп'ютерної та програмної інженерії
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0002-9785-9975
tolbatov@ukr.net

Красовська Євгенія Вікторівна
к.т.н., викладач
Фаховий коледж інженерії та управління
Національний авіаційний університет, Київ, Україна
ORCID ID: 0000-0002-5582-0984
krasovevg@gmail.com

ФОРМАЛІЗАЦІЯ ІНФОРМАЦІЙНИХ ПОТОКІВ ДЛЯ ЗАХИСТУ ОПЕРАЦІЙНИХ СИСТЕМ СІМЕЙСТВА BSD ВІД НЕСАНКЦІОНОВАНОГО ДОСЛІДЖЕННЯ

Анотація. Сьогодні спостерігається збільшення кількості та підвищення складності кібератак на об'єкти критичної інфраструктури. Це зумовило актуалізацію питання захисту систем, які є критично важливими для забезпечення національної безпеки. Програмне забезпечення (ПЗ), у тому числі операційні системи (ОС), розглядаються, як ресурс критичної інформаційної інфраструктури держави, яка як правило, будується на захищених ОС (UNIX-подібних, ОС сімейства BSD, Linux). Але будь-які ОС та користувацьке ПЗ мають недоліки та проблеми з безпекою на різних рівнях. Актуальним є моделювання інформаційних потоків в ОС, що дозволить більш ефективно виявляти загрози безпеці інформації, реалізувати превентивні і контрзаходи. З цих позицій, у роботі було проведено аналіз сучасних досліджень у напрямку захисту ОС та користувацького ПЗ, що дозволив виявити кілька базових напрямків, серед яких дослідження впливу шкідливого ПЗ на ОС та користувацьке ПЗ; аналіз уразливостей, а також дослідження загроз і ризиків. Проведений аналіз показав, що відкритими залишаються питання, пов'язані з урахуванням особливостей побудови та інформаційних процесів конкретної ОС, а також відсутністю адекватних математичних моделей, котрі можуть бути застосовані для різних систем захисту з метою одержання кількісних характеристик для порівняння параметрів систем захисту. Також, було розроблено структурно-аналітичні моделі інформаційних потоків ОС сімейства BSD, що дає можливість формалізувати інформаційні процеси досліджуваної операційної системи і розробляти ефективні превентивні та контрзаходи. Крім того, удосконалено математичну модель кількісного оцінювання програмних систем захисту інформації, що функціонують у користувацькому режимі. Ця модель буде корисною як для порівняння існуючих програмних систем захисту інформації, так і для аналізу змін в алгоритмах захисту програмних систем захисту інформації.

ISSN 2663 - 4023

Ключові слова: програмний захист інформації, операційна система, інформаційний потік, несанкціоноване дослідження, структурно-аналітична модель, математична модель, BSD.

1. ВСТУП

Збільшення кількості та підвищення складності кібератак на об'єкти критичної інфраструктури [1] зумовили актуалізацію питання захисту систем, які є критично важливими для забезпечення національної безпеки. Програмне забезпечення (ПЗ), у тому числі операційні системи (ОС), розглядаються, як ресурс критичної інформаційної інфраструктури держави. Критична інфраструктура будується на захищених ОС: UNIX-подібних, ОС сімейства BSD, Linux. Але будь-які ОС та користувацьке ПЗ мають недоліки та проблеми з безпекою на різних рівнях. Актуальним є моделювання інформаційних потоків в ОС, що дозволить більш ефективно виявляти загрози безпеці інформації, реалізувати превентивні і контрзаходи.

2. АНАЛІЗ ІСНУЮЧИХ ДОСЛІДЖЕНЬ І ПОСТАНОВКА ЗАВДАННЯ

Аналіз досліджень у напрямку захисту ОС та користувацького ПЗ дозволив виявити кілька базових напрямків [2-5]: 1) дослідження впливу шкідливого ПЗ на ОС та користувацьке ПЗ (дослідники Тінка Т., Смірнов О., Шадхін В. та інші); 2) аналіз уразливостей (дослідники Семенов С., МакГров Г. та інші); 3) дослідження загроз і ризиків (дослідники Козиракіс К., Ін-Шенг С. та інші).

Проведений аналіз показав, що відкритими залишаються питання, пов’язані з відсутністю адекватних математичних моделей, котрі можуть бути застосовані для різних систем захисту з метою одержання кількісних характеристик для порівняння параметрів систем захисту [6]. Існуючі моделі програмних систем захисту інформації розроблялися для конкретних систем захисту з метою урахування особливостей їх побудови, вони не призначені для оцінки довільної програмної системи захисту інформації чи конкретної ОС [7]. Одержання та порівняння кількісних характеристик ускладнено через використання моделями низки чинників, які важко формалізувати (кваліфікація дослідника, наявність у дослідника засобів сканування).

З огляду на це, **метою роботи** є спроба формалізації інформаційних потоків для більш ефективного захисту ОС сімейства BSD від несанкціонованого дослідження (НСД).

3. ОСНОВНА ЧАСТИНА ДОСЛІДЖЕННЯ

З метою пошуку можливих розв’язки завдання необхідно:

- розробити класифікації інформаційних потоків та структурних моделей програмного середовища в аспекті захисту від засобів НСД;
- на основі створеної класифікації та структурних моделей вдосконалити математичну модель, котра дозволить розраховувати кількісні характеристики для оцінки ефективності протидії засобам сканування.

3.1. Класифікація інформаційних потоків

ОС BSD складається з наступних програмних компонентів:

- завантажувача ядра ОС;
- ядра ОС;
- файлової системи;
- головного процесу *init;*
- командного інтерпретатору;
- системного і прикладного ПЗ;
- файлів.

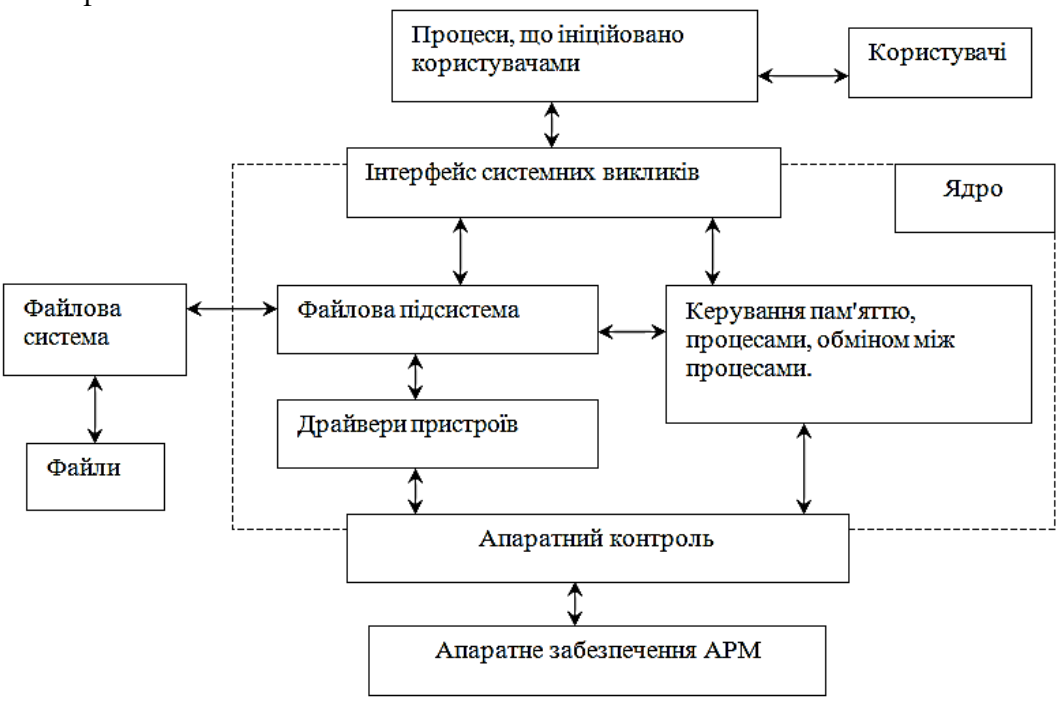

Рис.1. Функціональна структура BSD-системи

Визначимо поняття інформаційного потоку в середовищі функціонування програмної системи захисту інформації ОС.

Інформаційний потік – віртуальний канал обміну, який є в наявності між блоками структурної моделі тоді й лише тоді, коли між цими блоками відбувається обмін даними хоча б в одному напрямку. Особливістю інформаційного потоку є той факт, що подібний обмін даними за умови застосування засобів сканування, може бути проконтрольований. У будь-якій програмній системі захисту інформації ОС інформаційний потік, що захищається, повинен контролюватися з боку системи захисту способами протидії засобам сканування.

Під *контролем інформаційних потоків* будемо мати на увазі дії з боку засобів сканування, які спрямовані на одержання доступу до переданих віртуальними каналами даних з метою їх перегляду, модифікації або блокування. Контроль із боку засобів сканування буде повним, якщо він у змозі одержати доступ до початкових, проміжних та кінцевих даних, що передаються.

Виділимо наступні *три категорії інформаційних потоків,* що присутні у момент роботи застосунка:

1) інформаційні потоки, що існують усередині завантаженого образу файлу. Характерною рисою цієї категорії інформаційних потоків є їхній складний аналіз із боку засобів дослідження програмного коду, тому що такі інформаційні потоки не використовують для передачі даних ніяких допоміжних структур, доступ до яких може

бути легко перехоплений засобами сканування. Прикладом даної категорії інформаційних потоків може слугувати обмін даними між двома секціями даних того самого завантаженого образу файлу. Очевидно, що подібний обмін може бути здійснений без звернення до функцій операційної системи, які можуть контролюватися засобами сканування потенційного зловмисника (на структурних моделях дана категорія інформаційних потоків позначена суцільними лініями.);

2) інформаційні потоки взаємодії файлу, що виконується, з бібліотеками, що динамічно завантажуються. Така взаємодія здійснюється через таблиці імпорту й/або експорту файлу або шляхом використання API-функцій dlopen, dlclose, dlsym. Характерною особливістю даної категорії інформаційних потоків є їхній можливий аналіз із боку засобів дослідження програмного коду. Даний аналіз може бути здійснений шляхом модифікації таблиць імпорту й/або експорту цільової бібліотеки, що динамічно завантажується, або шляхом впровадження бібліотеки, що є програмною закладкою (на структурних моделях, що наводяться, дана категорія інформаційних потоків позначена штриховими лініями);

3) інформаційні потоки взаємодії користувацького режиму й режиму ядра. Можуть бути присутнім як в образі файлу, що виконується, так і в образі бібліотек, що динамічно завантажуються. Характерною рисою даної категорії інформаційних потоків є можливість їх повного контролю з боку засобів сканування та складний контроль із боку програмної підсистеми захисту логіки роботи цілісності переданої інформації. Такі складнощі виникають через відсутність прямого доступу до режиму ядра з боку програмної системи захисту інформації користувацького режиму, що ускладнює виявлення засобів сканування потенційного зловмисника, розташованих у режимі ядра (на наведених структурних моделях, дана категорія інформаційних потоків позначена штрих пунктирними лініями).

Розділення інформаційних потоків на зазначені категорії зроблене з таких причин: уведені категорії інформаційних потоків дозволяють ураховувати місця впровадження засобів дослідження програмного коду; уведені категорії інформаційних потоків враховують складність сканування кожної категорії з боку засобів дослідження програмного коду.

При вдосконаленні математичної моделі оцінки надійності програмної системи захисту інформації на основі марковських процесів, складність сканування може бути показана за допомогою інтенсивності подій. Уведені категорії інформаційних потоків присутні в будь-якому адресному просторі користувацького застосунка операційної системи, що дозволяє застосовувати розроблену класифікацію для довільних програмних систем захисту інформації.

Уведена класифікація інформаційних потоків використана при побудові структурних моделей програмного середовища в аспекті захисту інформації від засобів сканування.

3.2. Структурно-аналітичні моделі

Для наведених структурних моделей прийняті наступні позначення:

- користувацький режим позначається «Ring 3»;
- режим ядра позначається «Ring 0».

Структурна модель інформаційних потоків програмного середовища користувацького застосунку представлена на рис. 2.

Запропонована структурна модель враховує наступні особливості функціонування користувацьких застосунків:

1) відсутність прямого доступу до адресного простору нульового кільця захисту з боку коду користувацьких застосунків;

2) паралельна робота користувацьких застосунків, що відзначене у вигляді адресних просторів, що розділяються ;

3) відсутність можливості завантажити довільний файл, що виконується без бібліотек, що динамічно завантажуються. З цього випливає, що в будь-якому адресному просторі користувацького застосунка буде присутня одна або декілька бібліотек. Крім цього, виклики динамічних бібліотек можуть бути вкладеними, в адресному просторі застосунку може бути присутня бібліотека, завантаження яких сам, файл, що виконується не робив.

Доступ до нульового кільця з адресного простору користувацьких застосунків може здійснюватися лише через інтерфейс системних викликів.

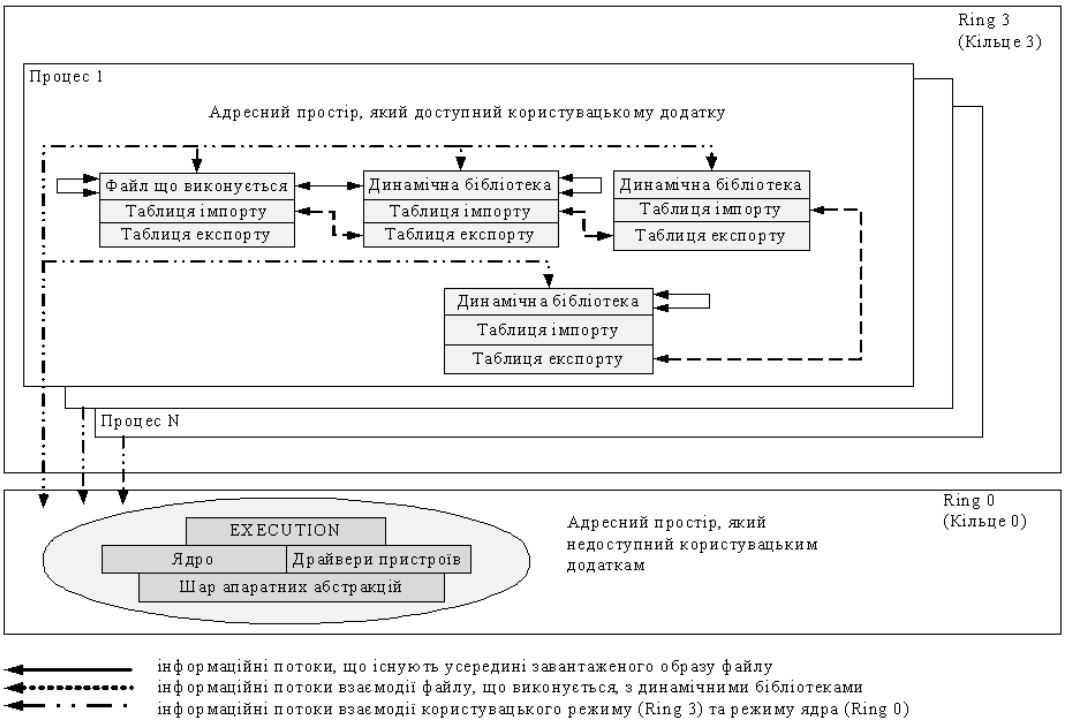

Рис.2. Структурна модель інформаційних потоків програмного середовища користувацького застосунку

Розподіл адресних просторів застосунків досягається перезаписом значення в керуючому регістрі базової адреси сторінкових таблиць - CR3. Адресний простір автоматично створюється для будь-якого нового процесу (на структурній моделі позначені процеси № 1...N). Для ефективного задіяння оперативної пам'яті операційною системою використовуються бібліотеки, що динамічно завантажуються. Ці бібліотеки завантажуються в адресні простори тільки тих застосунків, яким вони необхідні. Зв'язок між ними показано пунктирними лініями через таблиці імпорту й експорту.

Розглянуті особливості функціонування користувацьких застосунків впливають на розробку в програмних системах захисту інформації ефективної протидії засобам сканування.

Відобразимо на структурну модель результати проведеного аналізу засобів дослідження програмного коду. Результат виконаного відображення наведений на рис. 3,

який показує можливі впливи на адресний простір застосунка з боку засобів сканування програмного коду.

Сірим кольором (рис.3) показані інструменти дослідження програмного коду, що використовуються для сканування програмного середовища. Сірі стрілки показують впливи, які може здійснювати той або інший засіб сканування з метою одержання несанкціонованого доступу.

Сірі стрілки, котрі не ведуть до інформаційних потоків, а вказують на адресний простір застосунка, схематично позначають можливість контролю всього адресного простору, а отже, усіх категорій інформаційних потоків.

За допомогою засобів сканування без застосування способів протидії з боку програмної системи захисту інформації, може бути здійснений контроль усіх категорій інформаційних потоків.

Вплив активних емуляторів і зневаджувачів (дебагерів) користувацького режиму на аналізований застосунок прямо неможливий. Здійснення такої взаємодії можливо тільки через спеціальні механізми міжпроцесної взаємодії, підтримувані операційною системою. Звертання до механізмів міжпроцесної взаємодії з боку засобів сканування приводить до виклику сервісів ядра через інтерфейс системних викликів. Таким чином, якщо засоби сканування використовують механізми міжпроцесної взаємодії, то вони неявно звертаються до режиму ядра операційної системи, що на структурній моделі показано за допомогою стрілок, що впливають на адресний простір досліджуваного застосунка через нульове кільце захисту.

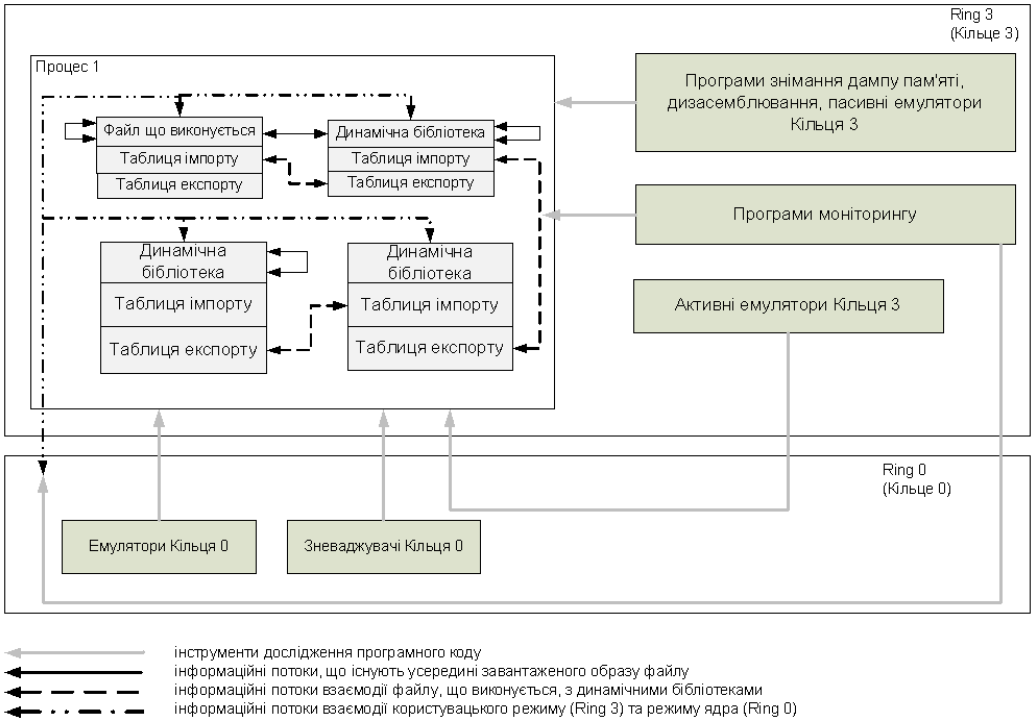

Рис.3. Структурна модель інформаційних потоків програмного середовища користувацького застосунку в контексті НСД

Вплив через нульове кільце захисту на досліджуваний застосунок також можуть виявляти програми моніторингу за умови використання драйверів-фільтрів.

Контроль інформаційних потоків програмного середовища з боку засобів НСД полегшує наявність централізованих структур даних, які використовуються для операцій передачі керування.

Існуючі способи протидії можуть бути використані для протидії лише частини розглянутих засобів НСД.

Розглянемо засоби НСД, яким може бути зроблена ефективна протидія існуючими способами захисту з боку програмної системи захисту інформації, що функціонує в режимі користувача:

- пасивні емулятори коду, що функціонують у користувацькому режимі;
- зневаджувачі, що функціонують у користувацькому режимі;
- засоби статичного сканування;
- програми одержання дампу пам'яті (зніманню дампу пам'яті ефективно протидіє використання шифрування й псевдокоду, але способів протидії самому процесу знімання дампу на теперішній момент не існує).

Наявні способи протидії, які використовуються в програмній підсистемі захисту логіки роботи, не можуть протидіяти наступним засобам НСД:

- програми моніторингу роботи аналізованого застосунку;
- активні емулятори коду, що функціонують у користувацькому режимі;
- що функціонують у режимі ядра (відлагоджувачі, емулятори).

Успішна протидія не може бути зроблена найбільш потужним засобам НСД, що функціонують у режимі ядра. Схематично це показано наявністю доступу (стрілки сірого кольору) з боку засобів сканування режиму ядра до адресного простору користувацького застосунка.

Структурна модель (рис. 3), підтверджує висновки про необхідність створення нових способів протидії засобам НСД.

3.3. Математична модель

Програмна система захисту інформації буде повністю зламана, якщо може бути здійснений контроль над усіма категоріями інформаційних потоків. Здійснення подібного контролю можливо у двох випадках:

1) інформаційні потоки відповідної категорії не були захищені;
2) для відповідної категорії інформаційних потоків здійснена успішна нейтралізація способів протидії засобам НСД.

Для введених категорій інформаційних потоків введемо наступні позначення – інформаційні потоки категорій 1 – 3 будемо позначати F_1 – F_3 відповідно. Усі розглянуті категорії інформаційних потоків є незалежними один від одного.

Під час аналізу довільної система захисту інформації засобами дослідження програмного коду, вона може перебувати в наступних станах, тобто процес злому може бути описаний наступними станами:

1) S_1 - система не зламана, засоби дослідження програмного коду успішно нейтралізуються реалізованими для інформаційних потоків категорій F_1 - F_3 способами протидії;

2) S_2 - система не зламана, зламані (або не були захищені) реалізовані для інформаційного потоку F_1 способи протидії, інформаційні потоки F_3 і F_2 успішно протидіють засобам сканування;

3) S_3 - система не зламана, зламані (або не були захищені) реалізовані для інформаційного потоку F_2 способи протидії, інформаційні потоки F_1 і F_3 успішно протидіють засобам сканування;

4) S_4 - система не зламана, зламані (або не були захищені) реалізовані для інформаційного потоку F_3 способи протидії, інформаційні потоки F_1 і F_2 успішно протидіють засобам сканування;

5) S_5 - система не зламана, зламані (або не були захищені) реалізовані для інформаційних потоків F_1 і F_2 способи протидії, інформаційний потік F_3 успішно протидіє засобам сканування;

6) S_6 - система не зламана, зламані (або не були захищені) реалізовані для інформаційних потоків F_1 і F_3 способи протидії, інформаційний потік F_2 успішно протидіє засобам сканування;

7) S_7 - система не зламана, зламані (або не були захищені) реалізовані для інформаційних потоків F_2 і F_3 способи протидії, інформаційний потік F_1 успішно протидіє засобам сканування;

8) S_8 (поглинаючий стан) - програмна система захисту інформації зламана, інформаційні потоки категорій F_1 – F_3 контролюються засобами сканування (зламані або не були захищені).

Таким чином, будь-яка програмна система захисту інформації може перебувати у восьми визначених вище станах.

Початковими станами можуть бути наступні:

1) S_1 – захищені від засобів сканування інформаційні потоки категорій F_1 – F_3;
2) S_2 – захищені від засобів сканування інформаційні потоки категорій F_2, F_3;
3) S_3 – захищені від засобів сканування інформаційні потоки категорій F_1,F_3;
4) S_4 – захищені від засобів сканування інформаційні потоки категорій F_1,F_2;
5) S_5 – захищені від засобів сканування інформаційні потоки категорії F_3;
6) S_6 – захищені від засобів сканування інформаційні потоки категорії F_2;
7) S_7 – захищені від засобів сканування інформаційні потоки категорії F_1.

Таким чином, початковий стан для розглянутої програмної системи захисту інформації визначається наявністю реалізованих способів протидії для відповідної категорії інформаційних потоків.

Для переведення програмної системи захисту інформації в стани, перехід у які характеризуються нейтралізацією способів протидії потоків категорії F_1 (перехід з S_1 в S_2, перехід з S_3 в S_5, перехід з S_4 в S_6, перехід з S_7 в S_8) діє пуасонівський потік успішних спроб нейтралізації способів протидії для F_1. Інтенсивність цього пуасонівського потоку дорівнює

$$\lambda_1(t) = \lambda_{12}(t) = \lambda_{35}(t) = \lambda_{46}(t) = \lambda_{78}(t) \quad (1)$$

Для переведення системи в стани, яке характеризується нейтралізацією способів протидії потоків категорії F_2 (перехід з S_1 в S_3, перехід з S_2 в S_5, перехід з S_4 в S_7, перехід з S_6 в S_8,), діє пуасонівський потік успішних спроб нейтралізації способів протидії для F_2. Інтенсивність цього потоку дорівнює

$$\lambda_2(t) = \lambda_{13}(t) = \lambda_{25}(t) = \lambda_{47}(t) = \lambda_{68}(t) \quad (2)$$

Для переведення системи в стани, які характеризуються нейтралізацією способів протидії потоків класу F_3 (перехід з S_1 в S_4, перехід з S_2 в S_6, перехід з S_3 в S_7, перехід з S_5 в S_8) діє пуасонівський потік успішних спроб нейтралізації способів протидії для F_3. Інтенсивність цього потоку дорівнює

$$\lambda_3(t) = \lambda_{14}(t) = \lambda_{26}(t) = \lambda_{37}(t) = \lambda_{58}(t) \quad (3)$$

ISSN 2663 - 4023

Таким чином, у програмної системи захисту інформації, що функціонує в користувацькому режимі, представленої графом *G*, діють потоки подій із трьома різними інтенсивностями:

$\lambda_1(t)$ – інтенсивність потоку успішних спроб нейтралізації способів протидії засобам сканування інформаційних потоків класу F_1;

$\lambda_2(t)$ – інтенсивність потоку успішних спроб нейтралізації способів протидії засобам сканування інформаційних потоків класу F_2;

$\lambda_3(t)$ – інтенсивність потоку успішних спроб нейтралізації способів протидії засобам сканування інформаційних потоків класу F_3.

Граф *G* з урахуванням уведених позначень інтенсивностей інформаційних потоків прийме вид, представлений на рис. 4. Для скорочення запису приймемо наступне позначення:

$$\lambda_i(t) = \lambda_i \tag{4}$$

Процес одержання контролю засобів сканування над програмною системою захисту інформації, що функціонує в користувацькому режимі, може бути представлений за допомогою графа, наведеного на рис. 4.

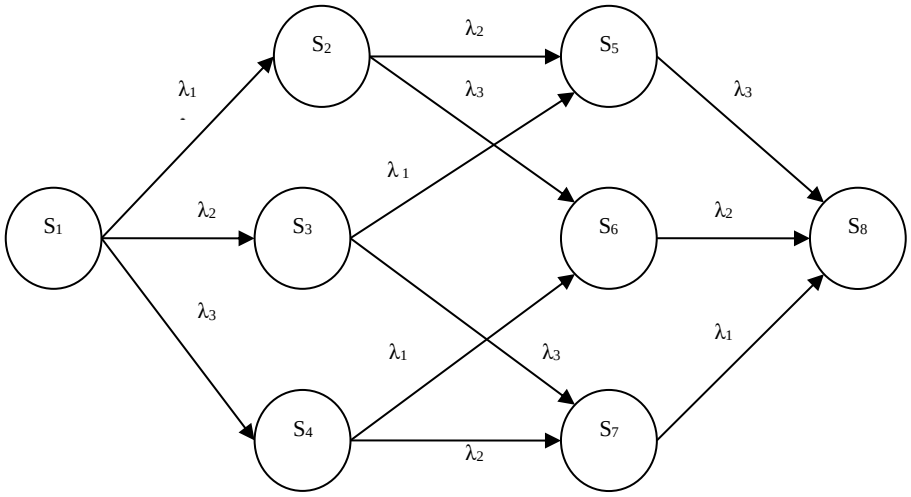

Рис. 4. Граф користувацького режиму G

Покажемо, що процес «злому» програмної системи захисту інформації, що функціонує в користувацькому режимі, який представлений графом *G* на рис. 4, може бути описаний за допомогою теорії марківських процесів з дискретними станами й безперервним часом:

- система містить кінцеву множину станів;
- умовні ймовірності знаходження системи в кожному зі станів не залежать від того, коли і як система прийшла в цей стан;
- потоки подій, що переводять систему зі стану в стан є пуасонівськими (ординарні, стаціонарні, без післядії).

Оскільки процес, що протікає в системі, яка представлена на рис. 4, є марківським, його можна представити за допомогою рівнянь Колмогорова. Система рівнянь Колмогорова, що описує граф *G*, прийме наступний вид:

$$\begin{cases} \frac{dp_1(t)}{dt} = -p_1(t)*(\lambda_1(t)+\lambda_2(t)+\lambda_3(t)) \\ \frac{dp_2(t)}{dt} = p_1(t)*(\lambda_1(t) - p_2(t)*(\lambda_2(t)+\lambda_3(t)), \\ \frac{dp_3(t)}{dt} = p_1(t)*\lambda_2(t) - p_3(t)*(\lambda_1(t)+\lambda_3(t)) \\ \frac{dp_4(t)}{dt} = p_1(t)*\lambda_3(t) - p_4(t)*(\lambda_1(t)+\lambda_2(t)) \\ \frac{dp_5(t)}{dt} = p_2(t)*\lambda_2(t) + p_3(t)*\lambda_1(t) - p_5(t)*\lambda_3(t) \\ \frac{dp_6(t)}{dt} = p_2(t)*\lambda_3(t) + p_4(t)*\lambda_1(t) - p_6(t)*\lambda_2(t) \\ \frac{dp_7(t)}{dt} = p_3(t)*\lambda_3(t) + p_4(t)*\lambda_2(t) - p7(t)*\lambda_1(t) \\ \frac{dp_8(t)}{dt} = p_5(t)*\lambda_3(t) + p_6(t)*\lambda_2(t) + p_7(t)*\lambda_3(t) \end{cases} \quad (5)$$

Застосування нормувальної умови дозволяє скоротити число рівнянь системи на одиницю. Нормувальна умова має такий вигляд:

$$\sum_{i=1}^{8} p_i(t) = 1. \quad (6)$$

Відповідно до нормувальної умови перепишемо $p_1(t)$ у такий спосіб:

$$p_1(t) = 1 - p_2(t) - p_3(t) - p_4(t) - p_5(t) - p_6(t) - p_7(t) - p_8(t) = 1 - \sum_{i=2}^{8} p_i(t). \quad (7)$$

Після застосування нормувальної умови, яка вірна в будь-який момент часу t, одержимо вираз (7). Після підстановки виразу (7) у систему рівнянь (5), одержимо систему рівнянь (8):

$$\begin{cases} \frac{dp_2(t)}{dt} = \left(1 - \sum_{i=2}^{8} p_i(t)\right)*\lambda_1(t) - p_2(t)*(\lambda_2(t)+\lambda_3(t)), \\ \frac{dp_3(t)}{dt} = \left(1 - \sum_{i=2}^{8} p_i(t)\right)*\lambda_2(t) - p_3(t)*(\lambda_1(t)+\lambda_3(t)), \\ \frac{dp_4(t)}{dt} = \left(1 - \sum_{i=2}^{8} p_i(t)\right)*\lambda_3(t) - p_4(t)*(\lambda_1(t)+\lambda_2(t)), \\ \frac{dp_5(t)}{dt} = p_2(t)*\lambda_2(t) + p_3(t)*\lambda_1(t) - p_5(t)*\lambda_3(t), \\ \frac{dp_6(t)}{dt} = p_2(t)*\lambda_3(t) + p_4(t)*\lambda_1(t) - p_6(t)*\lambda_2(t), \\ \frac{dp_7(t)}{dt} = p_3(t)*\lambda_3(t) + p_4(t)*\lambda_2(t) - p_7(t)*\lambda_1(t), \\ \frac{dp_8(t)}{dt} = p_5(t)*\lambda_3(t) + p_6(t)*\lambda_2(t) + p_7(t)*\lambda_1(t), \end{cases} \quad (8)$$

Отримана система рівнянь дозволяє визначити ймовірність злому програмних систем захисту інформації у випадкові моменти часу.

Удосконалена модель дозволяє отримати кількісні оцінки для розглянутих програмних систем захисту інформації, що функціонують у користувацькому режимі.

Отримані оцінки можуть бути використані як для порівняння існуючих програмних систем захисту інформації, так і для аналізу змін в алгоритмах захисту програмних

систем захисту інформації. Отриману кількісну оцінку можна розглядати як ймовірність того, що програмну систему захисту інформації не буде зламано протягом певного періоду часу. Для отримання оцінки використовується система рівнянь (9).

Удосконалена математична модель відповідає результатам експериментів і може використовуватися для будь-яких програмних систем захисту інформації, призначених для роботи в режимі користувача.

4. ВИСНОВКИ

У роботі було проведено аналіз сучасних досліджень у напрямку захисту ОС та користувацького ПЗ, що дозволив виявити кілька базових напрямків, серед яких дослідження впливу шкідливого ПЗ на ОС та користувацьке ПЗ; аналіз уразливостей та дослідження загроз і ризиків. Проведений аналіз показав, що відкритими залишаються питання, пов’язані з урахуванням особливостей побудови та інформаційних процесів конкретної ОС, а також відсутністю адекватних математичних моделей, котрі можуть бути застосовані для різних систем захисту з метою одержання кількісних характеристик для порівняння параметрів систем захисту.

Розроблено структурно-аналітичні моделі інформаційних потоків ОС сімейства BSD, що дає можливість формалізувати інформаційні процеси досліджуваної операційної системи і розробляти ефективні превентивні та контрзаходи.

Удосконалено математичну модель кількісного оцінювання програмних систем захисту інформації, що функціонують у користувацькому режимі. Ця модель буде корисною як для порівняння існуючих програмних систем захисту інформації, так і для аналізу змін в алгоритмах захисту програмних систем захисту інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1 Gnatyuk, S. (2016). Critical Aviation Information Systems Cybersecurity. *Meeting Security Challenges Through Data Analytics and Decision Support, NATO Science for Peace and Security Series, D: Information and Communication Security. IOS Press Ebooks*, *147*(3), 308-316.

2 Delimitrou, C., & Kozyrakis, C. (2016). Security Implications of Data Mining in Cloud Scheduling. *IEEE Computer Architecture Letters*, *15*(2), 109–112. https://doi.org/10.1109/lca.2015.2461215

3 Ravi, S., Kocher, P., Lee, R., McGraw, G., & Raghunathan, A. (2004). Security as a new dimension in embedded system design. У *the 41st annual conference*. ACM Press. https://doi.org/10.1145/996566.996771

4 Kaur, K., Garg, S., Kaddoum, G., Bou-Harb, E., & Choo, K.-K. R. (2020). A Big Data-Enabled Consolidated Framework for Energy Efficient Software Defined Data Centers in IoT Setups. *IEEE Transactions on Industrial Informatics*, *16*(4), 2687–2697. https://doi.org/10.1109/tii.2019.2939573

5 Alimseitova, Z., Adranova A,., Akhmetov, B., Lakhno, V., Zhilkishbayeva, G., Smirnov, O. Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources. *Journal of Theoretical and Applied Information Technology*, *98*(21), 3334-3346.

6 Gnatyuk, S., Berdibayev, R., Avkurova, Z., Verkhovets, O., Bauyrzhan, M. (2021). Studies on cloud-based cyber incidents detection and identification in critical infrastructure. *CEUR Workshop Proceedings*, *2923,* 68-80.

7 Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2021). Systematic Mapping Study on Security Approaches in Secure Software Engineering. *IEEE Access*, *9*, 19139–19160. https://doi.org/10.1109/access.2021.3052311

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 1 (13), 2021
ISSN 2663 - 4023

Sergiy O. Gnatyuk
DSc, Associate Professor, Vice-Dean of the Faculty of Cybersecurity, Computer and Software Engineering
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0003-4992-0564
s.gnatyuk@nau.edu.ua

Oleksii S. Verkhovets
Vice-Chief of the Research Center
State Scientific and Research Institute of Cybersecurity Technologies and Information Protection, Kyiv, Ukraine
ORCID ID: 0000-0002-3897-106X
o.s.verhts@gmail.com

Andrii V. Tolbatov
PhD, Associate Professor, Post-Doc Student of the Faculty of Cybersecurity, Computer and Software Engineering
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0002-9785-9975
tolbatov@ukr.net

Yevheniia V. Krasovska
PhD, Teacher
Professional College of Engineering and Management
National Aviation University, Kyiv, Ukraine
ORCID ID: 0000-0002-5582-0984
krasovevg@gmail.com

INFORMATION FLOWS FORMALIZATION FOR BSD FAMILY OPERATING SYSTEMS SECURITY AGAINST UNAUTHORIZED INVESTIGATION

Abstract. Today there is an increase in the number and complexity of cyberattacks on critical infrastructure. This has led to the actualization of the security systems that are critical to national security. Software, including operating systems, is considered a resource of critical information infrastructure of the state, which is usually built on secure operating systems (UNIX, BSD family, Linux). But any operating systems and user software have flaws and security issues at different levels. It is important to model information flows in the operating systems, which will more effectively identify threats to information security, implement preventive and countermeasures. From these positions, the analysis of modern research in the direction of operating systems security and user software was carried out, which allowed to identify several basic areas, including the study of the impact of malware on operating systems and user software; vulnerability analysis; threat and risk research. The analysis showed that the issues related to the peculiarities of construction and information processes of a particular operating systems, as well as the lack of adequate mathematical models that can be applied to different security systems to obtain quantitative characteristics to compare the parameters of security systems. Also, structural and analytical models of information flows of the BSD family of operating systems were developed, which makes it possible to formalize the information processes of the studied operating system and develop effective preventive and countermeasures. In addition, the mathematical model of quantitative evaluation of software systems for information security operating in user mode has been improved. This model will be useful both for comparison of existing software information security systems, and for the analysis of changes in security algorithms of software information security systems.

Keywords: software information security, operating system, information flow, unauthorized investigation, structural-analytical model, mathematical model, BSD.

REFERENCES

1 Gnatyuk, S. (2016). Critical Aviation Information Systems Cybersecurity. *Meeting Security Challenges Through Data Analytics and Decision Support, NATO Science for Peace and Security Series, D: Information and Communication Security. IOS Press Ebooks*, *147*(3), 308-316.

2 Delimitrou, C., & Kozyrakis, C. (2016). Security Implications of Data Mining in Cloud Scheduling. *IEEE Computer Architecture Letters*, *15*(2), 109–112. https://doi.org/10.1109/lca.2015.2461215

3 Ravi, S., Kocher, P., Lee, R., McGraw, G., & Raghunathan, A. (2004). Security as a new dimension in embedded system design. У *the 41st annual conference*. ACM Press. https://doi.org/10.1145/996566.996771

4 Kaur, K., Garg, S., Kaddoum, G., Bou-Harb, E., & Choo, K.-K. R. (2020). A Big Data-Enabled Consolidated Framework for Energy Efficient Software Defined Data Centers in IoT Setups. *IEEE Transactions on Industrial Informatics*, *16*(4), 2687–2697. https://doi.org/10.1109/tii.2019.2939573

5 Alimseitova, Z., Adranova A,., Akhmetov, B., Lakhno, V., Zhilkishbayeva, G., Smirnov, O. Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources. *Journal of Theoretical and Applied Information Technology*, *98*(21), 3334-3346.

6 Gnatyuk, S., Berdibayev, R., Avkurova, Z., Verkhovets, O., Bauyrzhan, M. (2021). Studies on cloud-based cyber incidents detection and identification in critical infrastructure. *CEUR Workshop Proceedings*, *2923,* 68-80.

7 Khan, R. A., Khan, S. U., Khan, H. U., & Ilyas, M. (2021). Systematic Mapping Study on Security Approaches in Secure Software Engineering. *IEEE Access*, *9*, 19139–19160. https://doi.org/10.1109/access.2021.3052311