DOI:
УДК 004.056.57
І.І. Жульковська, к.т.н., доцент, *inivzh@gmail.com*
А.В. Плужник, магістр, *andrew.pluzhnik@gmail.com*
О.А. Жульковський, к.т.н., доцент, *olalzh@ukr.net*
Дніпровський державний технічний університет, м. Кам'янське

СУЧАСНІ МЕТОДИ ВИЯВЛЕННЯ ШКІДЛИВИХ ПРОГРАМ

Виявлення та класифікація шкідливих програм стала однією з найважливіших проблеми в галузі кібербезпеки. Існує достатньо методик виявлення невідомого шкідливого програмного забезпечення, кожна з яких має свої переваги, недоліки та особливості використання.

В роботі виконано дослідження сигнатурного та евристичного методів виявлення шкідливого програмного забезпечення. Окремий аналіз присвячений застосуванню методів машинного навчання для класифікації шкідливих програм. Досліджено різні техніки машинного навчання для класифікації та виявлення зразків шкідливих програм та їх відповідних класів, їх фільтрації. Показано корисність графічної візуалізації байтів для виявлення шаблонів проектування програмного забезпечення для подальшої автоматизації виявлення вірусів. Виконано порівняльну характеристику сучасних, головним чином, евристичних методів виявлення шкідливого програмного забезпечення та систематизовано за значеннями точності пошуку.

Оскільки на даний момент не існує ефективної методики виявлення невідомого шкідливого програмного забезпечення, тому для проведення ефективного пошуку і знищення шкідливих програм потрібно комбінувати всі сучасні методи, способи і засоби, враховуючи всі особливості їх використання.

***Ключові слова**: шкідливе програмне забезпечення, сигнатура, статичний аналіз, динамічний аналіз, гібридні підходи, візуалізація шкідливих програм.*

Malware detection and classification has become one of the most important problems in the field of cybersecurity. There are many methods of detecting unknown malware, each of which has its advantages, disadvantages and features of use.

The research of signature and heuristic methods of malware detection is performed in the work. A separate analysis is devoted to the application of machine learning methods for the classification of malicious programs. Various machine learning techniques for classification and detection of samples of malicious programs and their corresponding classes, their filtration have been studied. The usefulness of graphical byte visualization for software design templates for further automation of virus detection is shown. A comparative description of modern, mainly heuristic methods of detecting malware is performed and systematized according to the values of search accuracy.

As there is currently no effective method of detecting unknown malware, so to effectively search and destroy malware you need to combine all modern methods, techniques and tools, taking into account all the features of their use.

***Keywords**: malware, signature, static analysis, dynamic analysis, hybrid approaches, malware visualization.*

Постановка проблеми

Інтернет став невід'ємною частиною нашого повсякденного життя. За оцінками ITU (International Telecommunication Union — міжнародна організація, що визначає рекомендації у галузі телекомунікацій та радіо), наприкінці 2019 р. 53,6 % світового населення, або 4,1 млрд людей, користуються мережею Internet. Її використовують для банківської діяльності, спілкування, розваг, покупок та інших комерційних та некомерційних видів діяльності.

Незважаючи на те, що Internet робить життя зручним, він також надав нові можливості для шахрайства. Зловмисники використовують шкідливі програми для здійснення фінансових

шахрайств або крадіжки приватної/конфіденціальної інформації користувачів. Кількість зареєстрованих атак щороку збільшується.

Антивірусна лабораторія PandaLabs компанії Panda Security зареєструвала і проаналізувала 14,9 млн зразків шкідливого ПЗ у 2019 р. Те, що почалося як хобі технічних ентузіастів та дослідників, перетворилося в міжнародне співтовариство висококваліфікованих програмістів, мотивованих легким прибутком. Зараз це — багатомільйонна галузь, де продаються та купуються хакерські інструменти, яка працює як і легальна індустрія програмного забезпечення (ПЗ).

Поширення шкідливих програм із постійно зростаючою швидкістю представляє серйозну загрозу у пост Internet-світі. Виявлення та класифікація шкідливих програм стала однією з найважливіших проблем у галузі кібербезпеки. Із постійно зростаючим ризиком «нападу», тягар лежить на дослідниках безпеки щодо розробки нових методів виявлення шкідливого ПЗ та розробки нових механізмів протидії останньому.

Аналіз останніх досліджень та публікацій

Проблемам виявлення шкідливого ПЗ та захисту від нього присвячено ряд робіт [1—16]. В цих працях достатньо повно аналізуються певні напрямки, методи або засоби протидії шкідливим програмам. Як показує даний аналіз, на теперішній момент не існує такої методики, яка б повністю вирішувала завдання виявлення невідомого шкідливого ПЗ з прийнятною ефективністю для будь-яких видів і за будь-яких вимог до системи.

Формулювання мети дослідження

Існує безліч методик виявлення невідомого шкідливого ПЗ. Кожна з них має свої переваги, недоліки та особливості використання. Проте на даний момент не існує методики, яка б повністю вирішувала всі завдання виявлення шкідливих програм з прийнятною ефективністю. Отже актуальним напрямком є дослідження сучасних методів виявлення таких програм.

Метою даної роботи є аналіз і систематизація основних методів виявлення шкідливого ПЗ та дослідження особливостей їх застосування.

Виклад основного матеріалу

Шкідливе ПЗ використовується для нападу на критичну інфраструктуру для шпигунства, крадіжки приватної інформації або для здійснення фінансових шахрайств.

На сьогодні для виявлення шкідливих програм переважно використовують підхід на основі сигнатури або підхід, заснований на аномаліях.

Методи, що базуються на сигнатурах, протягом десятиліть активно використовувались для антивірусного ПЗ. Сигнатура — це унікальна послідовність байтів, що є у шкідливому двійковому файлі та у файлах, які були пошкоджені цим шкідливим ПЗ. Сигнатура повинна містити тільки унікальні рядки з цього файлу, настільки характерні, щоб гарантувати мінімальну можливість неправильного спрацьовування — це головний пріоритет будь-якої антивірусної компанії.

Цей підхід швидкий і має високу точність. Але при всій своїй простоті має наступні значні недоліки.

1. Захист відбувається лише від відомих вірусів. Тут слід зазначити, що сигнатури, зазвичай, створюються так, аби покривати не один, а якомога більшу кількість вірусів — сімейство вірусів. Однак завжди існує така зміна виконавчого файлу, при якому сигнатура перестає виявлятися.

2. Постійне зростання бази даних сигнатур. Зі збільшенням кількості вірусів, їх типів, а також здатність вірусів змінюватися збільшується і швидкість наповнення бази.

3. При появі вірусу і до оновлення бази сигнатур клієнт вразливий для нової шкідливої програми. Тільки визначивши досліджуваний файл як вірус можна отримати його сигнатуру і додати в базу. Більш того, розробники шкідливих програм навчилися успішно обходити пошук сигнатур за допомогою обфускації тіла вірусу. Це змусило антивірусні компанії розробляти альтернативні способи захисту.

При підході виявлення аномалій розробниками антивірусного рішення формуються база даних дій, які вважаються безпечними. Якщо процес виконання програми порушує будь-який із цих заздалегідь визначених правил, він позначається як шкідливий. Хоча за допомогою методу,

заснованого на аномаліях, є можливість для виявлення нових зразків шкідливих програм, натомість частота помилкових викликів є дуже висока.

Використовуються у багатьох антивірусних пакетах алгоритми аналізу послідовності команд з метою формування деякої статистики та прийняття рішень про можливість зараження для кожного об'єкта, що перевіряється. Вони отримали назву методів евристичного сканування. Причому, на відміну від сигнатурного методу, евристичний підхід може детектувати як відомі, так і невідомі віруси (тобто ті, що створені після евристичної обробки).

У цьому підході аналітики застосовують методи машинного навчання для класифікації шкідливих програм. Статичні, динамічні, візуальні представлення елементів, або їх комбінації використовуються для навчання класифікатора (classifier) на наборі даних, що складається як із шкідливих, так і з нешкідливих двійкових файлів. Різні техніки машинного навчання, такі як метод опорних векторів (support vector machine, SVM), випадкові ліси (random forest, RF), дерева ухвалення рішень (decision trees, DT), наївний баєсів класифікатор (naive bayes, NB), k-найближчих сусідів (k-nearest neighbor, kNN) та посилення градієнта пропонуються для класифікації та виявлення зразків шкідливого ПЗ та їх відповідних класів, для фільтрації шкідливого ПЗ, яке вимагає подальшого дослідження аналітиком.

Статичний аналіз (StaticApproach) — це процес аналізу бінарного шкідливого ПЗ без фактичного запуску коду. Шаблони, виявлені при такому аналізі, включають рядки сигнатур, розподіл частоти послідовностей байтів або опкодів (opcode — operation code), n-грами на рівні байт або n-грами на рівні опкодів, виклики API (Application Programming Interface), структуру дизасембльованої програми тощо.

Далі розглянуто дослідження, в яких використовувались підходи до статичного аналізу шкідливих програм.

У роботі [1] вилучали інформацію з виконуваних файлів, таку як список DLL (Dynamic Link Library), що використовуються всередині виконуваного файлу; список системних викликів DLL; кількість різних системних викликів всередині кожної DLL, символи або рядки, закодовані у двійковому файлі, використовуючи шістнадцятковий дамп, як ознаки. Для класифікації використовували NB з набором даних для навчання, що складався із 4266 файлів, із яких 3265 вірусів та 1001 чистих зразків. Результатом роботи, як зазначено авторами, була отримана точність 97,11 %. Це одна із перших спроб виконати аналіз шкідливого ПЗ із використанням методів глибинного аналізу даних.

У [2] була досліджена точність класифікації різних методів машинного навчання, таких як NB, SVM, DT та їх підсилені версії, для класифікації шкідливих програм у різних сімействах із використанням особливостей, запропонованих у роботі [1]. При цьому кращою виявилася DT (C4.5 реалізація), яка показала результат в 99,6 % точності с 2,7 % помилкових спрацьовувань.

У [3] представили новий підхід використання послідовності інструкцій змінної довжини для виявлення «червів» у двійкових файлах за допомогою машинного навчання. Вони використовували алгоритми RF та DT для класифікації набору даних, що складається з 1444 «червів» та 1330 чистих файлів і домоглися точності класифікації 96 %.

У [4] досліджували використання послідовності викликів API для виявлення шкідливого ПЗ. Вони показали, що всі версії однієї і тієї ж шкідливої програми мають спільну основну сигнатуру, яку можна визначити за допомогою базової послідовності викликів API.

У своєму дослідженні у [5] запропонували схему, засновану на техніці обфускації, для вивчення недоліків підходів статичного аналізу. Експерименти показали, що статичний підхід не є достатнім для ефективного аналізу шкідливих програм. Оскільки статичного аналізу можна легко уникнути, якщо шкідливе ПЗ обфусковано або стиснуто. Тому потрібно звертати увагу на особливості поведінки для кращого аналізу.

Загальним для всіх підходів на основі динамічного аналізу (Dynamic Approach) є виконання шкідливого програмного зразка у контрольованому середовищі для вилучення особливостей поведінки (віртуальна машина, емулятор, пісочниця тощо). Моніторинг поведінки здійснюється за допомогою таких інструментів, як Process Monitor, Process Explorer, Wireshark або Capture BAT.

У дослідженні [6] запропоновано комплексний підхід до проведення аналізу шкідливих програм на основі поведінки та класифікація шкідливого ПЗ на нові групи за допомогою методів штучного інтелекту. Було використано honeypot (ресурс, що є приманкою для зловмисників) та системи виявлення вторгнень, такі, наприклад, як HoneyClients та Amun для збору зразків шкідливого ПЗ. Далі було створено звіт про поведінку для кожного зразка за допомогою платформ віртуальних машин типу CWSandbox та Anubis, а також кожен звіт було проаналізовано «вручну». За допомогою методів штучного інтелекту зразки шкідливого ПЗ були систематизовано у групи — «черви» та «трояни». Головним недоліком дослідження є те, що не було автоматизовано аналіз звітів. Тому, враховуючи величезний обсяг шкідливих програм, що генеруються в наш час, аналіз звітів «вручну» є неможливим.

У [7] запропонували метод автоматизованої ідентифікації нових класів шкідливих програм із подібним типом поведінки (кластеризація) та класифікацію раніше небачених шкідливих програм для виявлених класів (класифікація) за допомогою машинного навчання. Використовуючи кластеризацію та класифікацію, застосовується новий підхід для обробки поведінки великої кількості двійкових файлів шкідливого ПЗ. Цей підхід істотно скоротив час виконання методів аналізу. Дослідники фіксували особливості зміни стану, такі як відкриття файлу, блокування мьютексу, мережева активність, зараження запущених процесів або встановлення ключа реєстру, а далі відображали поведінку шкідливого ПЗ у багатовимірному векторному просторі. В експерименті було використано понад 10 тис зразків шкідливих програм, що належать до 14 різних сімейств. Ці зразки шкідливого ПЗ були зібрані з використанням honeypots та спам-пастки. В результаті була отримана точність 88 % класифікації сімейств. Недоліком роботи є те, що було розглянуто лише один шлях виконання двійкового файлу в аналізі.

З наведених вище робіт видно, що окремого статичного або динамічного аналізу недостатньо для точної та ефективної класифікації зразків шкідливих програм. Це відбувається тому, що окреме використання цих методів достатньо просто обійти, використовуючи обфускацію коду або різні техніки зупинення при виконанні. Також динамічним аналізом неможливо дослідити всі шляхи виконання програмного файлу. Контрольоване середовище, в якому здійснюється моніторинг шкідливого ПЗ, відрізняється від реального, програма може поводитися по-різному, тому що деяка поведінка шкідливого ПЗ може запускатися лише за певних умов, наприклад, за допомогою певної команди або на конкретну дату системи і, як наслідок, неможливо виявити у віртуальному середовищі.

Тому були створені гібридні підходи (Hybrid Approach), які поєднують статичні та динамічні підходи одночасно для поліпшення виявлення та більшої точності класифікації шкідливих програм.

У роботі [8] запропоновано новий підхід для підготовки класифікатора шкідливих програм, використавши об’єднано як статистичні так і динамічні методи, під назвою OPEM. В результаті навчання моделі було виявлено, що статистичний та динамічний аналіз разом працюють краще, ніж окремо. Досліди проводилися за допомогою двох різних наборів даних та багатьох алгоритмів машинного навчання, а саме DT, kNN, байєсівською мережею та SVM. Результати порівняння точності представлені у табл. 1.

У [9] також запропоновано гібридну модель класифікації бінарних файлів на чисті та шкідливі, в якій інтегруються як динамічні, так і статичні функції аналізу. Для цього було виділено статичну інформацію, таку як «printableString» (тип рядка з обмеженими символами у нотації ASN) та частоту довжини функцій і динамічну інформацію — параметри API та назви функцій API. Для тестування моделі було використано 2939 шкідливих та 541 чистих зразків. Далі, за допомогою інтегрованих мета-класифікаторів, таких як SVM, IB1, DT та RF, була виконана класифікація зразків шкідливого ПЗ з точністю 97,055 %.

Як можна побачити, гібридні підходи дуже перспективні, оскільки забезпечують значне покращення у порівнянні з окремо статичним або динамічним підходом.

Ще одним напрямком є підходи до аналізу шкідливих програм, що засновані на візуалізації.

Для візуалізації та редагування двійкового файлу було доступно кілька інструментів, які відображають файл у шістнадцятковому та ASCII-форматах, але не передають жодної структурної інформації для аналітика.

Таблиця 1. Точність результатів (%) при дослідженні сучасних підходів аналізу шкідливого ПЗ

Classifier	Static Approach	Dynamic Approach	Hybrid Approach
kNN k=1	94.83.	77.19	96.2
kNN k=2	93.15	76.72	95.36
kNN k=3	94.16	76.68	94.63
kNN k=4	93.89	76.58	94.46
kNN k=5	93.50	76.35	93.68
kNN k=6	93.38	76.34	93.5
kNN k=7	92.87	76.33	93.51
kNN k=8	92.89	76.31	93.3
kNN k=9	92.10	76.29	92.94
kNN k=10	92.24	76.24	92.68
DT: J48	92.61	76.72	93.59
DT: Random Forest N=10	95.26	77.12	95.19
SVM: RBF Kernel	91.93	76.75	93.25
SVM: Polynomial Kernel	95.50	76.87	95.99
SVM: Normalised Polynomial Kernel	95.90	77.26	96.60
SVM: Pearson VII Kernel	94.35	77.23	95.56
Naıve Bayes	90.02	74.36	90.11
Bayesian Network: K20	86.73	75.73	87.20
Bayesian Network: Hill Clim	86.73	75.73	87.22
Bayesian Network: TA	93.40	75.47	93.53

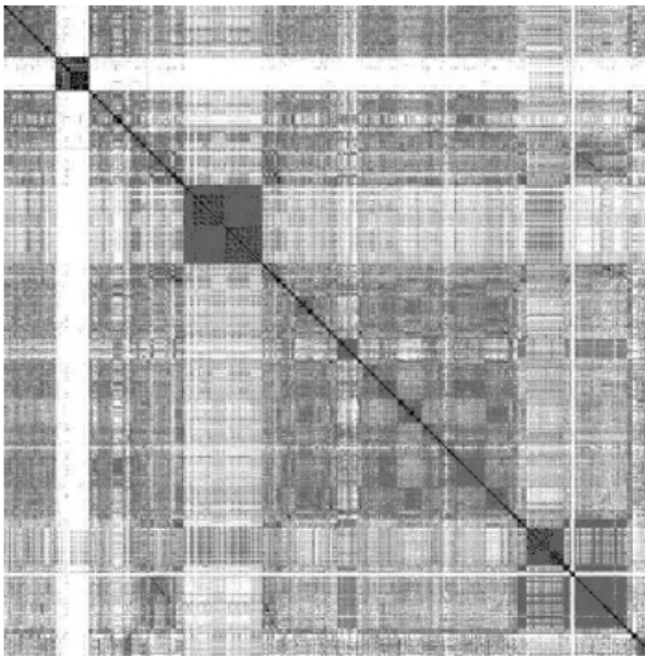

Рис. 1. Візуалізація 2 млн рядків коду мовою С

У роботі [10] застосовано техніку візуалізації даних точковою діаграмою та показано, що візуалізація може бути корисною для виявлення шаблонів проектування ПЗ. Dotplot — це техніка візуалізації патернів програмних продуктів, що дає візуальний огляд структури величезної системи. Такі візерунки корисні для проектування програмних систем через послідовну абстракцію — шаблон проектування, який допомагає усунути надмірність. Послідовність розбивається на лексеми, далі лексеми наносяться на графік точками. Приклад побудованої візуалізації коду представлено на рис. 1.

У роботі [11] намагалися виявити та візуалізувати віруси, що вбудовані всередину виконавчого файлу, за допомогою самоорганізованих карт Кохонена (SOM — тип штучних нейронних мереж, що використовується для візуалізації багатовимірних даних) [12], без використання інформації про сигнатуру. У дослідженні було виявлено, що кожне сімейство вірусів має свою маску, яку можна побачити при візуалізації двійкового файлу за допомогою SOM (рис. 2, 3).

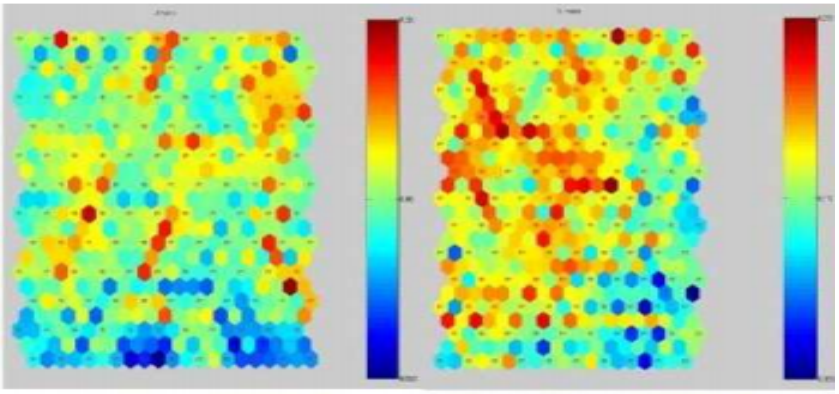

Рис. 2. Візуалізація за допомогою SOM незаражених файлів

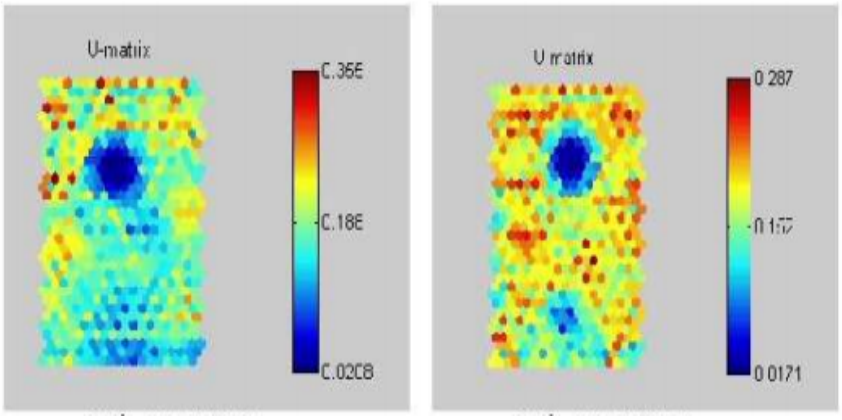

Рис. 3. Візуалізація за допомогою SOM заражених файлів

У [13] вперше дослідили використання графічної візуалізації байтів для автоматичної класифікації шкідливих програм. У ході дослідження весь зразок шкідливого ПЗ конвертували у зображення відтінків сірого. Набір даних дослідження складався із 9458 зразків шкідливого ПЗ, що належать до 25 різних класів, зібраних із системи Anubis [14]. Далі на цих зображеннях тренувалася модель kNN; для оцінки відстані використовувалася евклідова відстань. Зразки шкідливих програм було згруповано у відповідні класи та отримано точність 97,18 %.

У роботі [15] також конвертували шкідливе ПЗ у двовимірне зображення у градаціях сірого та класифікували зразки на основі отриманих текстур. Було виділено загальні риси на основі текстур, використовуючи вейвлет-перетворення Габора та дескриптор зображень GIST. Набір даних експерименту складався із 3131 зразків двійкових файлів із 24 унікальних сімейств шкідливих програм. Після побудови вектора ознак класифікація проводиться на зразках шкідливих програм на основі техніки машинного навчання SVM. В результаті була отримана точність 96,35 %.

У роботі [16] запропоновано поетапний підхід для автоматичного віднесення шкідливих програм до різних сімейств та виявлення нових шкідливих програми. В досліджені використовувалося комбінацію перетворення байтів у зображення у градаціях сірого, n-грам операційного коду та функції імпорту. Модуль прийняття рішень використовує ці особливості для класифікації зразків шкідливого ПЗ за їх відповідними сімействами та для ідентифікації нових невідомих шкідливих програм. Для виявлення нових сімейств шкідливих програм було застосовано алгоритм SNN (Shared Nearest Neighbour). Модель навчалася на базі даних, що складається із 21740 зразків шкідливих програм із 9 різних сімейств. У результаті була отримана точність класифікації 98,9 %, а виявлення 86,7 %.

Висновки

В роботі виконано дослідження сигнатурного та евристичного методів виявлення шкідливого програмного забезпечення. Окремий аналіз присвячений застосуванню методів машинного навчання для класифікації шкідливих програм. Досліджено різні техніки машинного навчання для класифікації та виявлення зразків шкідливих програм та їх відповідних класів, їх фільтрації. Показано корисність графічної візуалізації байтів для виявлення шаблонів проектування програмного забезпечення для подальшої автоматизації виявлення вірусів. Виконано порівняльну характеристику сучасних, головним чином, евристичних методів виявлення шкідливого програмного забезпечення та систематизовано за значеннями точності пошуку.

Оскільки на даний момент не існує ефективної методики виявлення невідомого шкідливого програмного забезпечення, тому для проведення ефективного пошуку і знищення шкідливих програм потрібно комбінувати всі сучасні методи, способи і засоби, враховуючи всі особливості їх використання.

Результати роботи може бути використано для розробки програмних засобів ефективного виявлення та знищення шкідливого програмного забезпечення з застосуванням штучних нейронних мереж і їх навчанням.

Список використаної літератури

1. Schultz, M. G., Eskin, E., and Zadok, F. Data Mining Methods for Detection of New Malicious Executables. In Proc. of the 22nd IEEE Symposium on Security and Privacy, 2001.
2. Kolter, J. and Maloof, M. Learning to detect malicious executables in the wild. In Proc. of the 10th ACM Int. Conf. on Knowledge Discovery and Data Mining, 2004.
3. Siddiqui, M., Wang, M. C. and Lee, J. Detecting Internet Worms Using Data Mining Techniques. In Journal of Systemics, Cybernetics and Informatics, 2009.
4. Sung, A., Xu, J., Chavez, P. & Mukkamala, S. Static Analyzer of Vicious Executables (SAVE). In Proc. of the 20th Annual Computer Security Applications, 2004.
5. Moser, A., Kruegel, C., and Kirda, E. Limits of Static Analysis for Malware Detection. In IEEE Computer Society, 2007.
6. Peter, E. and Schiller, T. A practical guide to honeypots, 2008. URL: http://www.cs.wustl.edu/~jain/cse571-09/ftp/honey.pdf.
7. Rieck, K., Holz, T., Willems, C., Dussel, P., and Laskov, P. Learning and classification of Malware behaviour. In Detection of Intrusions and Malware, and Vulnerability Assessment, 2008.
8. Santos, I., Devesa, J., Brezo, F., Nieves, J., and Bringas, P. G. OPEM: A Static-Dynamic Approach for Machine-Learning-Based Malware Detection, 2012.
9. Islam, R., Tian, R., Batten, L. M., and Versteeg, S. Classification of malware based on integrated static and dynamic features. In Journal of Network and Computer Applications, 2013.
10. Helfman, J. Dotplot patterns: A literal look at pattern languages. TAPOS, 1995, 2:31–41.
11. Yoo, I. S. Visualizing windows executable virus using self-organizing maps. In Proc. of ACM workshop on Visualization and data mining for computer security, 2004.
12. Kohonen, T. Self-Organizing Maps. Springer, 1995.
13. Nataraj, L., Karthikeyan, S., Jacob, G., and Manjunath, B. Malware Images: Visualization and Automatic Classification. In Proc. of International Symposium on Visualization for Cyber Security, 2011.
14. Kolbitsch, C. Anubis, 2011. URL: https://hybrid-analysis.com/
15. Makandar, A. and Patrot, A. Malware Analysis and Classification using Artificial Neural Network. In Trends in Automation Communications and Computing Technology, 2015.
16. Liu Liu, Bao-sheng Wang, Bo Yu, and Qiu-xi Zhong. Automatic Malware Classification and New Malware Detection using Machine Learning. In Frontiers of Information Technology and Electronic Engineering, 2016.

MODERN METHODS OF DETECTION OF MALWARE
Zulkovska I., Pluzhnik A., Zhulkovski O.

There are many methods of detecting unknown malware, each of which has its advantages, disadvantages and features of use. However, at present there is no methodology that would fully solve all the problems of detecting malware with acceptable efficiency. In the field of modern computer security, most solutions are implemented as a set of several technologies. In this regard, the study of modern methods of detecting malware is an important area. The purpose of this work is to analyze and systematize the main methods of detecting malicious software and study the features of their use.

The research of signature and heuristic methods of malware detection is performed in the work. A separate analysis is devoted to the application of machine learning methods (static, dynamic, visual representation of elements and their combinations) for the classification of malicious programs.

Various machine learning techniques have been studied: the machine method of reference vector, random forest, decision trees, naive bayes, k-nearest neighbor and gradient amplification to classify and detect malware samples and their respective classes, their filtering.

The usefulness of graphical byte visualization for detecting software design templates using Dotplot technique for further automation of virus detection is shown.

The comparative characteristic of modern, mainly, heuristic methods of detection of malware is executed and systematized on values of accuracy of search.

As there is currently no effective method of detecting unknown malware, so to effectively search and destroy malware you need to combine all modern methods, techniques and tools, taking into account all the features of their use.

The results can be used to develop software to effectively detect and destroy malware using artificial neural networks and their training.

References

[1] Schultz, M. G., Eskin, E., & Zadok, F. (2001). Data Mining Methods for Detection of New Malicious Executables. Proceeding of the 22nd IEEE Symposium on Security and Privacy. *IEEE Computer Society*, pp. 38–49.
DOI: https://doi.org/10.1109/SECPRI.2001.924286

[2] Kolter, J. & Maloof, M. (2004). Learning to detect malicious executables in the wild. Proceeding of the 10th ACM Int. Conf. on Knowledge Discovery and Data Mining, pp.470–478.
DOI: https://doi.org/10.1145/1014052.1014105

[3] Siddiqui, M., Wang, M. C. & Lee, J. (2009). Detecting Internet Worms Using Data Mining Techniques. *Journal of Systemics, Cybernetics and Informatics*, 6, pp. 48–53.
DOI: https://doi.org/10.1155/2016/8069672

[4] Sung, A., Xu, J., Chavez, P. & Mukkamala, S. (2004). Static Analyzer of Vicious Executables (SAVE). Proceeding of the 20th Annual Computer Security Applications Conference. *IEEE Computer Society,* pp. 326–334.
DOI: https://doi.org/10.1109/CSAC.2004.37

[5] Moser, A., Kruegel, C., & Kirda, E. (2007). Limits of Static Analysis for Malware Detection. *IEEE Computer Society*, pp. 421-430. DOI: https://doi.org/10.1109/ACSAC.2007.4413008

[6] Peter, E. & Schiller, T. (2008). A practical guide to honeypots.
URL: http://www.cs. wustl.edu/~jain/cse571-09/ftp/honey.pdf

[7] Rieck, K., Holz, T., Willems, C., Dussel, P., & Laskov, P. (2008). Learning and classification of Malware behaviour. Proceedings of the 5th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp.108–125.
DOI: https://doi.org/10.1007/978-3-540-70542-0_6

[8] Santos, I., Devesa, J., Brezo, F., Nieves, J., and Bringas, P. G. (2012). OPEM: A Static-Dynamic Approach for Machine-Learning-Based Malware Detection. Proceedings of the International Joint Conference CISIS'12-ICEUTE12-SOCO12 Special Sessions, 2012, vol. 189, pp. 271–280.
DOI: https://doi.org/10.1007/978-3-642-33018-6_28

[9] Islam, R., Tian, R., Batten, L. M., & Versteeg, S. (2013). Classification of malware based on integrated static and dynamic features. *Journal of Network and Computer Applications*, 36(2), 646–656. DOI: https://doi.org/10.1016/j.jnca.2012.10.004
[10] Helfman, J. Dotplot patterns: A literal look at pattern languages. TAPOS, 1995, 2:31–41.
[11] Yoo, I. S. (2004). Visualizing windows executable virus using self-organizing maps. Proceedings of ACM workshop on Visualization and data mining for computer security, 2004. Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, pp. 82–89. DOI: https://doi.org/10.1145/1029208.1029222
[12] Kohonen, T. Self-Organizing Maps. Springer, 1995.
[13] Nataraj, L., Karthikeyan, S., Jacob, G. & Manjunath, B. (2011). Malware Images: Visualization and Automatic Classification. Proceedings of the 8th International Symposium on Visualization for Cyber Security, *ACM*, pp. 1–7. DOI: https://doi.org/10.1145/2016904.2016908
[14] Kolbitsch, C. (2011). Anubis. URL: https://hybrid-analysis.com/
[15] Makandar, A. & Patrot, A. (2015). Malware Analysis and Classification using Artificial Neural Network. In Trends in Automation Communications and Computing Technology Proceedings International conference on trends in automation, communications and computing technology. *IEEE Computer Society,* pp. 1–6. DOI:https://doi.org/10.1109/ITACT.2015.7492653
[16] Liu, L., Wang, Bs., Yu, B. & Zhong, Qx. (2016). Automatic Malware Classification and New Malware Detection using Machine Learning. *Frontiers of Information Technology and Electronic Engineering*, 18**,** pp. 1336–1347. DOI: https://doi.org/10.1631/FITEE.1601325