УДК 004.732.056 **doi: 10.20998/2522-9052.2018.2.17**

С. Ю. Гавриленко [1], В. В. Челак[1], В. А. Василев [2]

[1] Національний технічний університет «Харківський політехнічний інститут», Харків, Україна
[2] Технічний університет – Софія, Софія, Болгарія

СИСТЕМА ІДЕНТИФІКАЦІЇ ШКІДЛИВОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ КОНТЕКСТНО-ВІЛЬНИХ ГРАМАТИК

Предметом статті є дослідження методів ідентифікації шкідливого програмного забезпечення в комп'ютерних системах. **Метою** є дослідження існуючих моделей виявлення вірусів на базі формальних мов та граматик та удосконалення моделі за рахунок використання LL(1)-граматики. **Завдання:** розробити математичну модель ідентифікації шкідливого програмного забезпечення на основі контекстно-вільних граматик; вибрати ефективний алгоритм її роботи, розробити програмну модель та виконати тестування. Використовуваними **методами** є: апарат формальних мов та граматик, математичні моделі на основі детермінованих магазинних автоматів. Отримано такі **результати.** Обґрунтовано вибір типу граматики та моделі магазинного автомату. Розроблено програмне забезпечення, яке генерує функції переходів магазинного автомату відповідно до заданих правил граматики, аналізує вхідний файл на наявність заданих ознак, характерних для шкідливого програмного забезпечення та моделює роботу детермінованого низхідного магазинного автомату. За результатом роботи магазинного автомату формується висновок щодо можливості зараження комп'ютерної системи. **Висновки**. Наукова новизна отриманих результатів полягає в наступному: досліджено існуючі моделі антивірусних сканерів на базі формальних мов та граматик; удосконалено модель за рахунок використання LL(1)-граматики, розроблено програмне забезпечення та виконано тестування. Проведені експериментальні дослідження підтверджують можливість використання запропонованого підходу, як додаткового засобу для виявлення шкідливого програмного забезпечення.

Ключові слова: комп'ютерні системи; антивірусне програмне забезпечення; контекстно-вільні граматики; LL(1) – граматики; низхідний магазинний автомат.

Вступ

Постановка проблеми. Одну з найзначніших загроз безпеці комп'ютерних систем та інформації в цілому складає шкідливе програмне забезпечення, або комп'ютерні віруси.

В 2017 зафіксовано зростання атак з боку сімейства вимагачів сімейства Trojan, особливо для мобільних систем. Кількість атакованих їм користувачів зросла більш ніж в 13 разів [1]. Так хакерська атака, проведена за допомогою вірусу Petya.A, в Україні влітку 2017 року за кілька годин вразила банки, заправки, магазини, сайти державних структур. Паралізованими виявилися навіть сайти Кабінету міністрів і ряду найбільших ЗМІ. Ця шкідлива програма вразила комп'ютери багатьох організацій і приватних осіб в 60 країнах світу. Збитки від атаки вірусу оцінено в 8 млрд доларів [2].

Слід зауважити, що зазначена проблема посилюється динамічним зростанням кількості мобільних пристроїв, загальним переходом на хмарні технології і поширенням Інтернет-технологій, що призводить до зростання кількості шкідливого програмного забезпечення.

Аналіз літератури [3-7], що сучасні антивірусні програми не можуть повністю захистити комп'ютерні систем (КС) і потребують певного часу для виявлення нових версії шкідливого програмного забезпечення. Сучасні евристичні технології не забезпечують належного рівня розпізнавання також при роботі з зашифрованими об'єктами. До недоліків існуючих методів розпізнавання модифікацій шкідливого програмного забезпечення (ШПЗ) також можна віднести високу ймовірність помилкових спрацьовувань.

Основним шляхом усунення зазначених недоліків є вдосконалення моделей виявлення ШПЗ і аргументований вибір вхідних критеріїв оцінювання. Одним із перспективних напрямків евристичного аналізу є використання апарату формальних мов та граматик Формальні мови і граматики застосовуються для опису об'єктів реального світу регулярної структури. Аналіз показав, що використання формальних мов і граматик є найбільш поширеними при захисті комп'ютерних мереж [8] і дозволяє адекватно формалізувати опис атак найкращим чином та побудувати адекватний формальний опис сценаріїв досить складних атак. Крім того, граматики можуть бути використані і в іншій ролі: вони можуть використовуватися при розпізнаванні атак, якщо її розглядати як завдання синтаксичного аналізу ланцюжків відомої структури

Відомо, що у машинному коді, завжди присутні певні закономірності, виявлення яких може принести користь при детектуванні вірусів Фактично, розташовані в певному порядку події є фразами певної мови. Саме ці закономірності можуть бути використані для виявлення вірусів та їх модифікацій.

У роботах [9-12].наведено тільки опис формальних моделей, що лягли в основу методу виявлення шкідливого програмного коду та відсутні дані щодо вибору типу граматики та їх реалізації.

Метою статті є дослідження існуючих моделей вірусів на базі формальних мов та граматик та удосконалення моделі за рахунок використання LL(n)-граматики

Результати розробки та досліджень

Відомо що є чотири типи формальних мов [15]. Доведено, що для кожного з цих типів мов існує свій

тип распізнавача з певним складом компонентів і, отже, із заданою складністю алгоритму роботи. Але найбільший інтерес представляє контестно-вільна граматика (КВ). Серед всіх КВ-мов можна виділити клас детермінованих КВ-мов, розпізнавачами для яких є детерміновані автомати з магазинною (стековою) зовнішньою пам'яттю. Ці мови мають властивість однозначності. Доведено, що для будь-якого детермінованої КВ-мови завжди можна побудувати однозначну граматику [16, 17]. Крім того, для таких мов існує алгоритм роботи розпізнавача з квадратичною складністю. Оскільки ці мови є однозначними, саме вони надалі будуть використані для вирішення проблеми ідентифікації шкідливого програмного забезпечення. Окремим випадком КВ-мови є автоматні граматики для яких розпізнавачем є односторонній недетермінований автомат без зовнішньої пам'яті, який передбачає лінійну залежність часу на розбір вхідного ланцюжка від її довжини. Даний тип розпізнавача може бути використаним лише за умови опису структури з використанням автоматної граматики, що для опису шкідливого програмного забезпечення можливо тільки в окремих випадках.

В даній роботі для подальшого аналізу використано підклас КВ-граматики, а саме LL(1)-граматику, для якої детермінований магазинний автомат М працює по одному вхідному символу, розташованому в поточній позиції. Відомо, що спадний розпізнавач на основі LL(1)-граматики, є більш наглядним і простим в реалізації [15-17],

Магазинний автомат М, визначається наступною сукупністю семи об'єктів [15]:

$$M = \{ P, S, s_o, f, F, H, h_o \}, \quad (1)$$

де P – вхідний алфавіт, S – алфавіт станів, s_o – початковий стан, $s_o \in S$, F – множина кінцевих станів , H – алфавіт магазинних символів, h_o – маркер дна магазину, $h_o \in H, f$ – функція переходів.

Робота автомата може бути подана як зміна конфігурацій:

$$(s, a\alpha, \gamma h) \vdash (s', \alpha, \gamma\beta). \quad (2)$$

Для кожної LL(1) граматики можна побудувати детермінований магазинний автомат М, що допускає мову, породжувану даною граматикою:

$$L(Г) = L(M). \quad (3)$$

Для вибору вхідних критеріїв системи ідентифікації шкідливого програмного забезпечення проаналізовано ШПЗ сімейства Trojan, Worm та Adware та виділено характерні ознаки.

Отримано, що сімейство Trojan-Ransom. AndroidOS.Egat після установки перевіряє, що воно запущено на цьому пристрої, а не на віртуальній машині. Якщо перевірка пройдена, з віддаленого сервера завантажується основний модуль, який, експлуатуючи уразливості в системі, намагається отримати права суперкористувача. Якщо це вдається, то вірус встановлює свої модулі в системні папки, а також модифікує налаштування пристрою таким чином, щоб залишитися в ньому навіть після скидання до заводських налаштувань. Шкідливе програмне забезпечення характеризується стандартною для вимагача функціональністю: блокує роботу пристрою, перекриваючи своїм вікном дисплей та вимагає гроші за розблокування.

Після запуску на комп'ютері троянська програма Trojan.Stoldt.Win32 копіює своє тіло в системну папку Windows або в папку Temp (використовуючи API-функції CreateFile, CloseFile, CloseHandle): C:\DOCUME~1\ User\LOCALS~1\Temp\haoxy.exe. Після чого реєструється в ключах автозапуску системного реєстру C:[HKCU\Software\Microsoft\ Windows\ Currentversion\Run]usbcommonide = C:\DOCUME~1\User\LOCALS~1\Temp\haoxy.exe. Зареєстровані спроби троянів даного сімейства завантажити шкідливе ПО з віддалених серверів:

– http://www.ananwg.com/
– www.dagewozhishihunkoufanchininxingxingh aobuyaodawo555555.com
– http://www.haoxiaoyao.com/
– cnc.haoxiaoyao.com.

За результатами аналізу можливо відмітити, що сімейство потенційно небажаних програм (Adware) для 64х бітових систем Adware.SearchSuite.Win64, які уповільнюють роботу web-браузерів і показують спливаючу рекламу на кожній сторінці та при проникненні на комп'ютер, інсталюються в наступні директорії (використовуючи API-функції CreateFile, CloseFile, CloseHandle):

– C:\Program Files (x86) \Movies App\ Datamngr.
– C:\Program Files\Movies App\Datamngr.
– C:\Program Files\jZip\.
– C:\Documents and Settings\test user\Local Settings\Temp.

Для автоматичного запуску при перезавантаженні комп'ютера, Adware. SearchSuite реєструється в реєстрі як сервіс під ім'ям Datamngr (можуть бути і інші назви): HKLM\SYSTEM\CurrentControlSet\ services \DatamngrCoordinator\ImagePath : "C:\Program Files\ Movies App \Datamngr\ Datamngr Coordinator.exe.

Аналіз механізму роботи кріптолокерів сімейства черв'яка-шифрувальника Trojan. Encoder.12544 (відомого як вірус Petya) показав, що він шифрує файли з певними розширеннями, а також перезаписує MBR (Master Boot Record), очищає лог-файли (журнали подій), знаходить велику кількість різнотипних файлів (рис. 1), шифрує їх виконує перезавантаження комп'ютера та виводить повідомлення з вимогою викупу.

Зразок отримує аутентифікаційні дані за допомогою функції CredEnumerate і утиліти mimikatz. За допомогою отриманих даних виконується поширення мережею за рахунок підключень до ресурсу admin$, утиліти PsExec.exe і wmic.exe (WMI). Також виконуються спроби експлуатації вразливостей SMB EternalBlue (CVE-2017-0144) і EternalRomance (CVE-2017-0145).

Отримання аутентифікаційних даних виконується таким чином: вірус виконує спроби отримання аутентифікаційних даних за допомогою функції CredEnumerate (виконується пошук даних, ім'я яких починається з «TERMSRV/»).

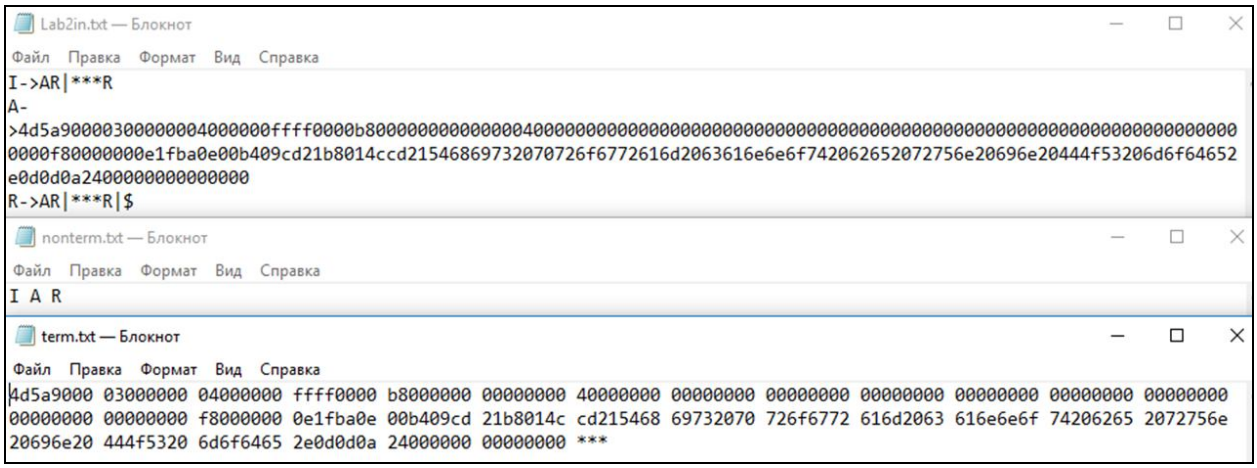

Рис. 1. Набір правил граматики для перевірки заголовку файлу

Отримані дані використовуються для поширення мережею. Вірус використовує функції системи (такі як GetExtendedTcpTable, GetIpNetTable, NetServerEnum, WNetEnumResource, DhcpEnum Subnets, DhcpEnumSubnetClients) для формування адреси ресурсів мережі.

Як результат, для подальшого дослідження, виділено перелік наступних дій, які у сукупності є шкідливими і подаються на вхід системи ідентифікації шкідливого програмного забезпечення а саме:

– Перевірка послідовностей на наявність хеш-сум назв антивірусного ПО.
– Звертання до великої кількості різнотипних файлів.
– Використання алгоритму шифрування за допомогою відкритого ключа.
– Використання утиліти mimikatz для вилучення облікових даних.
– Спроби отримати права адміністратора, підключення до різних ресурсів.
– Використання функцій отримання інформації про ресурси мережі.
– Спроба використання вразливостей SMB-механізму.
– Спроба отримати права адміністратора
– Спроба перезапису MBR.
– Перезавантаження комп'ютера.

Робота системи базується на множині низхідних магазинних автоматів, які аналізують вхідний файл. Система ідентифікації шкідливого програмного забезпечення працює наступним чином. Спочатку виконується перевірка, чи є вхідний файл виконуваним, тобто система буде аналізувати тільки файли, що виконуються (один із 11 вхідних критеріїв). Надалі, програма зчитує із файлу правила граматики (рис. 1, 2), знаходить функції First() та Follow(), формує множину елементів Choice() та генерує функції переходів магазинного автомата [11-13].

Очікувані функції переходів магазинного автомату для граматики наведено в табл. 1, а згенеровані програмою – на рис. 3. Для заданого класу шкідливого програмного забезпечення, на базі згенерованих функцій переходів магазинних автоматів виконується перевірка вхідного файлу на наявність вибраних ознак. Система ідентифікації ШПЗ дозволяє задавати кількість знайдених ознаках, які формують рішення про можливість враження комп'ютерної системи. Проведено тестування системи на множині безпечних файлів та файлів які містять шкідливе програмне забезпечення. На рис. 4 наведено результат тестування шкідливого файлу, у якому знайдено більше ніж 6 ознак із 11 заданих. Як видно із результатів експерименту система ідентифікації шкідливого програмного забезпечення на основі LL(1) - граматики може бути зручним інструментом для аналізу файлів,. Її програмна реалізація не є складною, але потребує аргументованого вибору вхідних критеріїв.

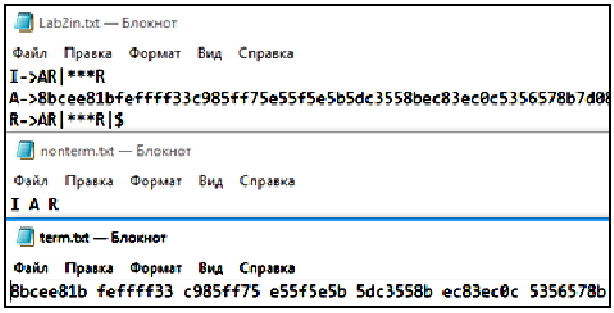

Рис. 2. Набір правил граматики для фіксації спроби перезапису MBR

Таблиця 1. **Функції переходів магазинного автомату**

1. f*(*s*, 8bcee81b, I) = (*s*, RA)	9. f(*s*, 5356578b, 5356578b) = (*s*, $)
2. f(*s*,***,I) = (*s*, R)	10. f(*s*, 7d086a01, 7d086a01) = (*s*, $)
3. f(*s*, 8bcee81b, A) = (*s*, 00578b4f 68b01f437d086a015356578bec83ec0c 5dc3558be55f5e5bc985ff75feffff33)	11. f(*s*, 68b01f43, 68b01f43) = (*s*, $)
4. f(*s*, feffff33, feffff33) = (*s*, $)	12. f(*s*, 00578b4f, 00578b4f) = (*s*, $)
5. f(*s*, c985ff75, c985ff75) = (*s*, $)	13. f*(*s*, 8bcee81b, R) = (*s*, RA)
6. f(*s*, e55f5e5b, e55f5e5b) = (*s*, $)	14. f(*s*, ***, R) = (*s*, R)
7. f(*s*, 5dc3558b, 5dc3558b) = (*s*, $)	15. f*(*s*, $, R) = (*s*, $)
8. f(*s*, ec83ec0c, ec83ec0c) = (*s*, $)	16. f*(*s*, $, h_0) = (*s*, $)

Висновки

В роботі виконано дослідження існуючих моделей виявлення шкідливого програмного забезпечення на базі формальних мов та граматик.

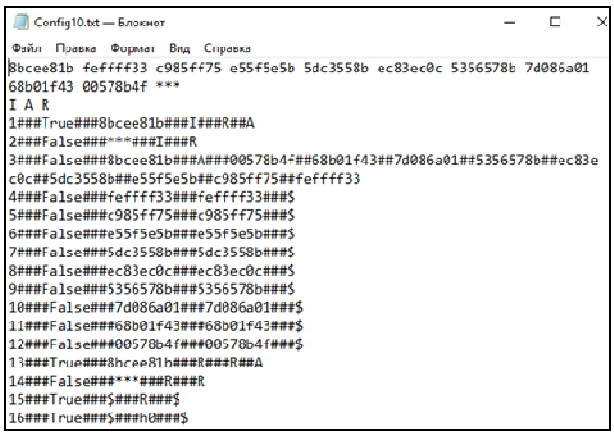

Рис. 3. Набір згенерованих програмою функцій переходів магазинного автомату для перевірки наявності функції перезапуску комп'ютера

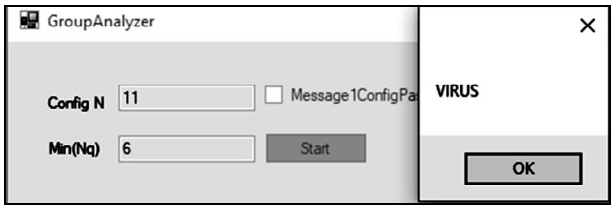

Рис. 4. Результати роботи система ідентифікації шкідливого програмного забезпечення на основі контекстно-вільних граматик

Обгрунтовано вибір типу граматики та моделі магазинного автомату.

Удосконалено існуючі моделі виявлення ШПЗ за рахунок використання LL(1)-граматики, яка передбачає лінійну залежність часу на розбір вхідного ланцюжка від її довжини.

Розроблено програмну модель системи ідентифікації ШПЗ.

Обгрунтовано вибір вхідних критеріїв для тестової системи, виконано тестування.

Пошук кожного із заданих критеріїв, які задаються правилами граматики та зберігаються в файлі, базується на роботі низхідного детермінованого магазинного автомату. Система на базі правил граматики знаходить функції First() та Follow(), формує множину елементів Choice() та генерує функцій переходів магазинного автомата. Використовуючи отримані функцій переходів магазинного автомату, сканується вхідний файл та будується модель низхідний магазинного автомату. Результатом роботи системи ідентифікації шкідливого програмного забезпечення є висновок про можливість враження комп'ютерної системи.

Проведені експериментальні дослідження підтверджують можливість використання запропонованого підходу, як додаткового засобу для виявлення шкідливого програмного забезпечення.

СПИСОК ЛІТЕРАТУРИ

1. Развитие информационных угроз в первом квартале 2017 года [Електронний ресурс] / Р. Унечек, Ф. Синицын, Д. Паринов, В. Столяров. – Режим доступу: https://securelist.ru/analysis/malware-quarterly/30657/it-threat-evolution-q1-2017-statistics.
2. Кіберексперт оцінив збитки від вірусу Petya у світі [Електронний ресурс]. – Режим доступу: https://tsn.ua/svit/kiberekspert-ociniv-zbitki-vid-virusu-petya-a-u-sviti-953633.html.
3. Шелухин О. И. Обнаружение вторжений в компьютерные сети / О. И. Шелухин, Д. Ж Сакалема, А. С. Филинова. – М.: Горячая линия-Телеком, 2013. – 220 с.
4. Гошко С. В. Технологии борьбы с компьютерными вирусами / С. В. Гошко. – М.: Солон-Пресс, 2009. – 352 с.
5. Касперский К. Записки исследователя компьютерных вирусов / К. Касперский. – СПб.: Питер, 2012. – 316 с.
6. Касперский К. Компьютерные вирусы изнутри и снаружи / К. Касперский. – СПб.: Питер, 2011. – 527 с.
7. Семенов С. Г. Защита данных в компьютеризированных управляющих системах (монография) / С. Г. Семенов, В. В. Давыдов, С. Ю. Гавриленко. – LAP LAMBERT ACADEMIC PUBLISHING, Germany, 2014.– 236 с.
8. Формальные методы защиты информации в компьютерных сетях [Електронний ресурс]. – Режим доступу: http://docplayer.ru/55189122-Proekt-1994p-formalnye-metody-zashchity-informacii-v-kompyuternyh-setyah.html.
9. Filiol Eric. Metamorphism, Formal Grammars and Undecidable Code Mutation [Електронний ресурс] / Eric Filiol // International Journal of Computer and Information Engineering. – 2007. – Vol. 1, No. 2, , pp. 281-286. – Режим доступу: https://waset.org/publications/1369/metamorphism-formal-grammars-and-undecidable-code-mutation/
10. Котенко И. В. Восстановление формальных грамматик, задающих сценарии компьютерных атак по прецендентам / И. В. Котенко // Искусственный интеллект. – Санкт-Петербург, 2002. – №3. – С. 584-589.
11. Климентьев К. Е. Компьютерные вирусы и антивирусы / К. Е. Климентьев. – М.: ДМК Пресс, 2013. – 656 с.
12. Збицкий П. В. Модель метаморфного преобразования исполняемого кода / П. В. Збицкий // Компьютерные технологии, управление, радиоэлектроника. – Российская федерация, 2009. – Вып. 10. – С. 57-61.
13. Савенко О. С. Модель процесу діагностування комп'ютерних систем на наявність поліморфного та метаморфного програмного коду / О. С. Савенко, С. М. Лисенко, А. О. Нічепорук // Інформаційні технології та комп'ютерна інженерія. – 2014. – Т. 3 [12]. – С. 46-51
14. Нічепорук А. О. Моделі життєвого циклу поліморфних вірусів / А. О. Нічепорук, О. С. Савенко // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – Луцьк, 2013. – Вип. 11. – С. 64-71.
15. Системне програмування. Системні сервісні компоненти / О. С. Дерев'янко, С. Г. Межерицький, С.Ю. Гавриленко, А. М. Клименко. – Х.: НТУ «ХПІ», 2009. – 160 с.
16. Альфред А. Компиляторы. Принципы, технологии, инструменты» / А. Альфред, С. Равви, Д. Ульман. – Москва-Санкт-Петербург-Киев, 2001.– 768 с.
17. Гордеев А. В. Системное программное обеспечение / А. В. Гордеев, А. Ю. Молчанов. – СПб.: Питер, 2002. – 734 с.

REFERENCES

1. Unecek, R., Sinitsyn, F., Parinov, D. and Stolyarov V. (2017), *Development of Information Threats in the First Quarter of 2017. Statistics*, available at: https://securelist.ru/analysis/malware-quarterly/30657/it-threat-evolution-q1-2017-statistics (last accessed February 08, 2018).

2. *Cyberexpert estimated the damage from the Petya virus in the world*, available at: https://tsn.ua/svit/kiberekspert-ociniv-zbitki-vid-virusu-petya-a-u-sviti-953633.htm (last accessed February 08, 2018).
3. Shelukhin, O.I., Sakalema, D.Zh. and Filinova A.S. (2013), *Intrusion Detection into Computer Networks*, Moscow, 220 p.
4. Goshko, S.V. (2009), *Technologies for combating computer viruses*, Solon-Press, Moscow, 352 p.
5. Kaspersky, K. (2012), *Notes of the researcher of computer viruses*, Peter, St. Petersburg., 316 p.
6. Kaspersky, K. (2011), *Computer viruses from the inside and out*, Peter, St. Petersburg., 527 p.
7. Semenov, S.G., Davydov, V.V. and Gavrilenko S.Yu. (2014), *Data Protection in Computerized Control Systems*, LAP LAMBERT ACADEMIC PUBLISHING, Germany, 236 p.
8. Formal methods for protecting information in computer networks, available at: http://docplayer.ru/55189122-Proekt-1994p-formalnye-metody-zashchity-informacii-v-kompyuternyh-setyah.html (last accessed February 08, 2018).
9. Eric Filiol (2007), "Metamorphism, Formal Grammars and Undecidable Code Mutation", *International Journal of Computer and Information Engineering*, Vol.1, No. 2, pp. 281-286, available at: https://waset.org/publications/1369/metamorphism-formal-grammars-and-undecidable-code-mutation (last accessed February 08, 2018).
10. Kotenko, I.V. (2002), "Restoration of formal grammars, setting the scenarios of computer attacks on the pre-dates", *Artificial Intelligence*, Saint-Petersburg, No. 3, pp. 584-589.
11. Klymentyev K.E. (2013), *Computer viruses and antiviruses: a programmer's view*, DMK Press, Moscow, 656 p.
12. Zbitsky, P.V. (2009), Model of metamorphic transformation of executable code, *Computer technologies, management, radio electronics*, Russian Federation, No. 10, pp. 57-61.
13. Savenko, O.S., Lysenko, S.M. and Nichiporuk A.O. (2014), "Model for the process of diagnosing computer systems for the presence of polymorphic and metamorphic code", *Information technology and computer engineering*, Vol. 3 [12], pp. 46-51.
14. Nichiporuk, A.O. and Savenko O.S. (2013), "Models of the Life Cycle of Polymorphic Viruses", *Computer-Integrated Technologies: Education, Science, Production*, Lutsk, No. 11, pp. 64-71.
15. Derevyanko, O.S., Mezheritsky, S.G., Gavrilenko, S.Yu. and Klimenko A.M. (2009), System Programming. System service components, NTU "KhPI", Kharkov, 160 p.
16. Alfred, A., Rabbi, S. and Ullman D. (2001), *Compilers. Principles, technologies, tools*, Moscow-St. Petersburg-Kyiv, 768 p.
17. Gordeev, A.V. and Molchanov, A.Yu. (2002), System software, Peter, St. Petersburg., 734 p.

Надійшла (received) 23.02.2018
Прийнята до друку (accepted for publication) 25.04.2018

Система идентификации вредоносного программного обеспечения на основе контекстно-свободных грамматик

С. Ю. Гавриленко, В. В. Челак, В. А. Василев

Предмет статьи - исследование методов идентификации вредоносного программного обеспечения в компьютерных системах. **Цель** статьи - исследование существующих моделей выявления вирусов на базе формальных языков и грамматик и усовершенствование модели за счет использования LL(1) -грамматики. **Задача**: разработать математическую модель идентификации вредоносного программного обеспечения на основе контекстно-свободных грамматик; выбрать эффективный алгоритм ее работы, разработать программную модель и выполнить тестирование. Используемыми **методами** являются: аппарат формальных языков и грамматик, математические модели на основе детерминированных магазинных автоматов. Получены следующие **результаты**. Обоснован выбор типа грамматики и модели магазинного автомата. Разработано программное обеспечение, которое генерирует функции переходов магазинного автомата в соответствии с заданными правилами грамматики, анализирует входной файл на наличие заданных признаков, характерных для вредоносного программного обеспечения и моделирует работу детерминированного нисходящего магазинного автомата. По результатам работы магазинного автомата формируется вывод о возможности заражения компьютерной системы. **Выводы.** Научная новизна полученных результатов заключается в следующем: исследованы существующие модели антивирусных сканеров на базе формальных языков и грамматик; усовершенствована модель за счет использования LL(1) - грамматики, разработано программное обеспечение и выполнено тестирование. Проведенные экспериментальные исследования подтверждают возможность использования предложенного подхода, в качестве дополнительного средства для обнаружения вредоносного программного обеспечения.

Ключевые слова: компьютерные системы; антивирусное программное обеспечение; контекстно-свободные грамматики; LL(1) – грамматики; нисходящий магазинный автомат.

Malicious software identification system provision on the basis of context-free grammars

S. Gavrilenko, V. Chelak, V. Vassilev

The subject of the article is the study of methods for identifying malicious software in computer systems. **The goal** is to study existing models of virus detection on the basis of formal languages and grammars and to improve the model through the use of LL(1)-grammar. **Objective**: to develop a mathematical model for identifying malicious software based on context-free grammar; choose an effective algorithm for its job, develop a software model and perform testing. **The methods** used are: formal languages and grammars, mathematical models based on deterministic pushdown automatons. The following **results** have been obtained. The choice of grammar and the model of the pushdown automaton is substantiated. The software is developed, which generates the transfer functions of pushdown automaton in accordance with the given grammar rules, analyzes the input file for presence of the specified attributes, characteristic for malicious software, and simulates the work of the deterministic top-down pushdown automaton. Based on the result of the work of pushdown automaton, a conclusion is drawn about the possibility of the computer system being infected. **Conclusions.** Scientific novelty of the obtained results is as follows: the existing models of antivirus scanners on the basis of formal languages and grammars are investigated; the model was improved due to the use of LL(1)-grammar, the software was developed and the testing performed. The conducted experimental studies confirm the possibility of using the proposed approach as an additional means to detect the detection of malicious software.

Keywords: computer systems; antivirus software; context-free grammars; LL(1) – grammars; top-down pushdown automaton.