Улічев Олександр Сергійович, аспірант кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету, Кропивницький, Україна.
E-mail: askin79@gmail.com.
Orcid ID: 0000-0003-3736-9613.
Уличев Александр Сергеевич, аспирант кафедры кибербезопасности и программного обеспечения Центральноукраинского национального технического университета, Кропивницкий, Украина.
Ulichev Oleksandr, graduate student of Cybersecurity and Software Academic Department, Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine.

Мелешко Єлизавета Владиславівна, кандидат технічних наук, доцент, докторант кафедри кібербезпеки та програмного забезпечення Центральноукраїнського національного технічного університету, Кропивницький, Україна.
E-mail: elismeleshko@gmail.com.
Orcid ID: 0000-0001-8791-0063.
Мелешко Елизавета Владиславовна, кандидат технических наук, доцент, докторант кафедры кибербезопасности и программного обеспечения Центральноукраинского национального технического университета, Кропивницкий, Украина.
Meleshko Yelyzaveta, Candidate of Technical Sciences, Associate Professor, Doctoral Student of Cybersecurity and Software Academic Department, Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine.

DOI: 10.18372/2410-7840.22.14980
УДК: 004.491.4

ПРОГРАМНИЙ ПРОДУКТ ТИПУ SPYWARE ТА АНАЛІЗ ЙОГО СТІЙКОСТІ ДО ВИЯВЛЕННЯ ЗАСОБАМИ ЗАХИСТУ

Олександр Ковальов, Олександр Чобаль, Василь Різак

В період активного розвитку інформаційних технологій проблема збереження конфіденційності є надзвичайно актуальною. На сьогоднішній день існують декілька тисяч різновидів шкідливих програм, які працюють за різними алгоритмами. Однак їх всіх об'єднує факт того, що вони створюються спеціалізовано для несанкціонованого користувачем модифікування, знищення, блокування та копіювання інформації, порушуючи роботу комп'ютера та комп'ютерних мереж. Існує вид шкідливих програм, які здатні нанести значної шкоди конфіденційності інформації і при цьому вони залишаються не помітними навіть для спеціалізованих програм. Мова йде про програмних шпигунів (spyware). Для виявлення в системі програмного шпигуна слід використовувати спеціалізоване програмне забезпечення, яке спрямоване на виявлення саме цього виду загроз. Однак навіть вони не можуть гарантувати повної безпеки. У даній роботі описано основні типи програмних шпигунів та розроблено Spyware типу "системний монітор", завданням якого є збір користувацької інформації з можливістю подальшої її обробки та передачі. Ефективність роботи розробленої програми продемонстровано на основі зібраних даних та від'ємних результатів сканування системи спеціалізованими програмними засобами. Розглянуто особливості роботи програмних шпигунів та проведено аналіз їх поведінки, результати якого можуть бути використані при розробці імовірнісних методів пошуку програм досліджуваного типу.

Ключові слова: Spyware, malware, програмний шпигун, кейлогер, Windows.

ВСТУП

Важливою загрозою безпеці, яка сьогодні зачіпає багатьох користувачів Інтернету є шпигунське програмне забезпечення [7, 8]. Шпигунське ПЗ - це шкідливе програмне забезпечення, яке намагається непомітно відстежувати поведінку користувачів, записувати їх звички під час вебсерфінгу або красти їх конфіденційні дані, такі як логіни та паролі. Як правило, зібрана інформація відправляється назад розповсюджувачу шпигунських програм, де вона використовується для цільової реклами або в маркетингових дослідженнях. Це відрізняє їх від інших типів шкідливих програм, таких як віруси та хробаки, які зазвичай прагнуть поширитися на інші системи та завдати їм шкоди [6].

Оскільки проблема шпигунських програм загострилася, був введений ряд комерційних рішень, спрямованих на виявлення і видалення небажаних шпигунських програм. Ці інструменти схожі на антивірусні продукти в тому, що вони ідентифікують відомі екземпляри шпигунських програм, порівнюючи бінарне зображення невідомих зразків з базою даних відомих сигнатур [9].

Часто ці сигнатури генеруються вручну шляхом аналізу відомих зразків шпигунських програм (що є досить складним завданням, врахувати те що кожного дня доводиться аналізувати сотні нових випадків). На жаль, засоби виявлення шпигунських програм страждають від відомих недоліків детекторів які працюють на основі сигнатур, таких як постійна необхідність оновлення бази даних сигнатур і неможливість ідентифікувати раніше невідомі зразки. Зауважимо, що основним недоліком сигнатурних методів є те, що вони також часто не можуть впоратися із простими методами обфускації коду [10].

ПОСТАНОВКА ЗАДАЧІ

Оскільки методи виявлення на основі сигнатур мають суттєві недоліки, основним нашим завданням є розробити програмний засіб для виявлення в системі підозрілих програмних процесів, поведінка яких відповідає програмам типу Spyware. До функціоналу розроблюваної програми входять моніторинг наявних в системі програм та процесів, файлової системи та мережевої активності. Першим кроком у цьому напрямку є розробка та аналіз стійкості до виявлення засобами захисту програмного шпигуна з метою отримання даних стосовно поведінки даного типу програм і виявлення його можливих уразливостей.

Spyware. У загальному випадку шпигунське ПЗ відноситься до категорії шкідливих програм, які відстежують користувача без його згоди, як правило, в інтересах третьої сторони [1]. Існує декілька форм шпигунських програм, кожна з яких представляє унікальні загрози:

Adware - це рекламне програмне забезпечення, яке відображає спливаючу рекламу щоразу під час роботи програми. Часто програмне забезпечення доступне в Інтернеті безкоштовно, і реклама використовується в ньому для створення доходу для компанії.

Однак в той же час рекламне ПЗ може встановлювати на користувацькі комп'ютери компоненти, що відстежують особисту інформацію (включаючи вік, стать, місце розташування, переваги при покупках або звички до серфінгу) в маркетингових цілях.

Рекламні файли cookie - це частини програмного забезпечення, які веб-сайти зберігають на жорсткому диску під час відвідування сайтів. Деякі файли cookie існують тільки для того, щоб заощаджувати час, наприклад, в момент коли проставляється прапорець для веб-сайту щоб запам'ятати пароль на робочому комп'ютері.

Однак деякі сайти також зберігають рекламні файли cookie, які містять в собі особисту інформацію (наприклад, звички до серфінгу, імена користувачів, паролі, а також області інтересів) і діляться цією інформацією з іншими веб-сайтами. Такий обмін інформацією дозволяє маркетинговим фірмам створювати профіль користувача на основі користувацької інформації і продавати цей профіль іншим фірмам.

Троянські коні - це шкідливі програми, які встановлюються під виглядом бажаних програм. Троянські коні призначені для крадіжки, кодування або навіть знищення комп'ютерних даних. Деякі троянські коні, так звані Rats (Remote Administration Tools), надають зловмисникам необмежений доступ до комп'ютера щоразу, коли користувач знаходитесь в мережі. Зловмисник може виконувати такі дії, як передача файлів, додавання або видалення документів і програм, а також управління мишею і клавіатурою.

Системні монітори можуть фіксувати практично все що робить користувач під час роботи за комп'ютером – починаючи від натискання клавіш, електронної пошти та діалогу в чаті до того, які сайти він відвідує та які програми запускає.

Системні монітори зазвичай працюють у фоновому режимі так, що користувач навіть не здогадується що за ним спостерігають. Інформація, зібрана системним монітором, зберігається в системі в зашифрованому файлі журналу для

подальшого вилучення. Деякі програми також можуть надсилати файли журналу електронною поштою [2, 3].

ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

Розробка та основні можливості програмного шпигуна. В ході розробки програмного шпигуна типу "системний монітор" в ньому реалізовуються наступні можливості:

- зчитування інформації з клавіатури;
- ведення записів стосовно програм з якими працює користувач;
- зчитування зображення з екрану монітора;
- анонімна відправка зібраної інформації на електронну пошту.

Це далеко не весь список функцій, якими можуть володіти програмні шпигуни цього типу, однак навіть цих функцій достатньо щоб отримати доступ до конфіденційної інформації.

Програма створюється на мові C#, середою розробки виступатиме Microsoft Visual Studio 2019.

Реалізація технології Keylogger.

Keylogger представляє собою клас шкідливих програм, які збирають конфіденційні дані, записуючи будь-яку введену інформацію [4].

Для реалізації цієї технології ми використовуємо Windows гак (Windows Hook). Windows гак представляє собою ядро ключових реєстраторів. Гак - це точка в механізмі обробки системних повідомлень, у яку додаток може встановити процедуру для перехоплення трафіку повідомлень до того, як вони досягнуть цільової процедури вікна.

Функція може перехоплювати події до того, як вони досягнуть програми за допомогою цього механізму. Вона може впливати на події, змінювати або відкидати їх.

Гачки надають потужні можливості, а саме: обробляти або змінювати кожне повідомлення; записувати або відтворювати події клавіатури і миші; запобігати виклику іншого фільтра та багато інших можливостей [11].

У нашій реалізації це виглядає наступним чином: в момент, коли користувач натискає на певні клавіші клавіатури – ці клавіші водночас з'являються в потрібному користувачеві місці, але разом із тим заноситися до журналу подій програми. На стороні коду це виглядає наступним чином:

```
switch (e.KeyCode){
case Keys.A: // Задаємо клавішу на яку буде реагувати програма
logArea.Items.Add(time + " Натиснута клавіша «A»"); // інформація яка буде логуватись
break;
}.
```

Так як при натиснення кожної клавіші операція обробки клавіші тепер буде виконувати також і паралельне логування натиснутих клавіш, задля уникнення можливого зниження продуктивності системи ми відслідковуватимемо лише необхідні для нас клавіші.

Відомо що логіни та паролі для авторизації здатні містити в собі лише букву, цифри та спецсимволи. Для успішного перехоплення цих даних достатньо вести логування лише клавіш що відповідають буквам, цифрам та клавіші Shift.

Відслідковування активних вікон.

Для того, аби розуміти для якого саме вікна програми користувач наразі виконує ввід тексту з клавіатури ми додаємо можливість відслідковувати заголовки активних вікон. Метод, який реалізує даний функціонал має наступний вигляд:

```
const int numChars = 256;
StringBuilder Buffer = new StringBuilder(numChars);
IntPtr handle = GetForegroundWindow();
if (GetWindowText(handle, Buffer, numChars) > 0)
{
return Buffer.ToString();
}
return null;.
```

За його допомогою ми бачимо з якою саме програмою в даний момент працює користувач:

10:08:26 | Активне вікно «Notepad++»
10:08:28 Натиснута клавіша «T»
10:08:29 Натиснута клавіша «E»
10:08:29 Натиснута клавіша «S»
10:08:31 Натиснута клавіша «T»
10:08:36 | Активне вікно «Google Chrome»
10:08:41 Натиснута клавіша «P»
10:08:42 Натиснута клавіша «A».

Зчитування інформації з екрану монітора.

При отриманні віддаленого доступу до комп'ютера користувача зловмисник має можливість переглядати всі файли в системи. Однак інколи достатньо зробити знімок екрану, аби отримати доступ до необхідної інформації.

Для отримання зображення з екрану монітора ми використовуємо наступний метод:

```
int sWidth = Screen.PrimaryScreen.Bounds.Width;
int sHeight = Screen.PrimaryScreen.Bounds.Height;
Graphics GraphicsFirst;
Bitmap BitmapFirst = new Bitmap(sWidth, sHeight);
GraphicsFirst = Graphics.FromImage(BitmapFirst);
GraphicsFirst.CopyFromScreen(Point.Empty,
Point.Empty, Screen.PrimaryScreen.Bounds.Size);
GraphicsFirst.Dispose();
BitmapFirst.Dispose(),.
```

Завдяки даному методу ми можемо робити знімок екрану монітора та зберігати їх у вказаній директорії. Однак при кожному взятті зображення перш ніж зберегти його, фото буде записане в оперативну пам'ять системи.

В результаті цього, за декілька годин роботи, комп'ютер почне виконувати задачі значно повільніше, і навіть не добре підкований у роботі системи користувач помітить, що якийсь із процесів використовує занадто великий об'єм пам'яті. Для того аби цього уникнути використовуємо функцію Dispose() після кожного збереження фотографії. При цьому фото буде автоматично видалятися із оперативної пам'яті системи.

Функцію знімку екрану можна встановити або на таймер, для того щоб зчитування зображення відбувалось періодично, або налаштувати на певну подію. Оскільки ми вже маємо можливість отримувати інформацію стосовно активного на даний момент вікна з яким працює користувач, ми також маємо змогу реагувати на окремі з них. Так, скажімо, коли користувач відкриє сайт свого банку і починає вводити свій пін-код через натискання графічних клавіш в інтерфейсі сайту – ми здатні робити знімок екрану в момент його вводу. Оскільки графічні клавіші прагнуть обійти введення PIN-кодів через клавіатуру, за допомогою такого зчитування інформації ми можемо обійтись і без клавіш.

Передача зібраної інформації.

Передавати дані можна по різному. В поточній реалізації зібрана інформація надсилається у вигляді електронного листа на вказаний поштовий адрес. Перш ніж надсилати зібрану інформацію ми проводимо їх архівацію аби процес відправки займав менше часу та був менш помітним.

В .Net для відправки електронної пошти виступає простір імен System.Net.Mail. За допомогою таких класів як SmtpClient та MailMessage ми маємо змогу надсилати лист через SMTP протокол на вказаний електронний адрес [12]. У поточній реалізації лист зі "звітом" надсилається циклічно наприкінці кожного робочого дня.

Однак даний алгоритм може бути змінено в залежності від завдань які перед ним ставляться. Наприклад, якщо зловмисник бажає отримати саме паролі що стосуються окремо взятих ресурсів, відправка листа відбудеться одразу після отримання цих даних, а шпигун, ймовірно, буде автоматично самовидалений.

Результати ідентифікації розробленого Spyware та особливості його роботи.

Перед тим як оцінити стійкість до виявлення професійним ПЗ, розроблений програмний шпигун було піддано аналізу за допомогою професійних антивірусних програм [16]. Як і очікувалось, ряд антивірусних програм не розцінюють розроблену програму як загрозу [17]. Для оцінки стійкість до виявлення більш спеціалізованими програмними засобами ми використали наступні рішення: Захисник Windows, IObit Malware Fight 7.7 Pro, Malwarebytes Premium 4.1.0.

Захисник Windows був розроблений навколо ядра більш старого продукту під назвою Giant AntiSpyware, який спочатку вироблявся компанією Giant company Software. Основним завданням Захисника Windows є перевірка наявність шпигунських, рекламних та інших шкідливих програм в системах під управління ОС Windows [5, 12]. IObit Malware Fight являє собою потужний програмний засіб який захищає систему від різного роду загроз, до числа яких входить також і шпи-

гунське програмне забезпечення, трояни, рекламне програмне забезпечення тощо.

Таблиця 1

Результати перевірки програми антивірусними засобами.

Назва антивірусного програмного забезпечення	Результат перевірки
BitDefender	Не виявлено
TrendMicro	Не виявлено
Kaspersky	Не виявлено
Webroot	Не виявлено
Avast	Не виявлено
Ad-Aware	Не виявлено
Panda	Не виявлено
DrWeb	Не виявлено
ESET-NOD32	Не виявлено
McAfee	Не виявлено

А з підтримкою Bitdefender, що містить в собі більш ніж 200-мільйонну базу даних захисту від шкідливих програм, дане ПЗ підтримує блокування будь-яких загроз [14, 13].

Malwarebytes представляє собою антивірусну програму яка зосереджена на пошуку та знешкодженні шкідливих програм у системі.

До основних функціональних складових програми відносяться антивірус, антируткіт, функція блокування шкідливих веб-сайтів та антишпигунський модуль [15].

Як показують результати перевірки – розроблений продукт є достатньо стійким до виявлення як засобами антивірусного захисту так і професійним засобом пошуку програм цього типу.

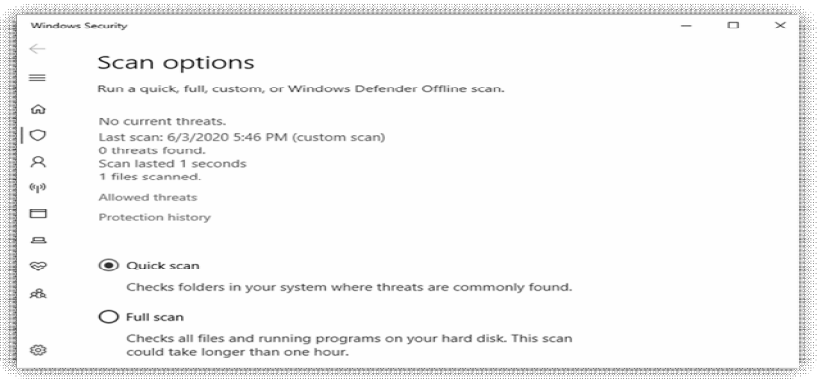

Рис. 1 Результат сканування розробленого Spyware програмою Windows Defender

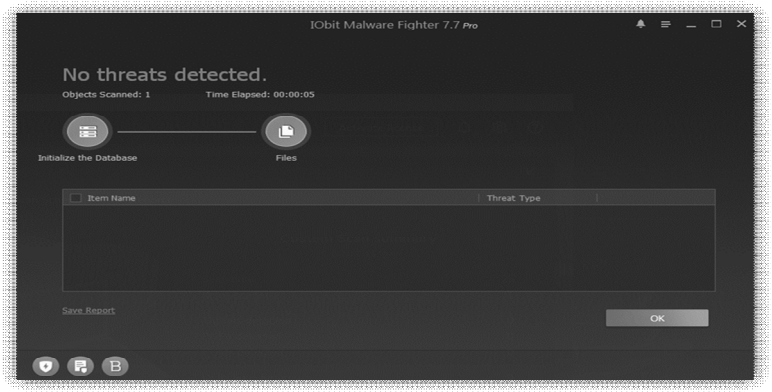

Рис. 2 Результат сканування розробленого Spyware програмою IObit Malware Fighter 7.7 Pro.

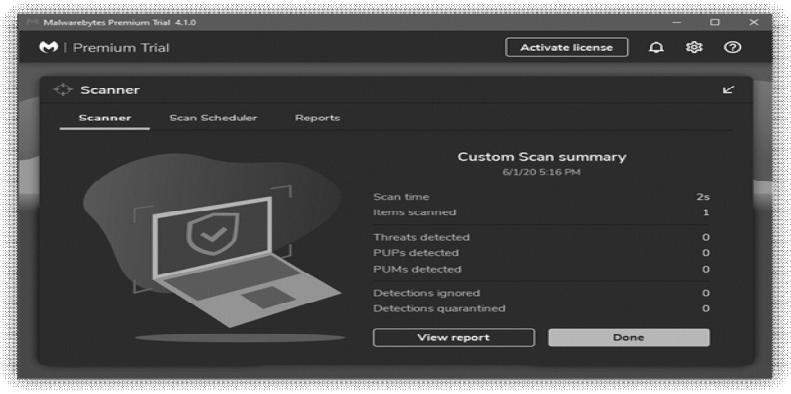

Рис. 3 Результат сканування розробленого Spyware програмою Malwarebytes Premium 4.1.0.

Аналіз можливих вразливостей.

Не дивлячись на те, що розроблений програмний шпигун показав себе стійким до виявлення, нам вдалось виявити також і слабкі його сторони які, ймовірно, можуть бути присутні і в інших програмах цього типу.

По-перше, під час роботи з функцією кейлогера програма "позичає" певний відсоток ресурсів центрального процесора. При повсякденному використанні комп'ютера даний процес є не помітним. Однак якщо кількість фактичних натисків клавіш буде занадто великою на одиницю часу – програмному шпигуну доведеться запозичити більше ресурсів від центрального процесора, аби мати змогу обробити кожну натиснуту клавішу. В результаті цього програма може значно виділятись на фоні інших робочих процесів.

По-друге, вдалось встановити, що при активній роботі програми із зображеннями або великою кількістю зібраної інформації - розмір споживаної оперативної пам'ять, яка виділяється під роботу програми, може різко змінюватись. При веденні моніторингу показника затрат на споживання оперативної пам'яті для наявних в системі процесів - виявити серед активних процесів spyware цілком можливо.

По-третє, при зборі інформації, для того аби залишатись непомітною, програма змушена зберігати зібрані дані в системі. Скануючи файлову систему на наявність нових файлів або файлів які постійно змінюються, цілком ймовірно відстежити файл, який використовується програмним шпигуном в якості сховища даних.

ВИСНОВКИ

У даній роботі розроблено програмний шпигун виду "системний монітор" на мові C# в середовищі програмування Visual Studio 2019. Розглянуто проблематику даного типу шкідливих програм. Представлено частини коду розробленого додатку, що реалізує досліджувані функції програм типу Spyware, а саме: технологію Keylogger; функцію відслідковування активних вікон; зчитування інформації з екрану монітора. На основі проведеного поведінкового аналізу шпигунського програмного забезпечення показано, що основними факторами, які дозволяють ідентифікувати його присутність у системі є динамічна зміна в показниках споживання ресурсів процесора, динаміка у роботі програм з оперативною пам'яттю а також можливість створення тимчасові лог файли перед подальшою їх передачею. Виявлені слабкі місця програмних шпигунів типу "системний монітор" можуть бути використані при розробці спеціалізованого програмного забезпечення, що використовує технології імовірнісного аналізу.

ЛІТЕРАТУРА

[1] J. Yan, Y. Qi and Q. Rao,"Detecting malware with an ensemble method based on deep neural network", *Secur. Commun. Netw.*, vol. 2018(1), 2018. – pp. 1-16.

[2] P. Wang and Y.-S. Wang, "Malware behavioural detection and vaccine development by using a support vector model classifier*", J. Comput. Syst. Sci.*, vol. 81(6), 2015. – pp. 62-67.

[3] R. Islam, R. Tian, L. M. Batten and S. Versteeg, "Classification of malware based on integrated static and dynamic features", J. *Netw. Comput. Appl.*, vol. 36(2) 2013. – pp. 1012-1026.

[4] Ladakis E., Koromilas L., Vasiliadis G., Polychronakis M., Ioannidis S. "You Can Type, but You Can't Hide: A Stealthy GPU-based Keylogger." *In Proceedings of the 6th European Workshop on System Security*. EuroSec, Prague, Czech Republic, April 2013. – 6 p.

[5] Hassell J., Campbell T.: "*Windows Vista: Beyond the Manual*"; New York: Apress , 2007. – 477 p.

[6] Steven D. Gribble Alexander Moshchuk, Tanya Bragin and Henry M. Levy. A CrawlerBased Study of Spyware on the Web. *In Proceedings of the Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2006. – pp. 1-17.

[7] *Combating Spyware*: H.R. 29, the SPY Act : Hearing Before the Committee on Energy and Commerce, House of Representatives, One Hundred Ninth Congress, First Session, January 26, 2005. – 71 p.

[8] Thompson, R. Why Spyware Poses Multiple Threats to Security. *Communications of the ACM* 48, 8 (2005), 2005. - pp. 6-8.

[9] Saroiu, S., Gribble, S., Levy, H. Measurement and Analysis of Spyware in a University Environment. *In Usenix NSDI* (2004), 2004. – 17 p.

[10] Christodorescu, M., Jha, S. Testing Malware Detectors. *In ACM International Symposium on Software Testing and Analysis (ISSTA)*, 2004. – pp. 1-11.

[11] Muhammad Aslam, Rana Naveed Idrees, Mirza

Muzammil Baig, and Muhammad Asif Arshad, "Anti-Hook Shield against the Software Key Loggers", *National Conference on Emerging Technologies*, 2004. – pp. 189-191.

[12] *The home for Microsoft documentation and learning for developers and technology professionals* [Електронний ресурс] – Режим доступу до ресурсу: https://docs.microsoft.com/en-us/.

[13] *Bitdefender* [Електронний ресурс] – Режим доступу до ресурсу: https://bitdefender.ua/.

[14] *IObit Malware Fighter* [Електронний ресурс] – Режим доступу до ресурсу: https://iobit.com /en/malware-fighter.php.

[15] *Spyware* [Електронний ресурс] – Режим доступу до ресурсу: https://malwarebytes.com/spyware.

[16] *Best Windows 10 antivirus of 2020* [Електронний ресурс] – Режим доступу до ресурсу: https:// techradar.com/best/best-windows-10-antivi rus.

[17] *Virtuozal* [Електронний ресурс] – Режим доступу до ресурсу: https://virustotal.com/gui /file/096f87ae423557d8d2b4a19437058f104fa7d ab58ec29fd85eac3a9e5aa10c1a/detection.

ПРОГРАММНЫЙ ПРОДУКТ ТИПА SPYWARE И АНАЛИЗ ЕГО СТОЙКОСТИ К ОБНАРУЖЕНИЮ СРЕДСТВАМИ ЗАЩИТЫ

В период активного развития информационных технологий проблема конфиденциальности является чрезвычайно актуальной. На сегодняшний день существуют несколько тысяч разновидностей вредоносных программ, которые работают по разным алгоритмам. Однако их всех объединяет факт того, что они создаются специализированно для несанкционированной пользователем модификации, уничтожения, блокирования и копирования информации, нарушая работу компьютера и компьютерных сетей. Существует вид вредоносных программ, которые способны нанести значительный ущерб конфиденциальности информации и при этом они остаются не заметными даже для специализированных программ. Речь идет о программных шпионах (spyware). Для выявления в системе программного шпиона следует использовать специализированное программное обеспечение, которое направлено на выявление именно этого вида угроз. Однако даже они не могут гарантировать полной безопасности. В данной работе описаны основные типы программных шпионов и разработан Spyware типа "системный монитор", задачей которого является сбор пользовательской информации с возможностью дальнейшей ее обработки и передачи. Эффективность работы разработанной программы продемонстрирована на основе собранных данных и отрицательных результатов сканирования системы специализированными программными средствами. Рассмотрены особенности работы программных шпионов и проведен анализ их поведения, результаты которого могут быть использованы при разработке вероятностных методов поиска программ исследуемого типа.

Ключевые слова: Spyware, malware, программный шпион, кейлоггер, Windows.

SPYWARE-TYPE SOFTWARE PRODUCT AND ANALYSIS OF ITS RESISTANCE TO DETECTION BY SECURITY TOOLS

In the period of active development of information technologies, the problem of confidentiality is extremely urgent. Today, there are several thousand varieties of malware that work according to different algorithms. However, they are all united by the fact that they are created specifically for unauthorized modification, destruction, blocking and copying of information by the user, disrupting the operation of the computer and computer networks. Spyware is software that collects and transmits information about a user without their consent. This information may include his personal data, the configuration of his computer and operating system, and Internet statistics. The basic set of functions of the spyware can include functions for reading information from the user's keyboard, taking screenshots of the monitor screen, logging sites visited by the user, unsanctioned analysis of the security system status, and much more. There is a type of malware that can cause significant damage to the user's privacy and at the same time they will be unnoticed even for specialized programs. We are talking about spyware. To detect spyware in the system, user should use specialized software that is aimed at identifying this type of threat. However, even they cannot guarantee complete security. This article describes the main types of spyware and has developed a "system monitor" Spyware, the task of which is to collect information about users with the possibility of further processing and transmission. The efficiency of the developed program is demonstrated on the basis of the collected data and negative results ofscanning the system by specialized software. The features of the work of software spies are examined and an analysis of their behavior is carried out, which results can be used at development of probabilistic methods for finding programs of the type under investigation.

Keywords: Spyware, malware, software spy, keylogger, Windows.

Ковальов Олександр Олександрович, аспірант кафедри твердотільної електроніки та інформаційної безпеки фізичного факультету УжНУ.
E-mail: alexandr.kovalev@uzhnu.edu.ua.
Orcid ID:0000-0003-0630-9132.

Ковалёв Александр Александрович, аспирант кафедры твердотельной электроники и информационной безопасности физического факультета УжНУ.

Kovalev Alexander, PhD student, Department of Solid State Electronics and Information Security of the Physics Faculty, UzhNU.

Чобаль Олександр Ілліч, кандидат фізико-математичних наук, доцент кафедри твердотільної електроніки та інформаційної безпеки фізичного факультету УжНУ.
E-mail: oleksandr.chobal@uzhnu.edu.ua.
Orcid ID:0000-0002-8042-8052.
Чобаль Александр Ильич, кандидат физико-математических наук, доцент кафедры твердотельной электроники и информационной безопасности физического факультета УжНУ.
Chobal Oleksandr, Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Solid State Electronics and Information Security of the Physical Faculty, UzhNU.

Різак Василь Михайлович, доктор фізико-математичних наук, професор, завідувач кафедри твердотільної електроніки та інформаційної безпеки фізичного факультету УжНУ.
E-mail:vrizak@uzhnu.edu.ua.
Orcid ID: 0000-0002-9177-0662.
Ризак Василий Михайлович, доктор физико-математических наук, профессор, заведующий кафедрой твердотельной электроники и информационной безопасности физического факультета УжНУ.
Rizak Vasyl, Doctor of Physical and Mathematical Sciences, Professor, Head of the Department of Solid State Electronics and Information Security of the Physical Faculty, UzhNU.

DOI: 10.18372/2410-7840.22.14981
УДК 621.391:519.2

ШВИДКА РОЗРІЗНЮВАЛЬНА АТАКА НА ШИФРОСИСТЕМУ NTRUCipher+

Александра Матійко, Антон Олексійчук

Шифросистему NTRUCipher запропоновано в 2017 р. як симетричну версію алгоритму шифрування NTRUEncrypt, який є на сьогодні одним з найшвидших постквантових криптографічних алгоритмів, що базуються на решітках у евклідовому просторі. Метою побудови NTRUCipher є створення симетричної шифросистеми для практичних застосувань, стійкість якої, аналогічно асиметричним, базується на складності розв'язанні лише однієї обчислювальної задачі. Проведені попередні дослідження зазначеної шифросистеми, проте за їх рамками залишається питання про стійкість NTRUCipher відносно розрізнювальних атак, спрямованих на побудову статистичних критеріїв для розрізнення послідовностей шифрованих повідомлень шифросистеми та суто випадкових послідовностей. У даній статті показано, що шифросистема NTRUCipher та навіть її природне удосконалення – NTRUCipher+, запропоноване по аналогії з відомою обґрунтовано стійкою версією асиметричної криптосистеми NTRU, є вразливими відносно розрізнювальних атак. Запропоновано швидку розрізнювальну атаку на шифросистему NTRUCipher+ та (для окремого випадку) ще більш швидку модифікацію цієї атаки. Отримано аналітичні оцінки трудомісткості обох атак, з яких випливає, що вони мають поліноміальну часову складність та можуть бути реалізовані в режимі реального часу (для стандартного набору параметрів шифросистеми). Отримані результати свідчать про те, що для побудови симетричних NRTU-подібних криптосистем слід використовувати інші загальні конструкції.

Ключові слова: *постквантова криптографія, криптосистеми на решітках, розрізнювальна атака, дискретне перетворення Фур'є, NTRUEncrypt, NTRUCipher.*

ВСТУП

На сьогодні асиметричні шифросистеми, побудовані за схемою алгоритму шифрування NTRUEncrypt [8], відносяться до найшвидших постквантових криптосистем і протоколів. Вони будуються на основі арифметичних операцій (додавання, множення та обернення) у кільцях зрізаних поліномів та за умови належного вибору параметрів забезпечують потрібну стійкість відносно відомих атак поряд із високою швидкістю шифрування та прийнятними для багатьох застосувань довжинами відкритих ключів і шифрованих текстів.

До NTRU-подібних (або близьких до них, типу LWE) криптосистем відноситься майже третина усіх постквантових криптографічних алгоритмів, представлених на конкурс NIST зі стандартизації постквантових криптопримітивів (див. роботи [3, 5] та наведені у них посилання).

В [11] запропоновано симетричну версію алгоритму NTRUEncrypt – шифросистему NTRUCipher, перші дослідження якої проведено в [11, 2]. Зауважимо, що за рамками цих досліджень