УДК 004.380

Фенко О.Г., Лисенко Д.І.

НАПРЯМКИ АУДІТУ ТА ЗАХИСТУ ІНФОРМАЦІЙНИХ СИСТЕМ

Полтавський національний технічний університет імені Юрія Кондратюка,
Полтава, пр-т. Першотравневий 24, 36011

Fenko O.G., Lysenko D.I.

DIRECTIONS OF AUDIT AND DEFENCE OF THE INFORMSYSTEMS

Poltava National Technical University named after Yuri Kondratyuk,
Poltava, Ave. Pershotravnevuy 24, 36011

Анотація. Розглядаються актуальні питання створення систем захисту інформації в умовах повної відкритості сучасних інформаційних технологій. Автори намагаються освітити низку запитань, пов'язаних із забезпеченням безпеки інформаційних технологій, а також прагнуть сформувати цілісне уявлення про шляхи створення систем захисту інформації.

Ключові слова: сучасні технології шкідливого програмного забезпечення (ПЗ), кіберзлочинність

Abstract. The pressing questions of creation of the systems of priv are examined in the conditions of complete openness of modern information technologies. An authors tries to light up the row of the questions related to providing of safety of information technologies, and also aims to form an integral idea about the ways of creation of the systems of priv.

Key words: modern harmful technologies software, cybercrime

Вступ.

Сьогодні Україна зіштовхнулася не тільки з проблемою інформаційної війни, а з цифровою (кібервойною). У цих визначеннях є загальне призначення, але різні рішення задач - це показали трагічні події останніх років. Глобальне проникнення інформаційних технологій у наше життя, поступовий перехід до

електронних способів ведення бізнесу ставлять перед учасниками ринку задачі по забезпеченню інформаційної безпеки. Загальна інформатизація супроводжується зростанням числа комп'ютерних злочинів і, як наслідок, матеріальних втрат. Тому інформаційна безпека стала обов’язковою умовою.

Огляд літератури.

На перший план висувається безпека технологій створення програмного забезпечення комп'ютерних систем [5,6]. Робота [1] охоплює питання безпечного кодування в широкому спектрі програмного забезпечення, що забезпечує його стійкість до експлойтів на етапі експлуатації, однак не може протидіяти шкідливому програмному забезпеченню (malware), що використовує руткіт-технології. У роботі [2] приведений опис більшості відомих насьогодні руткіт-технологій, що використовує шкідливе програмне забезпечення. У роботі [3] розглянуті загальні принципи побудови двомодульних обчислювальних процедур, методології самотестування й оцінки стійкості програм. Також у цій роботі зібрана значна історія з питань розробки таких програм, що можуть самостійно перевірятися (self-testing) і самостійно відновлюватися (self-healing).

Основний текст.

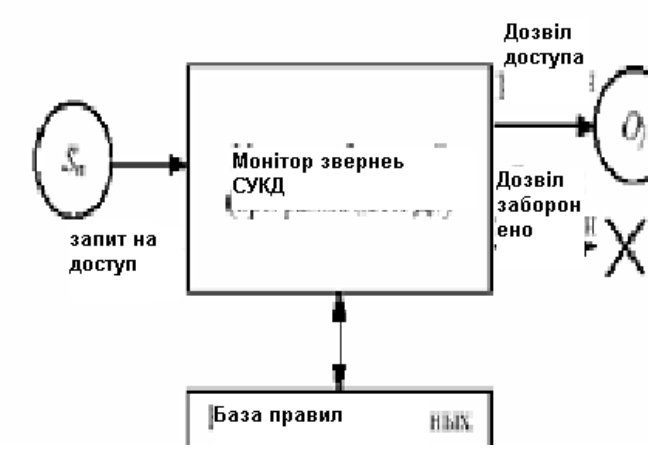
Дослідники звичайно виділяють три основних види погроз безпеки - це погрози розкриття, цілісності і відмовлення в обслуговуванні (DoS). Погроза розкриття полягає тім, що інформація стає відомої тому, кому не варто було б її знати. Іноді замість слова "розкриття" використовуються терміни "крадіжка" або "витік". Погрозу цілісності містить у собі будь-яка навмисна зміна даних, що зберігаються в обчислювальній системі або, що передаються з однієї системи в іншу. Звичайно вважається, що погрозі розкриття піддаються в більшому ступені державні структури, а погрозі цілісності - ділові або комерційні. Зазначимо, що за роки незалежності в Україні лише закладено основи для формування системи забезпечення інформаційної безпеки. В електронному просторі передачі інформації використовуються наступні прийоми досягнення терористичних цілей:

1. Нанесення збитку окремим фізичним елементам простору, руйнування мереж електроживлення, наведення перешкод, використання спеціальних програм, що стимулюють руйнування апаратних засобів, біологічні і хімічні засоби руйнування елементної бази і т.д.

2. Крадіжка або знищення інформаційного, програмного і технічного ресурсів простору, що мають суспільну значимість, шляхом подолання систем захисту, упровадження вірусів, програмних закладок і т.п.

Модель Take-Grant, що реалізує дискреційну політику розмежування прав (безпеки). Під політикою безпеки розуміється сукупність норм і правил, що регламентують процес обробки інформації, виконання яких забезпечує захист від визначеної безлічі погроз і складає необхідну умову безпеки системи (мал. 1). У випадку використання дискреційної політики безпеки виникає необхідність визначення правил поширення прав доступу й аналізу їхнього впливу на безпеку ІС. У моделі Take-Grant як основні елементи використовуються граф доступів і правила їх перетворень. Формальний опис моделі Take-Grant виглядає так:

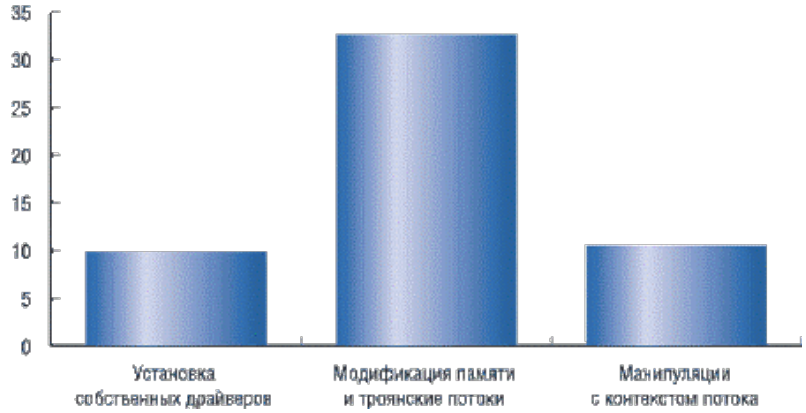
1. Безліч об'єктів – ПРО, де oj ∈ ПРО, ПРО ={про1, про2,…,oj}, j ∈ N;
2. Безліч суб'єктів – S, де sn ? S, S ={s1, s2,...,sn}, n ? N;
3. Безліч активних суб'єктів – S ?O.



Мал. 1 Монітор реалізації системи безпеки

Термін руткит (від англ. root kit – «набір для одержання прав адміністратора») є не що інше, як програма або набір програм для схованого

узяття під контроль зламаної системи. У контексті приховання вірусного коду в системі Windows під rootkit прийнято мати на увазі такий код, що, будучи впровадженим у систему, здатний перехоплювати системні функції (Windows API) (мал.2). Перезаписані інструкції API-функції виконуються у коді руткіту для забезпечення непорушності виконання функції. Таке перехоплення здійснюється цілеспрямовано з урахуванням версії ОС та її встановлених оновлень. Протидію цьому методу перехоплення функцій API організовано шляхом обчислення контрольної суми (CRC) бібліотеки API в пам'яті та порівняння її з відомою контрольною сумою «чистої» бібліотеки. Відмінність указаних контрольних сум дає змогу стверджувати про порушення оригінального коду бібліотеки. Код «чистої» бібліотеки API-функцій береться з власноруч створеного процесу, який уникнув атаки руткіту. Такий процес створюється без використання викликів функцій Win API шляхом використання NativeAPI викликів. Неважко здогадатися, що таке перехоплення і модифікація API- функцій дозволяють руткіту лехко і просто замаскувати свою присутність у зламаній системі.



Мал. 2 Монітор реалізації руткіт-технології

User-mode-категорія руткітів заснована на перехопленні функцій бібліотек користувацького режиму, kernel-mode – на установці в систему. Розрізняють наступні основні методи приховання:

1)Модифікація таблиці імпорту. Мабуть, саме ця методика приховання претендує на звання класичної. Технологія такого маскування полягає в

наступному: rootkit знаходить у пам'яті таблицю імпорту програми, що виконується, і коректує адреси цікавлячих його функцій на адреси своїх перехоплювачів. У момент виклику API-функції програма зчитує її адресу з таблиці імпорту і передає по цій адресі керування. Пошук таблиці імпорту в пам'яті нескладний, оскільки для цього уже відомі спеціалізовані API-функції, що дозволяють працювати з образом програми в пам'яті.

2)Модифікація машинного коду прикладної програми. Як випливає з назви, суть методу полягає в модифікації машинного коду, що відповідає в прикладній програмі за виклик тієї або іншої API-функції. Реалізація методики досить складна, обумовлено це багатою розмаїтістю мов програмування і версій компіляторів, до того ж і сама реалізація викликів API-функцій може бути різна.

3)Модифікація програмного коду API-функції. Методика полягає в тім, що руткіт повинен знайти в пам'яті машинний код цікавлячих його API-функцій і модифікувати його. При цьому втручання в машинний код функцій, що перехоплюються, мінімально. На початку функції звичайно розміщають 2-3 машинні команди, що передають керування основній функції-перехоплювачеві. Основною умовою такої методики є збереження вихідного машинного коду для кожної модифікованої їм функції.

4)Перехоплення функцій LoadLibrary і GetProcAddress. Виконується шляхом модифікації таблиці імпорту: якщо перехопити функцію GetProcAddress, то при запиті адреси можна видавати програмі не реальні адреси цікавлячих її функцій, а адреси своїх перехоплювачів. При виклику GetProcAddress вона одержує адресу і виконує виклик функції.

5)Перехоплення функцій у режимі ядра (kernel mode). Взаємодія з ядром здійснюється через ntdll.dll, більшість функцій якої є посередниками при звертанні до ядра через переривання INT 2Eh. Кінцеве звертання до функцій ядра засновано на структурі ServiceDescrip-torTable (або скорочено SDT), розташованої в ntoskrnl.exe. SDT – таблиця, що містить адреси крапок входу сервісів ядра NT. Спрощено можна сказати, що для перехоплення функцій

необхідно написати драйвер, що зробить модифікацію таблиці SDT. Перед модифікацією драйверові необхідно зберегти адреси функцій, що перехоплюються, і записати в таблицю SDT адреси своїх оброблювачів.

Заключення та висновки.

Відзначаючи високий рівень активності і зацікавленості міжнародного співтовариства у стратегічному вирішенні проблем розвитку інформаційного простору; розглянувши визначення інформаційної безпеки, яке є комплексним і досвід провідних країн у цій сфері, який може стати прикладом для України у формуванні її власної стратегії в інформаційній сфері, можна зробити відповідні висновки. Створення реального міжнародного консенсусу з цього питання між державами - партнерами є об'єктивною необхідністю, оскільки зростає кіберзлочинність як на національному, так і на міжнародному рівні. Україна вже сьогодні відчуває вплив кіберзлочинності, і об'єктивно зацікавлена у формуванні відповідної політики і побудові власної системи кібербезпеки, в першу чергу шляхом створення Стратегії, оскільки з керівних документів державної політики України, що визначають діяльність органів влади в інформаційній сфері доктринально визначені питання безпеки; закони, які пов'язані з новітніми інформаційними технологіями.

Література:

1. Ховард М., Лебланк Д. Защищенный код. /Пер.с англ. – М.: Издательско торговый дом «Русская Редакция», 2004. − 704 с.

2. Зайцев О.В. Rootkits, Spyware/Adware, Keyloggers&Backdoors: обнаружение и защита. – СПб.: БХВ-Петербург, 2006. – 304 с.

3. Казарин О. В. Теория и практика защиты программ. – М.: МГУЛ, 2004. − 450 с.

4. Зілінський Ю.В., Бельська В.Ю., Юдіна А.Л. Захист і оптимізація програмного забезпечення шляхом прямих викликів сервісів ядра операційних систем Windows NT // Вісник Кременчуцького державного політехнічного

університету імені Михайла Остроградського. – Кременчук: КДПУ, 2009. – Вип. 5/2009 (58), ч. 1. – С. 49–53.

5. Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу: НД ТЗІ 1.1–002–99. – Київ: ДСТСЗІ СБ України, 1999. – 16 с.

6. Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу: НД ТЗІ 2.2–004–99. – Київ: ДСТСЗІ СБ України, 1999. – 55 с.

Стаття відправлена: 26.11.2015р.