УДК 304.491

СУЧАСНІ ВІРУСНІ ЗАГРОЗИ ДЛЯ КОМП'ЮТЕРНИХ СИСТЕМ СІМЕЙСТВА ОПЕРАЦІЙНИХ СИСТЕМ *WINDOWS*

І. І. Пархоменко, канд. техн. наук, доц., ***Б. О. Молодан***

Національний авіаційний університет

parkh08@gmail.com, molodan.bohdan@gmail.com

Розглянуто найбільш технологічні та небзпечні вірусні загрози за 2010 рік, їх принципи роботи, внутрішню архітектуру, вразливості які використовуються «злошкодними» програмами та методи обходу захисту антивірусного програмного забезпечення.

Ключові слова: інформаційна безпека, комп'ютерні віруси, вразливості 0-day, bot-net, worm-malware.

Considered the most technologically and dangerous virus threats in 2010 year, their principles of work, internal architecture, vulnerability used malware and methods for circumventing anti-virus protection software.

Keywords: information security, computer virus, vulnerability 0-day, bot-net, worm-malware.

Постановка проблеми

На сьогодні гостро стоїть питання захищеності комп'ютерних систем від ураження комп'ютерними вірусами, «злошкодним» кодом, адже їх кількість кожен рік зростає в геометричній прогресії. Проте сучасні шкідливі програми усе рідше та рідше містять оригінальні ідеї та технологічні знахідки. 2010 рік був значущим з погляду питання боротьби з новітніми загрозами зі сторони комп'ютерних вірусів. Також треба зосередити увагу на тому, що хакери відходять від стандартних схем заробітку на комп'ютерних вірусах. Сьогодні все актуальнішою стає модель заробітку не шляхом крадіжки номерів банківських карточок або іншої приватної інформації, а створення мережі заражених комп'ютерів (*bot-net*). У подальшому *bot-net* може бути використаний для проведення масових *dDos* атак. Це становить велику небезпеку для інформаційної безпеки в цілому у світі, оскільки кількість атакуючих може складатися з декількох тисяч заражених комп'ютерів, а знищити або знайти центр керування цієї *bot-net* мережі майже неможливо. У 2010 році створено найтехнологічніший комп'ютерний вірус *Stuxnet*, який своєю деструктивною діяльністю зупинив роботу АЕС у місті Бушер (Іран). Перша реалізація вірусу, який був спрямований на ушкодження обладнання промислового класу.

Аналіз досліджень і публікацій

Питанням проблем антивірусної безпеки за 2010 рік було приділено увагу в звітах провідних розробників антивірусного програмного забезпечення, а саме: *ESET*, *Symantec*, *Kaspersky Lab*, і навіть у звітах *Microsoft*. При розгляді звітів з антивірусної безпеки кожен з авторів розглядав проблему лише на прикладі одного окремо взятого вірусу. Так, у звіті компанії *ESET* увага приділялася проблемі шкідливого коду *Stuxnet* [1]. У загально доступному звіті не освітлено проблему в цілому, лише назва вірусу за класифікацією вендора, та короткий опис можливих наслідків деструктивної діяльності шкідливого коду.

Мета статті — розглянути та проаналізувати сучасні комп'ютерні загрози з боку «злошкодного» коду за 2010 рік, їх принципи роботи, внутрішню архітектуру, методи обходу захисту антивірусного забезпечення, вразливості ОС *Windows*, які можуть бути використані шкідливими програмами.

1) ***Stuxnet*** — отримав назву найтехнологічнішої шкідливої програми за весь час існування комп'ютерних вірусів. Річ у тому, що *Stuxnet* використовував цілий набір цікавих технологічних знахідок, сконцентрованих в одній шкідливій програмі. Компоненти *Stuxnet* (у тому числі і драйвери) мали легальні цифрові підписи. Це давало змогу безперешкодно обійти ряд *HIPS*-систем, а в деяких випадках уникнути виявлення антивірусним монітором або сканером. Сертифікати, які були використані, належали таким компаніям, як *Realtek* і *JMicron*. А найголовніше вони надавали можливість підпису модулів без відправлення їх на верифікацію в *Microsoft* та безперешкодно підписати драйвери, що використовують руткіт-технології.

Звичайно розробники *Stuxnet* не самотні у своєму бажанні використовувати легальні цифрові сертифікати для підпису шкідливих компонентів. Але найчастіше ми маємо справу з сертифікатами якої-небудь маловідомої китайської компанії, яка спеціально була створена для отримання легального сертифіката та подальшого його перепродажу, але сертифікати належали відомим компаніям, виробникам комп'ютерних комплектуючих. Використання одразу шести

вразливостей в одній шкідливій програмі, п'ять з яких є 0-day уразливостями, безпрецедентний випадок за вартістю та кількістю вразливостей. Цікавим представником цього набору є вразливість *MS*10-046, яка дозволяла безперешкодно поширюватися через зовнішні носії, експлуатуючи вразливість в обробці *LNK*/*PIF*-файлів (автоматичний запуск на виконання шкідливої програми з носія без використання *Autorun.inf*, використовуючи лише ярлики). Вектор розповсюдження з використанням цієї уразливості був основним, але варто було «червяку» проникнути в локальний сегмент внутрішньої мережі, як він починав пошук жертв з використанням додаткових векторів проникнення, яких у процесі дослідження виявили досить багато:

1. *MS*10-061 — уразливість у сервісі *Print Spooler*, яка дозволяє віддалено виконати довільний код. Вона була закрита завдяки своєчасній реакції з боку декількох антивірусних компаній.

2. *MS*08-067 — та сама уразливість, яка використовується *Conficker*, пов'язане зі службою *Server service*. Використовуючи цю вразливість вірус має можливість самостійно завантажити себе на машину жертви.

3. *MS*10-073 — вразливість у драйвері *win32k.sys*, що дає змогу підвищити локальні привілеї під управлінням *Win*2000/XP за допомогою виконання довільного коду в ядрі. Уразливість працює навіть під гостьовим обліковим записом, але реалізація шелл-кода для режиму ядра — завдання досить нетривіальне.

4. Відсутній *Vendor-ID* — уразливість у планувальнику завдань (*Task Scheduler*), що дозволяє підвищити локальні привілеї до рівня *SYSTEM* під управлінням *Vista/Win*7/*Win*2008. Для підвищення привілеїв буде достатньо навіть прав гостьового облікового запису. Уразливість цікава тим, що вона є концептуальною, розробники допустили її на рівні архітектури роботи цього сервісу.

5. *CVE*-2010-2772 — уразливість нульового дня, яка була знайдена в системах *Siemens Simatic WinCC* і *PCS* 7 *SCADA* і полягає вона в жорстко прошитому паролі для доступу до бази даних *Microsoft SQL* з програми *WinCC*.

Продуманість архітектури *Stuxnet* теж унікальна, масштаби реалізованого функціоналу вражають: більш мегабайта, вивіреного до дрібниць об'єктно-орієнтованого коду, скомпільованого в одній з останніх версій компілятора *Microsoft Visual* C ++.

І це тільки сам дропер, не рахуючи драйверів з руткіт-функціоналом. Велика кількість експортованих функцій, кожна з яких відповідає за свій функціонал.

Використовується перевірка різних ситуацій, які можуть вплинути на процес виконання коду. Мережна взаємодія з центром управління також виконана на високому рівні і реалізована у вигляді власного протоколу обміну. Крім цього реалізований механізм взаємодії за протоколом P2P який дозволяє знаходити інші заражені комп'ютери в локальній мережі, обмінюватися з ними інформацією і навіть оновлювати себе, якщо знайдено більш нові версії шкідливої програми.

2) **TDL4** — єдиний повноцінний руткіт для x64-систем, який вже встиг отримати широке розповсюдження. TDL4 являє собою подальший розвиток шкідливої програми TDL3. TDL4 вдається успішно обходити захисний механізм перевірки цифрового підпису в x64-версіях *Windows*. Автори застосували досить витончений спосіб обходу, який полягає в зараженні зони MBR і старту шкідливого коду раніше самої операційної системи. TDL3 використовувала для завантаження шкідливого функціоналу механізм інфікування системних драйверів без зміни їх розміру, але, оскільки в x64-системах при завантаженні драйверів перевіряється цифровий підпис, автори від цього механізму відмовилися.

Зараження здійснюється таким чином:

- відкривається описувач фізичного пристрою (наприклад: \\??\PhysicalDrive0), на якому розташовується розділ з ім'ям «C:»;
- готується і записується в кінець жорсткого диска образ своєї файлової системи (успадкований з TDL3);
- перезаписується MBR-кодом, який здійснює завантаження модулів з раніше підготовленої файлової системи;
- після успішного зараження на x64-системах відбувається перезавантаження за допомогою виклику WinAPI функції ExitWindowsExd() або ZwRaiseHardErrorl().

У цілому цей процес має такий вигляд:

- BIOS зчитує перший сектор завантажувального диска і передає управління на код головному завантажувальному запису MBR. Таким чином починається виконання коду TDL4;
- далі відбувається розшифрування коду для подальшого його виконання, який призначений для завантаження модуля з ім'ям ldr16 з файлової системи руткіта;
- завантажений модуль ldr16 здійснює перехоплення переривання 13h, яке відповідає за роботу з жорстким диском. Основне завдання даного модуля — визначити розрядність операційної системи x32 або x64 і залежно від неї, здійснити завантаження Ldr32 або Ldr64;
- обидва модулі — і ldr32 і ldr64, призначені для завантаження основного драйвера TDL4, яка

здійснюється без використання стандартного API, щоб обійти механізм перевірки цифрового підпису;

- спочатку відбувається розміщення коду драйвера в пам'яті за адресами, що належить ядру ОС. Далі проводиться його реєстрація як драйвера ОС за допомогою виклику недокументовані функції *loCreateDriverd*(). Після цього драйвер можна вважати завантаженим. Потім завантажується операційна система з драйвером TDL4 на борту, після старту відбувається впровадження шкідливого коду в деякі процеси, і надалі його поведінка дуже схожа на попередню версію — TDL3.

3) **TDL3** — являє собою справжній шедевр з області системного програмування. Мова йде про останню поширену версію цього руткіта 3.273. Оновлення свого функціоналу TDL3 отримано на початку 2010 року, а навесні було виправлено декілька помилок (одна з них — несумісність руткіт-драйвера з одним з патчів від *Microsoft*) і використаний ще один раніше невідомий прийом обходу *HIPS*-систем.

Обхід *HIPS*-систем. Використання *WinAPI*-функцій *AddPrintProcessor* і *AddPrintProvidor*, які не контролювалися більшістю *HIPS*-систем, дало можливість безперешкодної установки в систему шкідливого коду в обхід багатьох антивірусних систем і не тільки. Річ у тому, що захист заснований на такому принципі — прикриваються відомі вразливості, а інше виправляється на ходу. Це дає ефект захисту від поширених типових загроз, але як показує практика, ймовірність появи нових технологічних шкідливих програм далеко не нульова. Отже, у випадку TDL3 вдалося знайти довірені функції, які ніким не контролювалися і давали можливість встановити руткіт у систему.

Процес встановлення TDL3 можна розділити на дві стадії:

- реєстрацію допоміжної бібліотеки служби друку;
- завантаження драйвера режиму ядра.

Для того, щоб зареєструвати допоміжну бібліотеку служби друку, необхідно володіти привілеєм SE_LOAD_DRIVER_PRIVILEGE, що дозволяє завантажувати/вивантажувати драйвери. Щоб отримати даний привілей, дропер викликає *WinAPI*-функцію *RtlAdjustPrivilge*. Якщо виклик даної функції здійснено успішно, дропер копіює себе в каталог %*PrintProcessor*% як динамічна бібліотека та здійснює виклик функції *AddPrintProcessor*/*AddPrintProvidor*, передаючи їй як параметр ім'я скопійованої бібліотеки та рядок «tdl». Цей виклик за допомогою RPC-механізму змушує службу друку завантажити зазначену бібліотеку і викликати її точку входу. Ось таким витонченим способом TDL3 вдалося оминути більшість систем захисту на етапі встановлення. Про це вже знають всі антивірусні компанії і так чи інакше намагаються виявляти цей спосіб установки шкідливих програм.

Інфікування системних драйверів. Зараження системного драйвера знадобилося для того, щоб допомогти руткіту вижити після перезавантаження системи, оскільки прописувати себе в явному вигляді в гілку автозапуску — це занадто великий ризик бути виявленим ще до першого перезавантаження системи.

Ранні версії TDL3 заражали суворо певний файл-драйвер мініпорта жорсткого диска, на якому розташовується ОС. Але автори вирішили ускладнити процедуру видалення руткіта з системи — тепер він заражає випадково обраний драйвер. Вибравши драйвер для зараження, інфектор упроваджує в нього завантажувач TDL3 (невеликий фрагмент коду, завданням якого є завантаження тіла руткіта з диска і передача йому управління). Цікавим моментом є той факт, що модифікація драйвера відбувається без зміни його споконвічного розміру.

Власна файлова система. Ще однією цікавою особливістю TDL3 є реалізація власної файлової системи, яка є однією з основних функцій, які ускладнюють життя розробникам антивірусних засобів захисту. Файлова система створюється при зараженні системи і вона використовується для прихованого зберігання таких даних:

- модулів для впровадження в процеси (tdlcmd.dll);
- конфігураційної інформації (config.ini);
- тіла руткіта (tdll);
- перезаписаних ресурсів зараженого драйвера (rsrc.datl);
- додаткових файлів завантажених по мережі.

Свою файлову систему, яка починається з останнього логічного блоку диска (тобто з останнього сектору) і росте до його початку, TDL3 розташовує в кінці жорсткого диска, так що теоретично вона може перезаписати дані операційної системи.

4) **Dalixi** — китайський вірус, який запам'ятався цікавою технологією обходу HIPS при інсталяції в системі. Перед установкою своїх компонентів ця шкідлива програма відновлює таблицю системних сервісів ядра, а також нейтралізує *callback*-процедури, що викликаються ядром при створенні потоку або процесу та відображенні виконуваних образів в його

адресний простір. Останні активно використовуються різними HIPS та антивірусними продуктами для детектування шкідливого коду (за допомогою функцій:
PsSetLoadlmageNotifyRoutine,
PsSetCreateProcessNotifyRoutine.
PsCreateThreadNotifyRoutine).

Примітним є той факт, що ці дії відбуваються кодом, що працює в призначеному для користувача адресному просторі.

Для цих цілей *Dalixi* використовує не документовану функцію *ZwSystemDebugControl*, експортовану модулем *ntdll.dll*. Для того, щоб відновити вихідну таблицю системних сервісів, *Dalixi* відображає образ в користувальницьке адресний простір, ініціалізує таблицю з урахуванням таблиці переміщуваних елементів і за її допомогою перезаписує неї таблицю системних сервісів в ядрі. Аналогічним чином нейтралізуються *callback*-процедури.

5) **Zeus 2.x.x** — еволюція цієї троянської програми триває вже кілька років, а її участь була офіційно підтверджена в досить великій кількості гучних крадіжок грошей.

Вражає своїм різноманіттям і склад модулів, з яким поширюється *Zeus*, починаючи від цільових розширень під конкретні платіжні системи і закінчуючи можливістю установки VNC або модуля відправки повідомлень на зазначений обліковий запис *Jabber*.

Ще однією цікавою функціональною особливістю є наявність функціоналу для крадіжки X.509-сертифікатів з відповідними секретними ключами, які, як правило, використовуються для здійснення процедури цифрового підпису, в тому числі і для виконуваних модулів.

Працює цей модуль з використанням стандартної функції *CryptoAPI PFXImport-CertStore* для імпорту сертифікатів (цей же функціонал був наявний і в першій версії).

В антивірусних колах існує думка, що, можливо, саме за допомогою *Zeus* були вкрадені сертифікати, використані для підписання модулів «черв'яка» *Stuxnet*. Загалом, це справжній комбайн для крадіжки персональних даних, а модулі розширення роблять його справді універсальним інструментом у руках кіберзлочинців. Інформації про другу версію *Zeus* достатньо багато, але хотілося б звернути увагу на деякі цікаві нюанси, обхід фішингових фільтрів для *MS Internet Explorer* останніх версій. Або, наприклад, видалення всіх файлів користувача за командою з центру управління, а також зняття скріншота екрану в заданий момент часу.

Головною особливістю є те, що покупець може замовити розробку модуля, який буде «заточений» винятково під його завдання.

Нещодавно було проаналізовано цікаву модифікацію *Zeusa*, яка дозволяла через командний центр віддалено звертатися до пристроїв різного роду смарт-карт за допомогою вбудованого *Smartcard API*.

Висновки

З кожним роком ми спостерігаємо все більше ускладнення функціоналу шкідливого програмного забезпечення. У першу чергу це пов'язано з удосконаленням механізмів обходу захисних засобів і збільшенням часу життя кожної конкретно взятої зараженої машини.

ЛІТЕРАТУРА

1. *Stuxnet* Under the Microscope: Aleksandr Matrosov, Eugene Rodionov, David Harley — San Diego (USA), ESET, 2010. — 72 с.

2. *Алексеев П. П.* Антивирусы / П. П. Алексеев, А. П. Корш, Р. Г. Прокди. — М. : Наука и техника, 2010. — 80 с.

3. *Михайлов А. В.* Компьютерные вирусы и борьба с ними / А. В. Михайлов. — М. : Диалог-МИФИ, 2011. — 104 с.

Стаття надійшла до редакції 14.03.2011.