УДК 004.056.5

Р.М. Минайленко[1]
Кіровоградський національний технічний університет

Забезпечення захисту мережі від несанкціонованого доступу при перехваті API функцій

У наш час найбільшу поширеність одержали операційні системи сімейства Windows XXX. Вони широко використовуються не тільки як домашні системи, але і як сервери. Ця лінійка операційних систем відрізняється непоганою захищеністю від шкідливих програм, а також для них існує велика кількість додаткових систем безпеки (різні антивіруси, фаєрволи).

Встановивши антивірус і фаєрвол багато користувачів вважають, що вони стовідсотково захищені, і навіть більшість програмістів ввважають, що достатньо частіше перевіряти свій комп'ютер на підозрілі речі (автозавантаження, процеси і.т.ін.) і ніяка шкідлива програма до них не зможе проникнути. У більшості випадків це дійсно так, 99% «троянів» дотепер завантажуються через HKLM/Run, і мають назви подібні WinLoader32.exe.

Можна легко сховати присутність троянської програми в системі так, що навіть ретельна перевірка комп'ютера не дасть нічого. Можна легко одержати паролі на вхід у систему. Можна знищити антивіруси і обійти фаєрволи. Все це конкретні застосування технології, але цьому легко придумати й безліч інших застосувань (створення систем безпеки, різні емулятори). Можна на цій основі знімати тривіальні обмеження серійно з багатьох програм.

Перехоплення системної функції API полягає в зміні деякої адреси в пам'яті процесу або деякого коду в тілі функції таким чином, щоб при виклику цієї API функції керування передавалося не їй, а функції, яка використовується замість системної. Ця функція виконує заплановані користувачем дії, і потім, або викликає оригінальну функцію або не викликає її взагалі.

Проблема в тому, що Native API функції не документовані в SDK, але визначити модель їх виклику можна дизасемблювавши Kernel32.dll. Не можна стверджувати, що адреси функцій у системних бібліотеках не змінюються залежно від версії ОС, її комплектації або навіть конкретної ситуації. Це відбувається через те, що переважна база образу бібліотеки (dll preferred imagebase) є константою, яку можна змінювати при компіляції. Тому статичний імпорт функцій відбувається за іменем модуля та імененем функції (або її номера - ординала), який надається цим модулем. Завантажувач PE файлу аналізує таблицю імпорту й визначає адреси імпортованих функцій. У результаті в необхідному місці певної секції PE файлу (є як мінімум, атрибути "readable" й "initialized data") заповнюється масив адрес імпортованих функцій. У процесі роботи кожен модуль звертається до свого масиву для визначення місця входу в яку-небудь функцію.

[1] кандидат технічних наук, доцент кафедри програмного забезпечення