МЕТОДИКА ПЕРЕХОПЛЕННЯ СИСТЕМНИХ ФУНКЦІЙ ОПЕРАЦІЙНИХ СИСТЕМ СІМЕЙСТВА WINDOWS NT 5.X (2000/XP/Server 2003)

Бабенко І.В., Шевченко О.Г.
Донецький національний технічний університет

Більшість сучасних персональних ком'ютерів, що використовуються для особистого чи робочого користування, працюють під керуванням операційних систем сімейства Windows NT з п'ятою версією ядра, до складу яких входить Windows 2000, Windows XP та Windows Server 2003. Задля власної безпеки, виконанння користуватських програм у рамках цих операційних систем жорстко лімітоване: вони виконуються в обмеженому віртуальному адресному просторі та не мають права виконувати цілий ряд привієльованих інструкцій. Проте, на практиці іноді виникають завдання, що потребують підвищених привілеїв для виконання певної частини коду. Наприклад, програми налагодження драйверів, програми моніторингу системи, програми захисту від шкідливодійних програм (т.н. «вірусів») тощо. Інтеграція із системою значно підвисла би їх можливості та ефективність.

В цій доповіді розглядається механізм впровадження коду користувача в реалізацію функцій ядра системи Windows NT п'ятої версії, задля доповнюючої обробки системних запитів; а також проблеми обміну данними між впровадженим кодом та аплікацією користувача.

1. Отримання системних привілеїв

Існують декілька способів підвищити рівень привілеїв аплікацій, що виконується. Найбільш поширений спосіб – встановити у систему драйвер, який містить код, що потребує підвищених привілеїв. Проте цьому підходу властивий ряд недоліків: необхідно писати та реєструвати у системі драйвер, без якого програма не зможе працювати; ускладнений обмін даними між аплікацією та драйвером, адже ці дві задачі мають різний простір адрес. Проте, для аплікацій, що виконуються від імені облікового запису адміністратора, існує й інший спосіб, що позбавлений вищезазначених недоліків. Цей метод використовує об'єкт-секцію PhysicalMemory, що відображує усю оперативну пам'ять комп'ютера.

Простір імен ОС, де міститься ця секція, організований подібно до дерева файлової системи. Рядовій аплікації дозволений доступ лише до директорій \GLOBAL?? та \BaseNamedObjects, що містять переважно посилання на інші об'єкти з простору імен. Об'єкт-секція PhysicalMemory знаходиться у захищеній директорії \Device разом із об'єктами-драйверами різноманітних пристроїв. Доступ до цієї секції на читання та запис має лише сама ОС, проте дозвіл на модифікацію атрибутів секції має також група адміністраторів. Для решти облікових записів, ця директорія, разом з усіма її об'єктами не видима.

Алгоритм метода, що пропонується, достатньо простий. Аплікація, що виконується від імені облікового запису адміністратора, модифікує атрибути об'єкта \Device\PhysicalMemory таким чином, що секція становиться доступною як для читання, так і для запису. Після модифікації атрибутів відбувається її відкриття. Після того, як був отриманий хендл секції з потрібними правами, її атрибути рекомедується повернути у попередній стан задля більшої безпеки системи. Алгоритм реалізується за допомогою функції ZwOpenSection з библиотеки ntdll.dll, що автоматично завантажується у Windows NT з будь-якою аплікацією.

Далі, після отримання прав читання та запису до фізичної пам’яті, аплікація може встановити власний шлюз, що підвищує привілеї, у таблиці глобальних дескрипторів (GDT), точкою входу якого буде користувацька процедура, що містить код, який потребує системних привілеїв. Звернення на виконання до цього шлюзу спровокує визов цієї процедури.

2. Встановлення модуля перехоплення

Недокументовані низькорівневі сервіси операційної системи реалізовані у файлі ntoskrnl.exe [1]. Саме вони обслуговують усі запити користувача та операційної системи, й тому їх перехоплення представляє собою найбільшу цікавість. Їх виклик здійснюється за номером за допомогою звернення до системного переривання 2Eh (щось на зразок до int 21h для MS-DOS). Диспетчеризація викликів здійснюється за допомогою спеціальної таблиці KeServiceDescriptorTable, що також зберігається у файлі ntoskrnl.exe. Саме зміна адреси виклику у цій таблиці на адресу власного обробника спричиняє перехоплення певного сервісу.

3. Вирішення проблеми ізольованості адресного простору

Щоб не спричинити краху системи у момент виклику сервісу, що перехоплений, сторінку(ки), що містить код користувацького обробника, необхідно спроектувати до глобального адресного простору системи.

Система трансляції віртуальних адресів у x86 системах грунтується на дворівневій системі таблиць, що індивідуальна для кожного процесу. Перший рівень називається каталогом сторінок, другий – таблицею сторінок. Адресу початку каталога сторінок можна отримати із регістра CR3, для читання якого потрібні системні привілеї. Кожен елемент будь-якої з таблиць містить посилання на сторінку з даними. Для елементів каталогів сторінок такими данними є таблиці сторінок, а для таблиці сторінок – безпосередньо данні, що розташовані за конкретною віртуальною адресою. Тридцять два біти віртуальної адреси інтерпретуються як сукупність трьох елементів: індекс каталога сторінки (10 біт), індекс таблиці сторінки(10 біт), індекс байта. Отже, старші 20 біт дозволяють визначити фізичну адресу сторінки, а молодщі 12 біт – зсув у межах сторінки до потрібного байту [2]. Тобто, задля проектування власних сторінок пам’яті у простір системи необхідно: по-перше, визначити фізичну адресу процедури-перехоплювача; по-друге, знайти вільні елементи у однієї з таблиць сторінок, що у каталозі сторінок були помічена як глобальні; по-третє, заповнити цей елемент необхідними атрибутами безпеки та вказівником на фізичну сторінку пам’яті.

За допомогою вищенаведеної методики на прикладі функції NtWriteFile був розроблений та налагоджений механізм перехоплення низькорівневих функції ядра операційної системи. При розробці цього механізму у якості допоміжних засобів були також налагоджені методики отримання системних привілеїв та подолання обмежень віртуального адресного простору процесів. Ці здобутки можуть бути використані при подальшому дослідженні системи, а саме її недокументованої частини, написанні програм щодо моніторінгу системи загалом, чи певних її процесів.

Література

[1] Руссинович М., Соломон Д. “Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP и Windows 2000” /пер с англ., 4-е изд. – СПб.: Питер, 2005. – 992 с. ил.

[2] Рихтер Дж. “Windows для профессионалов: создание эффективных Win32 приложений с учетом специфики 64-разрядной версии Windows”/пер с англ., 4-е изд. – СПб.: Питер, 2001. – 752 с. ил.