УДК 004.454 + 004.02 + 004.7

РАЗРАБОТКА ПРОМЕЖУТОЧНОГО NDISД РАЙВЕРА

Игнатов Е.Ю.,Теплинский С.В.
Донецкий Национальный Технический Университет
кафедра компьютерных наук и технологий
E-mail: 0..255@mail.ru

Аннотация

Игнатов Е.Ю., Теплинский С.В. Разработка промежуточного NDIS драйвера. *Рассмотрены базовые концепции построения драйверов ОС Windows, рассмотрена спецификация интерфейса сетевых драйверов. Даны определения и изучены основы DoS-атак. Определен алгоритм работысетевого драйвера в случае атаки на компьютерную систему.*

Общая постановка проблемы

В последнее время остро встал вопрос о проблемах безопасности компьютерных систем.По данным Лаборатории Касперского, мощности DDoS-атак к 2012 году достигли новых высот (максимальная составила 600 Мбит/с или 1 100 000 пакетов/секунду (UDP-flood короткими пакетами по 64 байта). Таким образом, вопросызащиты от такого рода атакактуальны [4].

DoS-атака (атака типа «отказ в обслуживании», от англ.DenialofService) – атака на вычислительную систему с целью довести её до отказа, то есть создание условий, при которых доступ к предоставляемым системой ресурсам будет затруднён или невозможен. Если атака производится с большого числа компьютеров, она называется DDoS-атакой (от англ. DistributedDenialofService) [6].

Методы выявления DoS-атак делятся на следующие типы[6]:

- сигнатурные, основаны на качественном анализе трафика;
- статистические, основаны на количественном анализе трафика;
- гибридные, основаны на сочетании двух первых методов.

Алгоритм обработки пакетов в данном случае построен по гибридному методу. В данном случае он состоит из 2х этапов. Первый основан на количественном анализе потока:

- отброс большого количества мелких пакетов;
- отброс пакетов от не доверенных источников;

Далее идёт следующий этап – анализ содержимого пакетов. На данном этапе может стоять фильтр пакетов с одинаковым содержимым, либо анализ содержимого по словарю и отброс вредных пакетов.

В случае DoS-атаки(от одного сервера), данные пакеты отсекаются без анализа и сам сервер добавляется в черный список. При получении одинаковых пакетов с разных серверов(DDoS) в близкие моменты времени блокируются все данные сервера. При необходимости, компьютерная система перенастраивает адаптер на другой IP адрес, а по возможности и перебрасывает все пакеты на другой адаптер, зарегистрированный на другую ветвь компьютерной сети.

Для реализации алгоритмов защиты от DoS-атак необходим промежуточный модуль, который мог бы выявлять данные атаки и либо блокировать поток пакетов, либо переносить атакуемый ресурс (компьютерную систему) на иное месторасположение в сети, то есть менять адрес. Естественно, самым оптимальным решением такой проблемы было бы

подключение отдельного firewall-устройства с необходимым функционалом на маршруте «Интернет – Компьютерная система», преобразовавего в цепочку «Интернет – Firewall– Компьютерная система». Данный подход является самым оптимальным, так как он не создает дополнительных нагрузок на саму компьютерную систему.При наличии в данном устройстве мощных процессорных элементов, задержка пакетов будет минимальной.

Однако такого типа устройства с максимальными характеристиками имеют и минус – высокая стоимость. Самые простые вариации таких устройств стоят не менее 500$, а устройства, характеризующиеся высокими показателями – от 50000$ [5]. Следовательно, надо искать другое решение.В данной работе предлагается программа выявления и защиты от атак на основе промежуточногоNDIS драйвера. Плюсами данного подхода будут следующие факторы:

- потребляет малое количество вычислительных ресурсов при отсутствии сетевых потоков либо при наличии активности только доверенных потоков;
- блокирует пакеты DoS-атак до момента поступления их к прикладному ПО либо операционной системе;
- firewallтакого типа позволяет в дальнейшем дорабатывать функции защиты;
- при реализации NDISдрайвера есть возможность переброса пакетов на виртуальный MAC/IP для их утилизации,минуя ядро операционной системы.

ОС Windowsорганизует драйвера в виде дерева устройств (рис.1), в котором на общий корневой уровень (Root) цепляются все основные устройства/шины, уровнем выше располагаются устройства, взаимодействующие с драйверами первого уровня и т.д. На верхнем уровне, как правило, располагаются конечные драйвера реальных устройств, либо драйвера концентраторов/портов, ожидающих подключение устройств. После уровня шин, драйвера начинают выстраиваться в одной ветви в виде стека драйверов устройства.

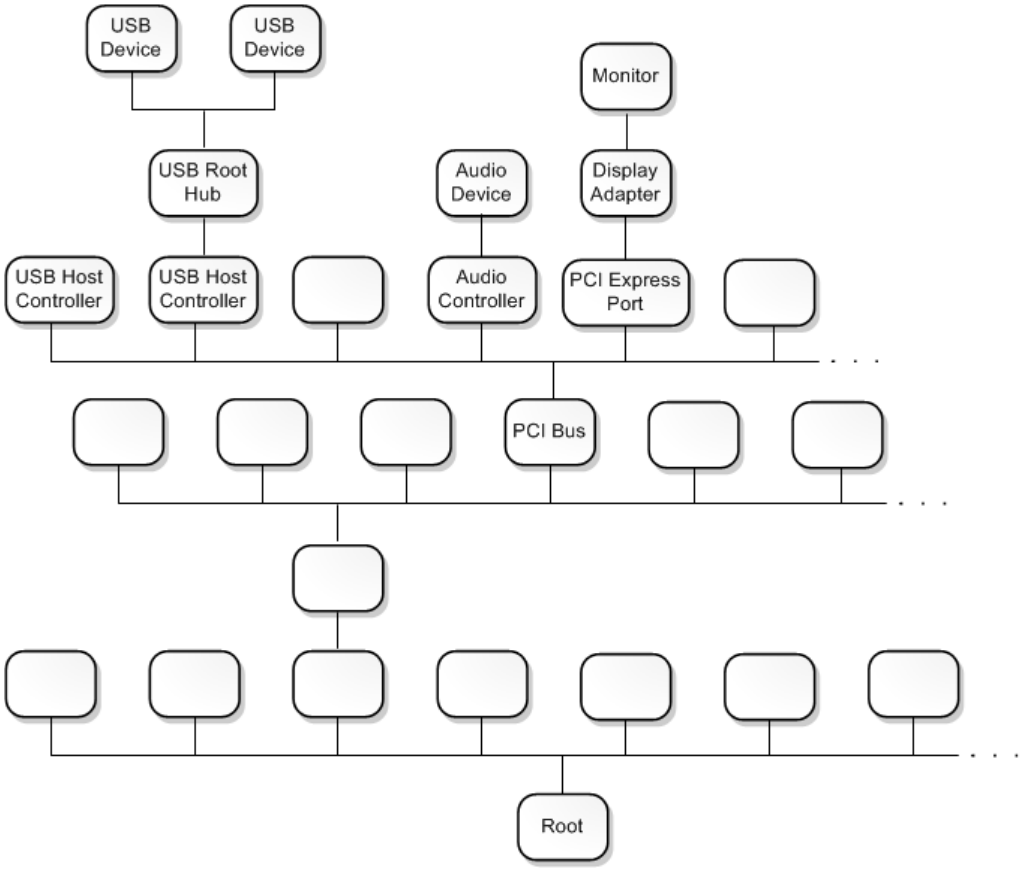

Рисунок 1 – Дерево драйверов ОС Windows

Драйвера принимают запросы и обрабатывают их. Некоторые запросы могут обрабатываться несколькими стеками драйверов. Данная общая концепция драйверов справедлива и к сетевым драйверам.

NDIS – спецификация интерфейса сетевых драйверов, спецификация интерфейса MACуровня, разработана совместно Microsoftи 3Comдля драйверов ЛВС. Скрывает особенности реализации сетевого адаптера (NIC) от сетевой ОС. Драйвер, написанный в этом стандарте, может поддерживать транспортный протокол связи для всех сетевых адаптеров.

NDIS-драйвера являются некоторого рода посредниками между аппаратурой и низкоуровневыми сетевыми драйверами (рис.2).NDIS-драйвера бывают трёх типов: минипорт-драйверы, промежуточные драйверы и протокольные драйверы.

Минипорт-драйверы составляют самый низкий уровень сетевых драйверов. Они могут получать некоторые команды. Основные действия: инициализация устройства, управление сетевыми подключениями, получение и отправка пакетов, изменение параметров или перезагрузка адаптера. Одна из полезных особенностей минипорт-драйверов – это наличие функции FFP (FastForwardingPath), которая позволяет адаптерам маршрутизировать/фильтровать пакетыаппаратно, без участия ОС и не нагружая основные процессоры компьютерной системы.

Промежуточные драйверы представляют собой нечто среднее между минипорт-драйверами и протокольными драйверами. Они позволяют фильтровать входящие пакеты, могут перенаправлять их между сетями и распределяют работу адаптера между процессами компьютерной системы.

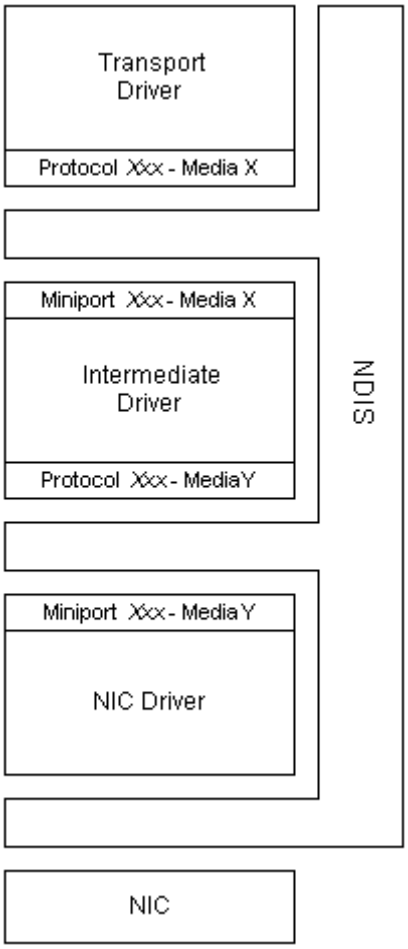

Рисунок 2 –Структура интерфейса сетевых драйверов

Протокольные драйверы работают непосредственно с пакетами – выделяют для них ресурсы, формируют и отправляют.

Как видно из описания, для задачи вполне хватает функций промежуточного сетевого драйвера. Краткий алгоритм работы данного драйвера следующий:

- привязка драйвера к адаптеру, регистрация всех необходимых функций;
- приём всех пакетов, пришедших на интерфейс;
- отправка копий трафика на исследование, а оригиналы пакетов отсылаются далее прикладным программам.

Драйвера NDIS-интерфейсаиспользуютопределенный набор функций. Вначале регистрируется протокол, сам процесс описан в теле функции DriverEntry(). Заполняются все поля структуры протоколаNDIS_PROTOCOL_CHARACTERISTICS, а именно назначается соответствие функциям драйвера и событиям на адаптере.После успешной регистрации протокола мы можем вызывать функцию NdisOpenAdapter(), которая соединяет драйвер с указанным интерфейсом. Регистрируются функции приёма и передачи пакетов (Receive/Send), а также функция на событие TransferDataCompleteHandler–которая вызывается при окончании приёма блока пакетов и будет возвращатьзаголовок и содержимое пакета. Вызов функции копирования пакетов размещается в конце тела функции TransferDataDone().Также тут описаны функции на события выгрузки протокола UnloadHandler(), и функции обработки статусов адаптера. Завершает структуру поле закрытия адаптера, которое связано с функциейCloseAdapter().

После заполнения структуры протокола, следует заполнение структуры минипортаNDIS_MINIPORT_CHARACTERISTICS.Онасодержитполяобработкипрерыванийа даптераENABLE_INTERRUPT_HANDLER /DISABLE_INTERRUPT_HANDLER, поля функцийзапуска / перезапуска/останова/переконфигурации / завершения работы адаптераINITIALIZE_HANDLER / RESET_HANDLER / HALT_HANDLER /RECONFIGURE_HANDLER /SHUTDOWN_HANDLERсоответственно, а также поля функции приёма/обработки пакетов SEND_PACKETS / TRANSFER_DATA. В теле функции TransferDataDone() будет размещён модуль количественного анализа пакетов.

После всего вызывается функция выгрузки драйвера DriverUnload().

Выводы

Результатом выполненной работы представлен драйвером сетевого интерфейса, который позволяет фильтровать входящие пакеты при выявлении DoS-атак. Результаты деятельности драйвера сохраняются в протоколе.Успешно проведены эксперименты по обнаружению и защите от атаки TCPSYNFloodи TCPFlood. Улучшить данную программу можно, добавив модуль анализа ICMP и UDPпротоколов.

Список литературы

1. ПенниОрвик, ГайСмит.WindowsDriverFoundation. Разработка драйверов. – Спб.: «БХВ_Петербург», 2008. – 880с.

2. MicrosoftDevelopmentNetwork [Electronicresourse] / Интернет-ресурс.: www/ URL: http://msdn.microsoft.com

3. NDIS.com / Интернет-ресурс.: www/URL: http://ndis.com

4. Лаборатория Касперского / Интернет-ресурс.: www/URL: http://www.securelist.com/ /ru/analysis/208050745/DDoS_ataki_vtorogo_polugodiya_2011_goda

5. Radware Defense Pro / Интернет-ресурс.: www/URL: http://www.infobezpeka.com/ /products/apatnye/?view=395

6. Хакер. Wiki. / Интернет-ресурс.: www/URL: http://wiki.xakep.ru/ /otkaz_v_obsluzhivanii.ashx