глубины и вершины сетевого стека висты

крис касперски ака мыщъх, no-email

сетевой стек висты был переписан с чистого листа. о лучшем подарке от Microsoft хакеры не могли и мечтать. раз новый — значит еще сырой и не протестированный. анализируя ранние бета-версии висты, исследовательская лаборатория корпорации Symantec обнаружила огромное количество дыр, что уже указывает на кромешное качество кодирования. сколько "сюрпризов" таится внутри этой твари никто не знает, а, значит, хакерам будет во что вонзить зубы!

введение

Устав латать старый сетевой стек, доставшийся ей в наследство еще от NT, команда разработчиков висты решила, что переписав его с нуля она достигнет выдающегося уровня безопасности, попутно повысив производительность и реализовав ряд дополнительных фич. Весьма нехилое решение надо сказать. Сетевой стек это грандиозное сооружение, к проектированию которого нужно подходить с головой, а умных людей в Microsoft с каждым годом становится все меньше и меньше. Печально, но факт, бесспорным доказательством которого служит новый сетевой стек, повторяющий ошибки десятилетней давности и добавляющий к ним целое полчище новых, многие из которых гнездятся на концептуальном уровне и будут исправлены нескоро (если будут исправлены вообще).

Вопреки всем заверениям Microsoft, **сетевой стек висты намного менее надежен и безопасен, чем в XP**, к тому же он совершенно не изучен и абсолютно непредсказуем. У администраторов нет опыта решения проблем, с которыми они прежде не сталкивались, разработчики защитных компонентов (от программных брандмауэров до аппаратных комплексов) еще не включили поддержку висты и ее протоколов в свои продукты.

Во времена NT, когда ядро и прилегающие к нему компоненты, не менялись раз в квартал, существовали сетевые стеки от сторонних производителей и, между прочим, неплохо работали, успешно конкурируя с Microsoft, но теперь "политическая" ситуация изменилась и у нас остался только один путь — Microsoft way, ведущий непосредственно в Hell. Только плазмогана там не будет и от хакеров придется обороняться разве что кулаком да кастетом.

Сетевой стек тесно интегрирован с осью и "отодрать" его, вернув старый стек на место никакой возможности нет. Нам предлагают множество новых компонентов, причем это предложение из разряда тех, от которого невозможно отказаться — ведь ни отключить, ни заблокировать ненужные фичи все равно нельзя. То есть можно конечно, но отнюдь не через графический интерфейс и большинство пользователей этого сделать не сможет, а, значит, черви, хакеры и удаленные атаки будут процветать.

Рисунок 1 Microsoft way

что нового в сетевом стеке висты

Главным и, пожалуй, единственным достижением Microsoft'а стала интеграция IPv4 и IPv6 в единый стек (до этого они были реализованы как раздельные компоненты), что и плохо, и хорошо одновременно. Хорошо то, что конечный пользователь получает готовый IPv6 без всякой головой боли и установки дополнительных пакетов. Катастрофическая нехватка IP-адресов с каждым сезоном ощущается все острее и острее, но переход на IPv6 сдерживается как необходимостью смены сетевого оборудования, так и обновлением серверных и клиентских осей. В исторической перспективе переход на IPv6 неизбежен. Это ясно всем, но почему же

нельзя реализовать его поддержку опционально, как это сделано во всех "правильных" системах, той же xBSD например?

Большинству _сегодняшних_ пользователей IPv6 _совершенно_ не нужен, поскольку для локальной сети и IPv4 хватает с лихвой, а основная масса провайдеров еще не поддерживает IPv6 и в обозримом будущем переходить на него не собирается. Проблема в том, что IPv6 несет в себе множество нововведений, еще не обкатанных и не протестированных в платентарном масштабе. Это настоящий кот в мешке, от которого можно ожидать всего чего угодно. Новых проблем, новых атак…

Теоретически (и только теоретически!) IPv6 обеспечивает более высокую производительность и лучшую защиту от атак, но практически вопросы производительности решаются "тонкой" настройкой опций TCP/IP протокола, которые в Windows доступны лишь частично, и крайне отрывочно "документированы" в виде заметок в Knowledge Base. TCP/IP протокол это не трактор — сел, завел и поехал. Над ним еще поколдовать нужно, учитывая ширину канала, характер запросов и т. д.

Настойки по умолчанию стремятся удовлетворить сразу всех и каждого, в результате чего, по-настоящему не остается не удовлетворен **никто**. Отсутствие _легальных_ рычагов управления, не позволяет оценивать реальную производительность сетевого стека, и громкие заявления Microsoft'a, что в висте стек намного более производителен следует расценивать не более как пропаганду. Результаты тестов ни о чем не говорят! Откуда мы знаем, может Microsoft просто подкрутила настойки, чтобы виста выдавала лучше результаты при испытаниях на заранее подготовленном полигоне? Тем более, что в большинстве случаев, реальный CPS определяется отнюдь не "качеством" сетевого стека, а загруженностью удаленного сервера, пропускной способностью каналов связи и т. д. Глупо ожидать, что установив висту на свой компьютер, мы "разгоним" свой модем хотя бы на десяток процентов…

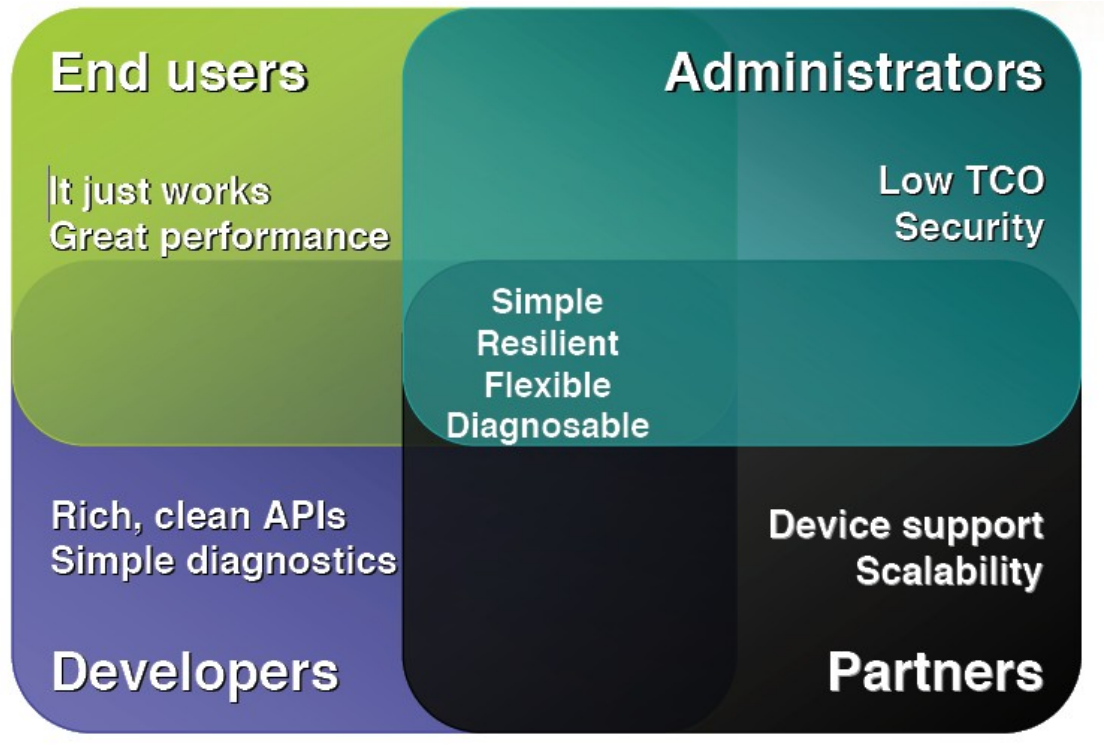

Рисунок 2 фрагмент из рекламного буклета Microsoft, показывающий какой хороший у нее получился сетевой стек и как замечательно нам с ним будет жить

хакеры штурмуют тоннели

Следствием интеграции IPv4 с IPv6 в единый сетевой стек, стало появление туннельных протоколов Teredo, ISATAP, 6to4 и 6over4, причем Teredo уже успел попасть в RFC и осесть под номером 4380 (http://www.rfc-editor.org/rfc/rfc4380.txt). Разжеванное описание на понятным даже для неспециалистов языке можно найти в статье "Teredo Overview", опубликованной на Microsoft Tech Net: http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/teredo.mspx.

Говоря на пальцев (ну типа сурдоперевод для глухонемых), Teredo инкапсулирует (упаковывает) IPv6 трафик внутрь IPv4 пакетов (см. рис. 3), используя протокол UDP, слабости и недостатки которого хорошо известны. Если два IPv6 узла разделены IPv4 сегментом сети (наиболее типичная на сегодняшний день конфигурация), висте задействует Teredo, направляя запрос одному из публичных Teredo-серверов, который в свою очередь передает его получателю, фактически выполняя роль proxy-сервера.

Рисунок 3 структура заголовка Teredo-пакета, с "зашифрованным" (obscured) истинным целевым портом и адресом

Самое интересное, что Teredo позволяет обходить трансляторы сетевых адресов (они же NAT'ы) и брандмауэры, причем это не баг, а фича. Рассмотрим два узла, защищенные NAT'ом, прямое взаимодействие между которыми невозможно. Но это оно по IPv4 невозможно, а если использовать Teredo-туннель, то истинный адрес и порт назначения окажется скрыт в Teredo-заголовке, а в IPv4 попадает адрес узла, "смотрящего" в Интернет и UDP-порт самого Teredo (3544), который, конечно, можно и закрыть, но как же тогда с остальными виста клиентами общаться?! Поскольку NAT не может установить ни реального целевого адреса, ни реального целевого порта протокола IPv6, он беспрепятственно пропускает IPv4 пакет. Это же самое относится и к другим защитным механизмам, не поддерживающим протокола Teredo.

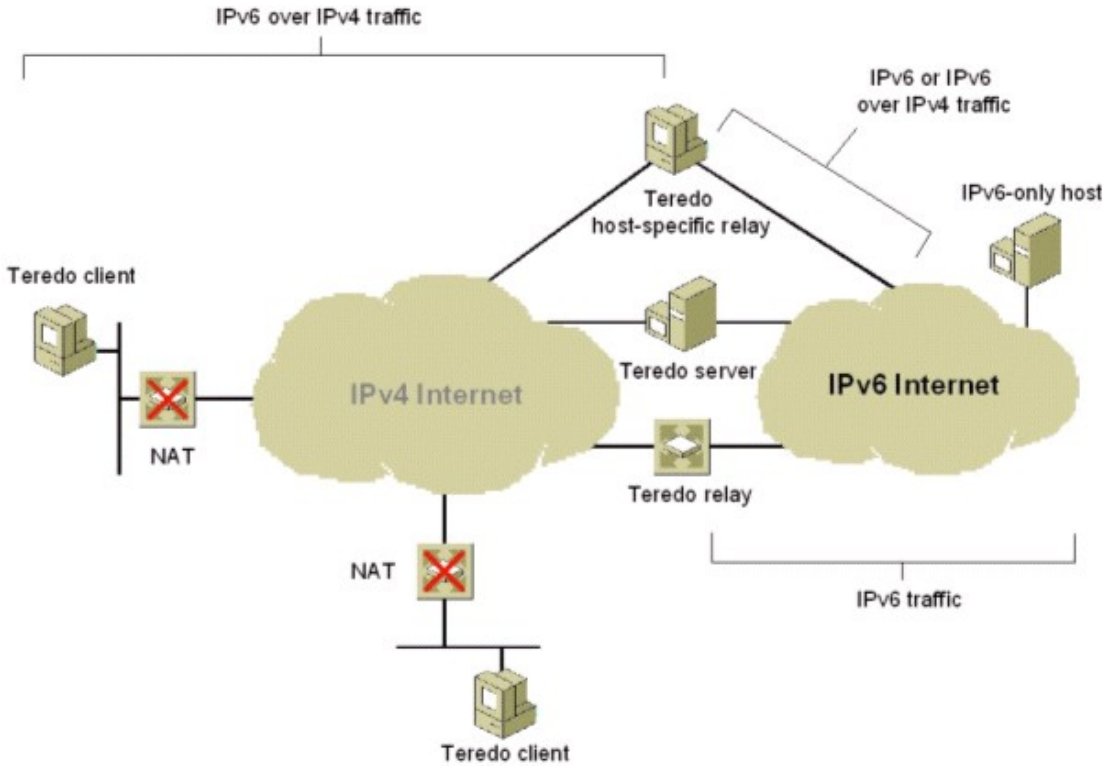

Рисунок 4 обход NAT'а при помощи протокола Teredo

Но это еще что! Поддержка новых протоколов — всего лишь вопрос времени. Переход на висту означает переход на Teredo, а переход на Teredo навязывает глобальную маршрутизацию, заставляющую забыть о приватных IP-адресах, использующихся в локальных сетях и невидимых снаружи. Ну это раньше они были невидимы, а теперь… Помнится, что когда один паренек просканировал внутреннюю сеть Пентагона и "добыл" приватные адреса, Пентагон так перестремался, что довел дело до суда. Оно и понятно. Чем больше хакер знает о структуре защищенной сети, тем проще ее атаковать.

В довершении ко всем несчастьям, виста поддерживает инкапсуляцию IPv4 в IPv4 и IPv6 в IPv6, что позволяет скрывать истинные целевые адреса и порты, вынуждая брандмауэры

и другие защитные средства проводить скрупулезный анализ трафика, а это сразу же увеличивает потребности в памяти и мощности процессора со всеми вытекающими отсюда последствиями.

Да… с появлением виста-узлов в господствующем IPv4 Интернете следует ожидать больших потрясений. Вот такая она, безопасность, которую нам обещают!

TCP/IP и его свита

На самом деле, TCP/IP никогда не используются в "чистом" виде и всегда окружены свитой вспомогательных протоколов, причем, далеко не все из них нужды домашнему пользователю, но… увы! Привычка Microsoft пихать все в одну коробку без возможности отделить одно от другого дает о себе знать и мы не можем удалить лишние протоколы, которые не только жрут системные ресурсы, но еще и служат источником потенциальных ошибок, дыр и люков, в существовании которых можно не сомневаться.

По этому поводу вспоминается один хороший анекдот: заходит Билл в Мак-Дональдс, заказывает: колу, биг-мак с картошкой фрии и… получает все это в одном стакане, безнадежно перемешанное друг с другом и еще кучей других непонятных вещей (мух, тараканов), которые Билл совсем не заказывал, и был бы раз отказаться от них, если бы только знал как.

Вот так же и здесь. Сетевой стек висты включает в себя следующие протоколы:

- ICMP;
- IGMP;
- IPv4;
- IPv6;
- ICMPv6,
- TCP;
- UDP;
- IP6;
- GRE;
- ESP;
- AH;
- 43;
- 44;
- 249;
- 251;

Порывшись в RFC, легко убедиться, что добрая половина протоколов не нужна не только рабочим станциям, но и серверам, а многие из них даже не имеют собственного имени, ограничиваясь только номером. В частности, протоколы 43 и 44 отвечают за маршрутизацию и фрагментацию в IPv6. Причем, в ранних бетах посылка мусора по 43 протоколу вводила висту в глубокую задумчивость, граничащую с суицидом, но через некоторое время она все-таки возвращалась к обработке сетевых запросов, как ни в чем ни бывало. А вот мусор, переданный по 44 протоколу обрушивал систему в голубой экран смерти. Сейчас это уже исправлено, но неизвестно сколько еще ошибок реализации предстоит узнать.

Алгоритм сборки IP-пакетов изменился в _худшую_ сторону и частично перекрывающиеся пакеты теперь безжалостно отбраковываются как неверные, порочные и вообще недостойные для существования (на самом деле, программистам лень было топтать клавиатуру, вот они и "срезали углы"). Исключение составляет ситуация, когда два пакета перекрываются на 100%, тогда отбрасывается последний пакет в пользу первого, хотя LINUX системы поступают с точностью до наоборот. Вообще же говоря, проблем со сборкой перекрывающихся пакетов у всех систем хватает и каждая из них имеет свои особенности, в результате чего принятые данные искажаются вплоть или пакет не собирается вообще! Рисунок, приведенный ниже, показывает порядок сборки UDP пакета, состоящего из нескольких перекрывающихся фрагментов. Как видно, виста оказывается далеко не на высоте.

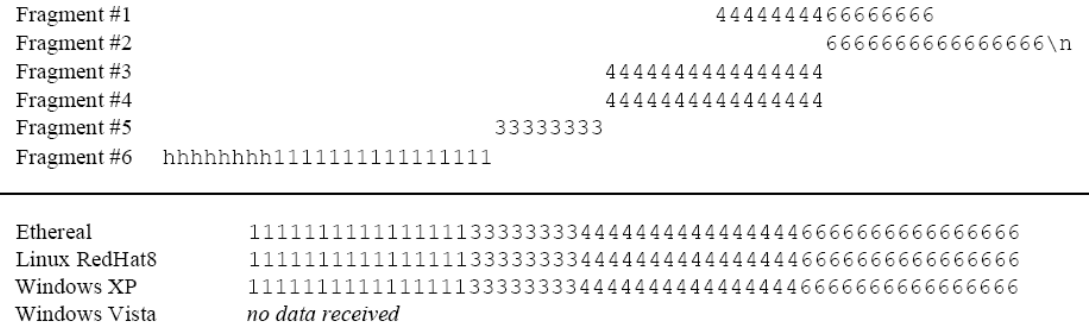

Рисунок 5 результат сборки UDP-пакета с перекрывающимися фрагментами на разных операционных системах

новые TCP/UDP порты

Нормальный клиентский узел вообще не должен содержать никаких открытых TCP/UDP портов! Он даже может не обрабатывать ICMP-сообщения, в частности, игнорировать echo-запросы (на чем основан ping) и не отправлять уведомлений о "казне" пакета с "просроченным" TTL (на чем основана работа утилиты tracert), хотя все это считается дурным тоном и создает больше проблем, чем их решает.

Наличие открытых портов указывает на присутствие серверных служб, обслуживающих удаленных клиентов. Каждая такая служба — потенциальный источник дыр, переполняющихся буферов и прочих лазеек, через которые просачиваются черви, а воинствующие хакеры берут компьютер на абордаж.

Вот неполный список портов, открываемых системой в конфигурации по умолчанию:

- IPv6 UDP
  - MS-RPC (135)
  - NTP (123)
  - SMB (445)
  - ISAKMP (500)
  - UPnP (1900)
  - Web Services Discovery (3702)
  - Windows Collaboration (54745)
  - совместный доступ к файлам и принтерам (137, 138)
  - порты-призраки: (49767, 62133)
- IPv4 UDP
  - Teredo (4380, 61587)
  - MS-RPC (135)
  - SMB (445)
  - NBT (139)
  - PNRP 3540
- IPv4 TCP
  - P2P Grouping Meetings (3587)
  - Windows Collaboration (54744)
  - совместный доступ к файлам и принтерам (137, 138)

Изобилие открытых портов подогревает хакерский интерес и создает серьезную угрозу безопасности. Только наивный может верить в то, что все эти сервисы реализованы без ошибок, тем более, что ошибки уже начинают выползать. Как видно, IPv6 отображает часть UDP-портов на IPv4, однако, забывает "объяснить" этот факт своему же собственному брандмауэру и если мы закрываем печально известный 135-порт на IPv4, его необходимо закрыть так же и на IPv6, равно, как и наоборот.

В ранних бетах факт закрытия портов было очень легко обнаружить, поскольку при попытке установки соединения с несуществующим портом система возвращала пакет с флагом RST (как, собственно, и положено делать по RFC). Соответственно, порты не возвратившие

пакета с таким флагом, но и не установившее соединения, все-таки существуют, но закрыты брандмауэром, который можно легко обойти, например, через RPC. Правда, эта лазейка была быстро закрыта, но зато при отправке сообщения на несуществующий UDP IPv6 порт до сих пор возвращается ICMPv6 сообщение об ошибке, опять-таки позволяющее отличить отсутствующие порты от портов, закрытых брандмауэром.

```
linux# nmap -P0 -sT -p1-65535 10.200.200.127
PORT STATE SERVICE
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds
49152/tcp filtered unknown
49153/tcp filtered unknown
49154/tcp filtered unknown
49155/tcp filtered unknown
49156/tcp filtered unknown
49157/tcp filtered unknown
```

Листинг 1 успешное сканирование портов, закрытых встроенным в висту брандмауэром

Протокол, обеспечивающий совместный доступ к файлам и принтерам — SMB, так же полностью переписан и представлен в инаугурации новой версии — SMB2, ориентированной на передачу больших файлов данных и как будто бы обеспечивающий лучшую производительность, однако, реализованный далеко не самым лучшим образом. В частности, засылка мусора в порт 445 обрушивала бету build 5270 в голубой экран смерти, и была исправлена только в следующей версии.

Механизмы аутентификации, вызывающие множество нареканий еще со времен 9x, похоже не претерпели никаких радикальных изменений, откатившись назад в мрачную готическую тьму средневековья, когда нестандартные клиенты типа SAMBA предоставляли доступ ко многим защищенным ресурсам не требуя авторизации. Помнится, реакция Microsoft была такова: "SAMBA – это _неправильный_ клиент, пользуйтесь штатными средствами Windows и у вас не будет никаких проблем". Похоже, парни из Рейдмонда не понимают чем клиент отличается от сервера и до сих пор не въезжают в тему. А может, просто трава такая попалась. Термоядерная. Уж всяко не местная подзаборная. Иначе чем можно объяснить тот факт, что протокол SMB держит для своих внутренних целей именованный канал (по-английски — pipe) "IPC$", через который можно подключаться к ресурсам netlogon, lsarpc и samr _без_ аутентификации! В SMB2 этот список пополнился: "protected_storage" и "lsass", что легко проверить с помощью скрипта trypipes.py, входящего в состав знаменитого хакерского набора dcerpc:

```
xp> net use \\10.200.200.123\ipc$ /u:"" ""
xp> c:\python24\python trypipes.py -m 10.200.200.123 pipes.txt
\\10.200.200.123\PIPE\netlogon
\\10.200.200.123\PIPE\lsarpc
\\10.200.200.123\PIPE\samr
```

Листинг 2 перечисление именованных каналов протокола SMB

```
vista> c:\python24\python trypipes.py -m 10.200.200.123 pipes.txt
\\10.200.200.123\PIPE\lsass
\\10.200.200.123\PIPE\protected_storage
\\10.200.200.123\PIPE\netlogon
\\10.200.200.123\PIPE\lsarpc
\\10.200.200.123\PIPE\samr
```

Листинг 3 перечисление именованных каналов протокола SMB2

Механизм именованных каналов тесно связан со столь горячо любимым Microsoft'ом механизмов удаленного вызова процедур Remote Procedure Call или, сокращенно, RPC, через который распространялся MSBlast и другие черви подобного типа. В висте до сих пор сохранилась возможность определять список доступных интерфейсов и вызывать некоторые из них (ServerAlive2, OXIDResolver, etc) и все это безо всякой авторизации!

глобализация ARP

В локальных Ethernet сетях на физическом уровне используется MAC-адресация, поверх которой натягивается TCP/IP, использующий IP-адресацию. В результате чего, каждый узел имеет как минимум один MAC-адрес и один IP-адрес, которые никак не связаны с друг

другом и чтобы отправленный пакет дошел до места назначения, маршрутизатору необходимо иметь таблицу соответствия IP- и MAC-адресов, которая динамически создается при помощи протокола ARP. Грубо говоря, в сеть посылается широковещательный запрос: "обладатель такого-то IP, сообщите своей MAC-адрес!". Никакой аутентификации при этом не производится и присвоить себе чужой IP — плевое дело. Атаки такого типа давно изучены и подобно описаны. Хакер может: разрывать TCP/IP соединения, установленные жертвой, перехватывать трафик, выдавать себя за другой узел и т. д. Но все это — строго в рамках локальной сети, причем предыдущие версии Windows, обнаружив, что хакер захватил их IP-адрес выплевывали на экран предупреждение. Виста же просто отмечает этот факт в системном журнале и… прекращает реагировать на сетевые запросы.

Но внутрисетевые атаки это еще куда бы то ни шло. Существует масса способов вычислить злоумышленника, подняв по тревоге бригаду каратистов быстрого реагирования, доходчиво объясняющих незадачливому хакеру, что лучше уйти по собственному, чем всю жизнь работать на больницу. Заботясь о пользователях, тьфу, о хакерах, Microsoft добавила новый протокол для разрешения адресов — Neighbor Discovery или, сокращенно ND, доступный извне локальной сети. И хотя реализация удаленной атаки сопряжена с рядом трудностей, она все-таки осуществима! Система уязвима только во время так называемой probe-фазы, в течении которой происходит ожидание отклика от "соседних" (neighbor) узлов. Все остальное время, поддельные пакеты, посланные злоумышленником игнорируются. Однако, учитывая значительную продолжительность probe-фазы, а так же ее высокую периодичность, хакеру даже не понадобится запасаться терпением! Ну разве за пивом сгонять, пока его компьютер методично бомбардирует жертву запросами.

Трудность номер два: ND-пакет содержит специальный счетчик, начальное значение которого равно 255, и при пересылке через каждый узел оно уменьшается на единицу, таким образом, атакующий должен находится достаточно близко от жертвы. "Близко", естественно, не в географическом смысле. Атаковать американские компьютеры из российской глубинки вполне реально. А там попробуй найти Васю Пупкина из деревни Нью Васюки!

идентификация сетевого стека висты

Сетевой стек всякой операционной системы имеет свои особенности реализации (они же fingerprint — отпечатки пальцев), позволяющие идентифицировать жертву, что значительно упрощает атаку, поскольку хакер заранее знает какие дыры там есть и какие действия предпринимать.

Снять отпечатки пальцев (также называемые "сигнатурой") с удаленного узла можно, например, с помощью знаменитой утилиты nmap. В частности, для Server Longhorn они выглядят так:

```
linux# nmap -sT -p 445,999 -O 10.200.200.124
Fingerprint Microsoft Windows Longhorn eval build 4051
Class Microsoft | Windows || general purpose
TSeq(Class=TR%gcd=<6%IPID=I%TS=100HZ)
T1(DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T2(Resp=Y%DF=Y%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=0%ACK=O%Flags=AR%Ops=)
T4(DF=Y%W=0%ACK=O%Flags=R%Ops=)
T5(DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
T6(DF=Y%W=0%ACK=O%Flags=R%Ops=)
T7(DF=Y%W=0%ACK=S++%Flags=AR%Ops=)
PU(DF=N%TOS=0%IPLEN=164%RIPTL=148%RID=E|F%RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

Листинг 4 "отпечаток пальцев" сетевого стека висты

А вот сигнатура Server 2003 (к слову сказать, XP SP2 ведет себя точно так же):

```
Fingerprint Microsoft Windows 2003 Server or XP SP2
Class Microsoft | Windows | 2003/.NET | general purpose
Class Microsoft | Windows | NT/2K/XP | general purpose
TSeq(Class=TR%gcd=<6%IPID=I)
T1(DF=Y%W=402E|FB8B%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=402E|FB8B%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
```

```
PU(DF=N%TOS=0%IPLEN=B0%RIPTL=148%RID=E%RIPCK=E|F%UCK=E|F%ULEN=134%DAT=E)
```

Листинг 5 "отпечаток пальцев" сетевого стека Server 2003/XP SP2

Как говорится — почувствуйте разницу! Еще можно послать TCP-пакет с перекрывающимися фрагментами и посмотреть на результат его сборки.

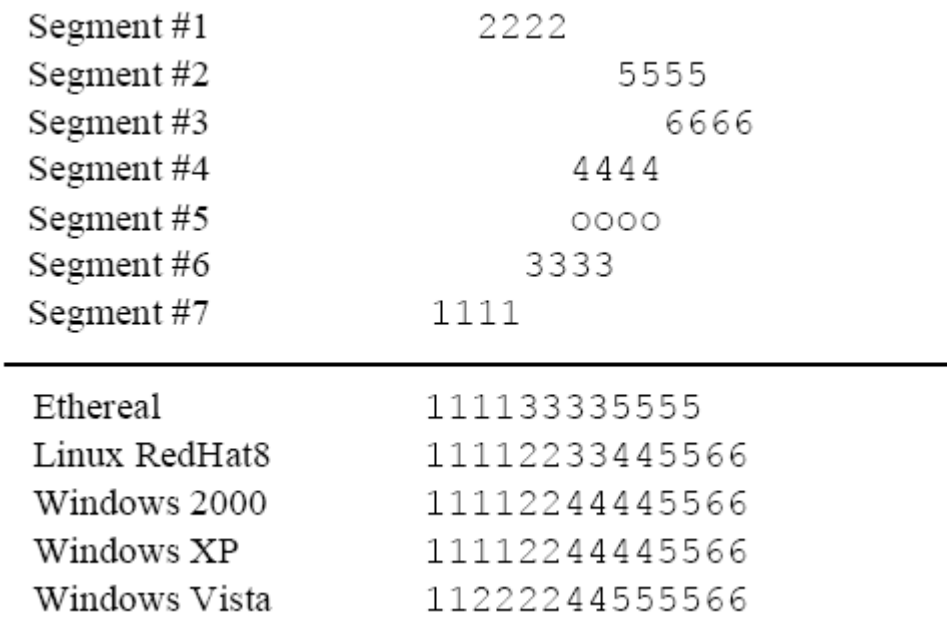

Рисунок 6 результат сборки TCP-пакета с перекрывающимися фрагментами на различных операционных системах

заключение

Переход на висту несомненно сулит большие перспективы. Для хакеров. А так же для всех сторонних разработчиков, предлагающие защитные комплексы разной степени сложности и… стоимости. В проигрыше остаются только домашние пользователи и администраторы, впрочем, администраторы будут сопротивляться переходу на висту изо всех сил и, быть может, массового перехода так и не произойдет.

Подробный анализ особенностей нового сетевого стека можно найти в статье: "Windows Vista Network Attack Surface Analysis: A Broad Overview", лежащей на www.symantec.com/avcenter/reference/ATR-VistaAttackSurface.pdf