IPv6 –демократия или диктатура?
tagline1: неизведанное — рядом, но оно запрещено
tagline2: чем чреват принудительно обязательный переход на IPv6

крис касперски, ака мыщъх, a.k.a. nezumi, a.k.a souriz, a.k.a. elraton, no-email

активное вторжение протокола IPv6 в нашу жизнь продолжается: все больше и больше операционных систем, маршрутизаторов, разработчиков серверов и прочего программного обеспечения рапортует о его поддержке, вызывая у пользователей естественное недоумение: что это такое, зачем оно вообще нам нужно и как обстоят дела с (не)безопасностью? попробуем ответить на эти вопросы без лишней воды, затронув не только давно протоптанные темы, но и заглянув в малоизвестные заповедные уголки, куда еще не ступала нога человека (но мыщъх там пасется уже давно)

введение в протокол IPv6

Ужасным недостатком протокола IPv4 (и основным мотивом перехода на IPv6) оказалась ограниченное количество IP-адресов, катастрофическая нехватка которых ощущается уже сейчас. И хотя DHCP-сервера (выдающие динамические IP) и системы трансляции сетевых адресов (Network Address Translation или, сокращенно, NAT) до некоторой степени смягчают остроту проблемы, как ни крути, а 32-битное поле, отведенное разработчиками протокола под IP-адрес, обеспечивает $2^{32} = 4.294.967.296$ уникальных адресов, часть из которых зарезервирована под служебные нужды, так что приведенная цифра слегка завышена.

В эпоху ранней молодости Интернета, когда количество узлов исчислялось десятками, это ограничение не казалось столь существенным недостатком (точнее, никому и в голову не приходило назвать его "ограничением"), но взрывной рост сети сделал свое дело и в ближайшем будущем к Интернету планируется подключить холодильники, микроволновые печи и даже… унитазы, требующие IP-адресов.

Стало ясно, что "дальше так жить нельзя" (с) и протокол IPv4 нужно либо расширять, либо переходить на нечто совершенно иное. Широко распространенное заблуждение гласит, что IPv6 представляет собой слегка доработанную версию IPv4, но это не так. Протокол IPv6 возник не вчера, и даже не позавчера. А очень даже давно. Еще в начале 1990 года в RFC 1750 появилось первое упоминание о грядущей нехватке IP-адресов, которое дало толчок к рассуждениям и поискам новых решений для координации которых в 1993 году комитет IETF сформировал рабочую группу "IPng Area" (где "IPng" расшифровывается как "IP Next Generation" — "IP следующего поколения"), одобренную комитетом "Internet Engineering Task Force" в 25 июля 1994 года. Эту дату можно считать своеобразным историческим моментом в истории создания протокола IPv6. В исследовательских центрах закипела работа, на свет рождалось множество RFC, зачастую противоречащих друг другу.

К концу 2000 года протокол IPv6 достаточно "созрел" и оброс большим количеством спецификаций, "ядро" которых состоит из следующих документов: RFC 2460 — базовое описание протокола; RFC 4291 — система адресации узлов; RFC 2461/RFC 4311 — поиск соседних узлов (Neighbor Discovery), RFC 4443 — ICMP-версия для IPv6 и некоторых других, изучив которые мы с удивлением обнаружим, что IPv6 фактически представляет собой смесь идей, почерпнутых из протоколов: **ISO CLNP** (также известным под кодовым именем **TUBA**), **IPv7** (да-да! это не опечатка! протокол IPv7 — он же **TP/IX** – действительно существует и описан в RFC 1475), протоколе **IP-in-IP** и **RIP**'е, слившиеся с **SIP'ом** в гибрид под именем **SIPP**, описанный в RFC 1710, причем, IPv6 архитектурно стоит гораздо ближе к ISO CLNP, чем, собственно, к своему "прародителю" IPv4.

Первое коммерческое использование протокола IPv6 состоялось 20 июля 2004 года, когда ICANN добавил IPv6 AAAA-записи для Японии (.jp) и Кореи (.kr) сделав их видимыми для всего мира, а чуть позже в этот список попала и Франция (.fr). Естественно, в локальных сетях, протокол IPv6 может использоваться и без "санкции" со стороны ICANN, но… очень трудно представить себе локальную сеть, состоящую более чем из 2^{32} узлов.

Тем не менее, джин уже выпущен из бутылки и IPv6 начинает распространяется по миру, образуя своеобразные "островки", соединенные тоннельными протоколами типа IPv6-over-IPv4, но об этом мы поговорим чуточку позднее, а пока обсудим какие _практические_ преимущества несет в себе переход на IPv6 и чем за них придется расплачиваться.

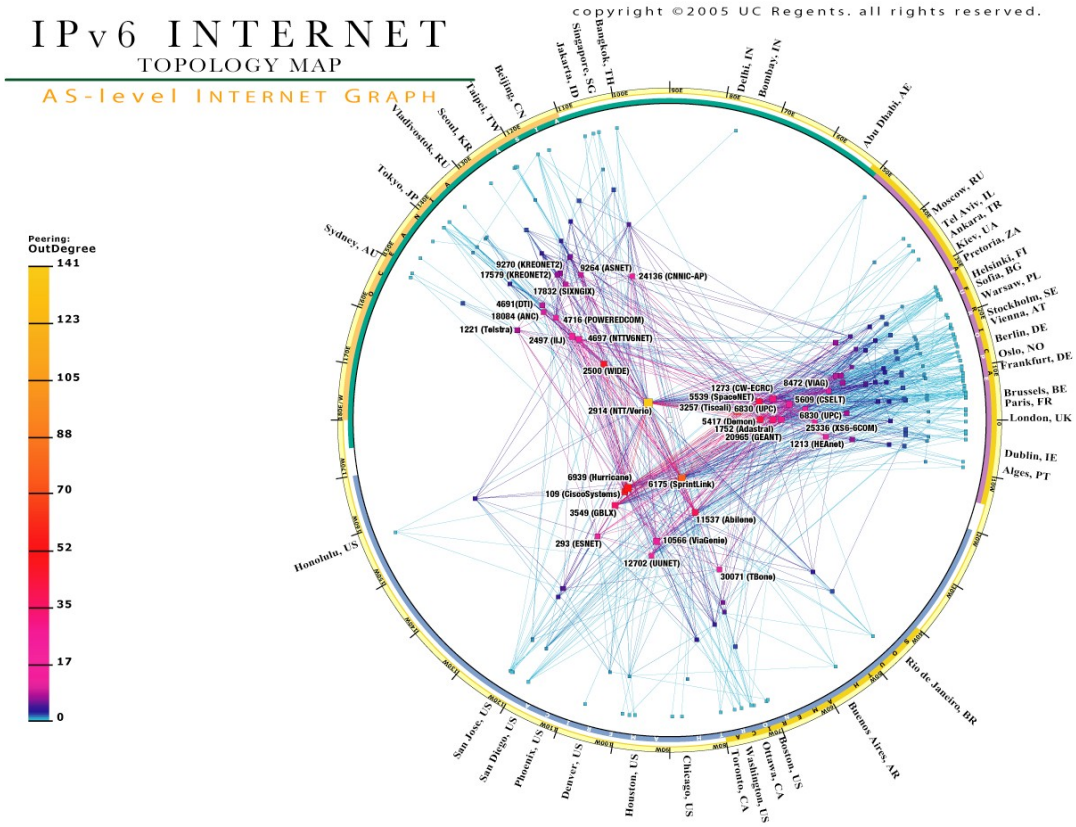

Рисунок 1 топология IPv6 сетей на 2005 год по данным аналитического центра компании Ascore

достоинства и недостатки IPv6

Первое и главное. IPv6 использует 128-битную адресацию, что дает нам порядка $3*10^{38}$ уникальных адресов. Это число настолько велико, что даже если каждый _атом_ Земного Шара подключится к Интернету, в его распоряжении может быть выделено по меньшей мере семь IP-адресов! Другими словами, адресный голод закончится раз и навсегда. Это, несомненно, важное преимущество.

Еще одно преимущество заключается в том, что при переходе с IPv4 на IPv6 протоколы прикладного уровня (TCP/UDP) и использующее их программное обеспечение, ничего не "почувствуют", а, значит, их не придется переписывать. Во всяком случае, так утверждает реклама, но разработчиков на мякине не проведешь!!! Начнем с того, что в IPv6 полностью изменился формат записи IP-адресов и теперь они записываются в виде девяти групп из четырех шестнадцатеричных цифр, разделенных знаком двоеточия, в результате чего типичный IPv6-адрес выглядит как "2001:0DB8:85A3:08D3:1319:8A2E:0370:7334". Попробуй ввести такой с клавиатуры! А что? Может ведь возникнуть потребность связаться с web-сервером, не имеющего доменного имени, тогда мы должны будем ввести в адресной строке браузера следующую абракадабру: "http://[2001:0DB8:85A3:08D3:1319:8A2E:0370:7344]:8080/", заключив IPv6 адрес в квадратные скобки, за которыми (при необходимости) может следовать адрес порта. Вот только… чтобы браузер понял, что в квадратных скобках находится именно IPv6 адрес, а не что-то другое, его код должен быть переписан. Старые версии с новым форматом URL работать не смогут. Увы! Вот такая "замечательная" обратная совместимость.

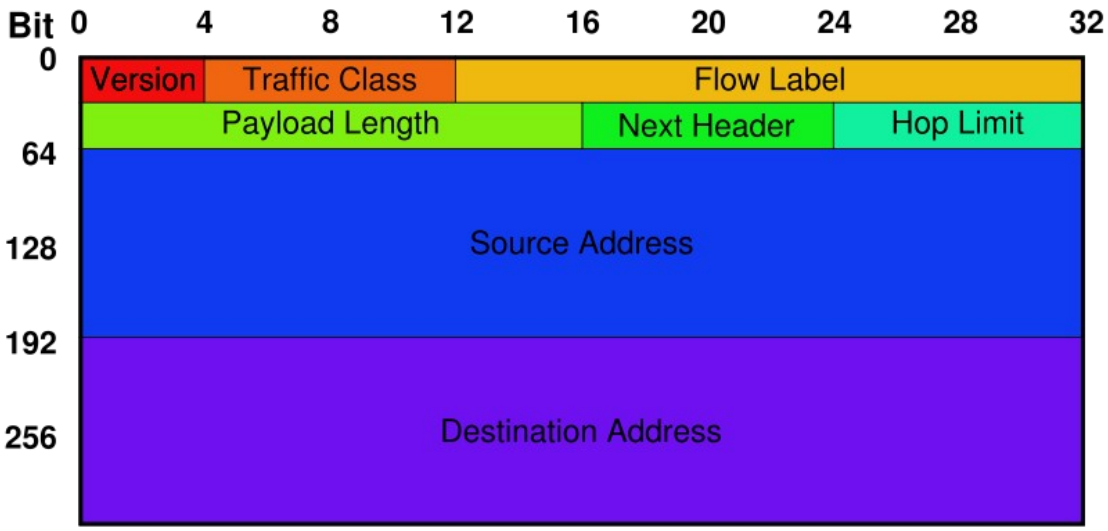

Рисунок 2 формат IPv6 пакета

Фрагментация пакетов из IPv6 была исключена во благо маршрутизаторов и брандмауэров. В IPv4 передаваемый пакет при необходимости разбивается на пакеты меньшего размера, передаваемые вперемешку со всеми остальными, причем порядок поступления пакетов может отличаться от порядка их взаимного расположения, а это значит, чтобы проанализировать пакет на "вшивость" брандмауэр должен собрать весь пакет целиком, складывая поступающие фрагменты в свой локальный стек и задерживая их поступление по каналу. И только по приходу последнего фрагмента (который вполне может оказаться первым _физическим_ фрагментом пакета, несущим в себе заголовок с адресом отправителя и получателя), решить — пропускать его или нет. При этом, накопившиеся в стеке фрагменты выплевываются всем скопом в сети наступает перегруз, а сам брандмауэр требует нехилого количества оперативной памяти. Короче, мрак полный. И большинство IPv4-атак было основано именно на ошибках реализации ассемблера/дизассемблера пакетов (в смысле: сборщика/разборщика).

В IPv6 фрагментации уже нет и максимальный размер пакета составляет 65535 байта, что существенно превышает величину, поддерживаемую большинством маршрутизаторов, поэтому, узел-отправитель должен _самостоятельно_ определить величину MTU (maximum transmission unit – максимальный передаваемый блок), руководствуясь алгоритмом, описанным в RFC 1191, и разбить IPv6 пакет на заданное количество фрагментов, оформив каждый из них как самостоятельный IPv6-пакет. Короче говоря, те же яйца, только зелены. Нагрузка с марштутизаторов теперь перенесена на узлы-отправители, требования к мощности которых существенно возросли (особенно на быстрых каналах).

Другая интересная концепция — опция необязательных подключаемых заголовков (concatenated headers), содержащих различную служебную информацию. В настоящее время поддерживается шесть типов дополнительных заголовков, одним из которых является… заголовок ***фрагментации***, содержащей идентификатор дейтаграммы, номер фрагмента и бит, указывающий: является ли данный фрагмент последним. Выходит, что в IPv6 фрагментация все-таки есть, правда, в отличии от IPv4 фрагментировать пакеты может только узел-отправитель, а не промежуточные марштутизаторы. Кстати говоря, посылка фрагментированного IPv6 пакета позволяла захватить контроль над OpenBSD и это была вторая крупная дыра, обнаруженная в ней 13 марта 2007 года более чем за десять лет промышленной эксплуатации, что доказывает сырость реализации IPv6.

Еще, в отличии от IPv4, протокол IPv6 способен поддерживать пакеты сверхбольшого размера (jumbograms) вплоть до 4 Гбайт, что очень полезно для суперкомпьютеров, обрабатывающих данные и не желающих отвлекаться каждый раз когда приходит очередная порция 64 Кбайтных данных. На первый взгляд, jumbogram'ы создают серьезную проблему, ведь если какой-то зловредный пользователь отправит 4 Гбайтный IPv6 пакет в медленную сеть, то остальные пакеты надолго застрянут в очереди. На самом деле этого не произойдет, поскольку максимально допустимый размер пакета определяется промежуточными

марштутизаторами и очень часто он составляет 576 байт, так что ни о каких "запорах" сети не может быть и речи.

И последнее. В IPv6 (как и в IPv4) имеется специальное поле, определяющее срочность доставки пакета, что очень полезно при работе с потовым аудио/видео, однако, большинство реализаций IPv4 обрабатывают все пакеты на равных основаниях, что затрудняет работу программ, нуждающихся в передаче данных в реальном времени. Посмотрим, что измениться в реализациях IPv6…

Список остальных достоинств IPv6 перечислен в одноименной врезке, к которой мы отсылаем всех желающих, а сами тем временем попробуем разобраться в механизме взаимодействия IPv4 и IPv6 узлов.

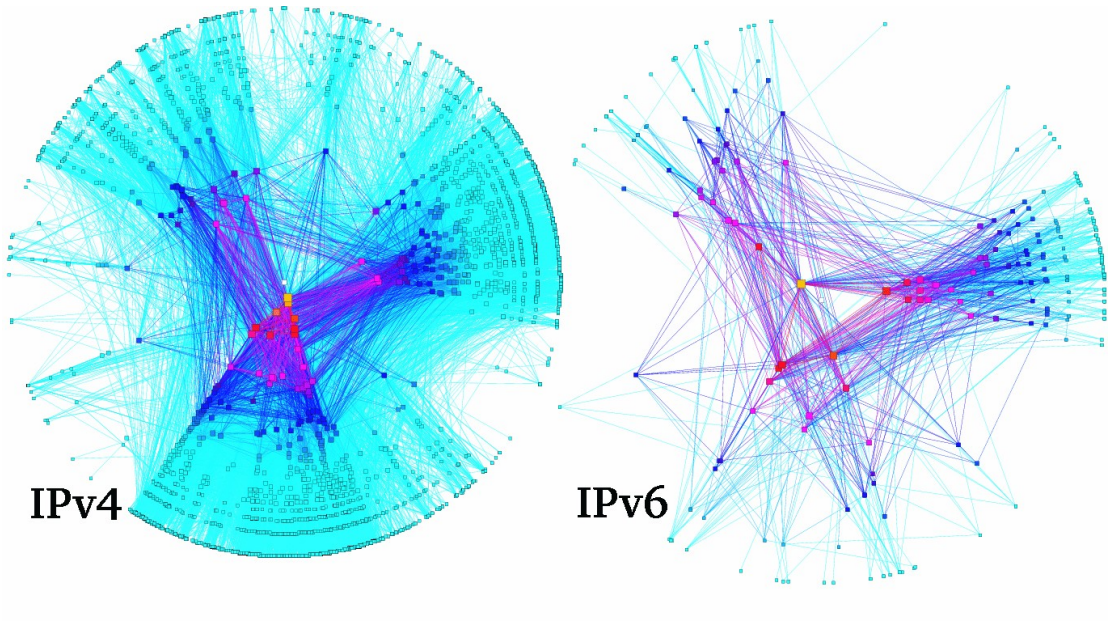

Рисунок 3 сравнение топологий IPv4 и IPv6 узлов по данным аналитического центра компании Ascore (2006 год)

тоннели и проблемы безопасности

Для обеспечения обратной совместимости большинство узлов, поддерживающих IPv6, так же поддерживают и IPv4. Данная технология получила название "двойного стека" (dual stack) и описана RFC 4213. Ее достоинство в том, что сервер, оснащенный двойным стеком, может обслуживать как IPv6, так и IPv4 клиентов, правда, с одной небольшой оговоркой. Старшие разряды IPv6-адреса заполняются нулями и мы получаем обыкновенный 32-битный IPv4 адрес (про дефицит которых уже говорилось выше), только записанный в IPv6 нотации. К "полноценному" IPv6 адресу IPv4 клиент подключиться не сможет (надеюсь, не нужно объяснять почему?!).

С маршртутизаторами — та же самая история. Поскольку, IPv4 в ближайшие несколько лет умирать не собирается, IPv6 маршрутизатор вынужден поддерживать технологию двойного стека, что: а) предъявляет повышенные требования к ресурсом; б) увеличивает сложность реализации, а, значит, и вероятность возникновения ошибок. Теоретически, можно выключить стек IPv4, создав "чистый" IPv6 узел для общения с себе подобными, однако, в существующих операционных системах это сделать очень непросто, а потому "чистых" IPv6 узлов в природе не наблюдается, несмотря на то, что количество "гибридных" стеков неуклонно растет, поскольку последних версиях BSD, Linux, Mac OS X и Windows стек IPv6 автоматически включается инсталлятором по умолчанию.

Вот только… если между двумя IPv6 узлами окажется расположен хотя бы один IPv4 маршрутизатор, то "совокупляться" им придется либо по технологии двойного стека (но как быть если IPv6 узел не имеет IPv4 адреса?!) либо же использовать тоннели IPv6 over IPv4. Таких тоннелей много, но принцип действия у них один: IPv6 пакет укладывается в IPv4, передаваемый обычным путем, а получатель проделывает обратную операцию. Стоп! Как мы сможем отправить IPv4 пакет узлу, имеющему только IPv6 адрес? Единственный выход

заключается в установке специального сервера, устроенного наподобие Proxy. Отправитель берет IPv6 пакет, кладет его внутрь IPv4 пакета, отправляет его одному из глобальных Proxy-серверов с двойным IPv6/IPv4 стеком. Сервер извлекает IPv6 пакет, смотрим на адрес получателя и передает его следующему Proxy-серверу, соединенным с получателем только IPv6/IPv4 марштутизаторами. Естественно, таких Proxy серверов должно быть много (в противном случае они просто лягут под нагрузкой) и далеко не во всех случаях удается отыскать подходящий маршрут.

В Windows Vista встроен тоннельный протокол Teredo (включенный по умолчанию), разработанный сотрудником Microsoft по имени Christian Huitema с претензией на всеобщий стандарт и потому детально описанный в RFC 4380 (http://www.rfc-editor.org/rfc/rfc4380.txt). Подобная открытость совсем не характерна для Microsoft, зато сам протокол выполнен в свойственном для нее стиле: главное маркетинг, а технические аспекты традиционно откладываются на потом.

Специалисты из Symantec'a проанализировали Teredo, подвергнув его резкой критике: www.symantec.com/enterprise/security_response/weblog/2006/11/the_teredo_protocol_tunneling.html http://www.symantec.com/avcenter/reference/ATR-VistaAttackSurface.pdf (Windows Vista Network Attack Surface Analysis, http://www.symantec.com/avcenter/reference/Teredo_Security.pdf (The Teredo Protocol: Tunneling Past Network Security and Other Security Implications). И критиковать было за что!!!

Teredo инкапсулируя IPv6 пакеты внутрь UDP-дейтаграмм, переедаемых через IPv4 (причем, уровень вложенности формально неограничен и при желании можно создавать "матрешку" или слоеный торт в стиле "наполеон"), без труда обходит существующие защитные механизмы типа брандмауэров, NAT'ов, систем обнаружения вторжений, и т. д., и т. п., поскольку все, что они видят — это "честный" UDP пакет, посылаемый на один из внешних узлов. Истинный целевой адрес, порт и содержимое пакета, скрыто внутри IPv6, но чтобы его "достать", брандмауэр/IDS должен знать о существовании Teredo! С учетом же новизны Windows Vista вероятность такой осведомленности крайне невелика и на время смены оборудования для хакеров, червей и прочих вирусов наступает настоящая благодать!!! И если программные защитные комплексы еще хоть как-то можно обновить, то "железо" придется отправлять на свалку и приобретать новое (дождавшись его появления на рынке).

Впрочем, тоже самое относится и к остальным тоннельным протоколам, реализованных вне Microsoft и поддерживаемых операционными системами Linux и xBSD, например, 6to4, описанном в RFC-3056 (http://tools.ietf.org/html/rfc3056), правда, учитывая, что 6to4 появился в 2001 году к настоящему моменту многие производители уже успели его поддержать.

Второе обвинение, предъявляемое протоколу Teredo намного серьезнее первого: Teredo "глобализует" локальные адреса, делая их доступными отовсюду, что расходится с политикой безопасности многих компаний (достаточно вспомнить нашумевшее дело, как один парень определил IP-адреса сети Пентагона). О лучшем сканере узлов, закрытых NAT'ом или брандмауэром, хакеры не могли и мечтать!

Выход Висты — огромный подарок для хакеров, активность которых значительно возрастет, а защитные средства за один-два дня не появятся!

Рисунок 4 основные центры сосредоточения IPv6 узлов в Северной Америке по данным The NLANR Active Measurement Project

сырость IPv6 и атаки на него

Протокол IPv6 очень молод. Настолько молод, что окончательно не стандартизован. Реализации стека IPv6 едва вылезли из пеленок и до сих пор остаются не протестированными в "промышленных" условиях. Значительная часть ошибок, обнаруживаемых в Linux и BSD относится именно к IPv6. Старик Google по запросу "IPv6 Vulnerability" выдает около миллиона (!) ссылок и хотя большинство дырок уже давно исправлено, окончательно вылизанный IPv6 стек мы получим лет… через несколько. В BSD/Linux системах. Что же касается Microsoft, то здесь очень трудно найти повод для оптимизма.

Самое неприятное, что от атак на IPv6 нельзя защититься брандмауэром/марштутизатором, поскольку брандмауэр/маршрутизатор, поддерживающий IPv6 сам по себе представляет нехилый объект для удаленных атак.

Кстати, о маршрутизаторах…

внутри маршрутизатора

Большинство аппаратных марштутизаторов и брандмауэров (в том числе и CICSO) строятся на базе BSD-подобных операционных систем, с минимальной адаптацией под конкретную архитектуру или даже без таковой. Чем же они тогда отличаются от обычного ПК, с установленной Free/Net/OpenBSD?! Ответ — на ПК BSD еще поставить надо, и железо правильное подобрать, и еще программное обеспечение для управления всей этой системой написать. К тому же ПК содержит много лишнего. Зачем марштутизатору интегрированная видео-карта, AC3-звук и прочие компоненты? К тому же, x86 процессоры далеко не самое лучше средство для обработки сетевых пакетов. Но это, так сказать, сугубо техническая сторона вопроса. С точки же зрения безопасности, код, основанный на BSD, наследует все ошибки IPv6 стека, причем, исправить их значительно труднее, чем в ПК-версии. Как минимум потребуется дождаться заплатки от производителя маршрутизатора/брандмауэра, а потом "залить" ее внутрь "железки", если, конечно, она вообще предусматривает возможность обновления.

Вот такая, значит, напряженная ситуация.

основные методы защиты от атак

Какую выгоду можно извлечь из IPv6 протокола? Первое, что приходит в голову: покупаем один IPv4-адрес и разворачиваем за ним огромную сеть со множеством IPv6 адресов, доступных, благодаря туннельным протоколам, из любой точки планеты. Даже из Антарктиды. Правда, с одной небольшой оговоркой — IPv6 адреса смогут увидеть лишь IPv6 клиенты, с поддерживающие тот же самый тоннельный протокол, что установлен на нашем IPv4/IPv6 узле. В настоящий момент таких узлов практически нет и очень немногие пользователи согласятся совершать дополнительные телодвижения только затем, чтобы зайти на наш IPv6 сервер. Уж лучше "посадить" все серверы на один IPv4, используя виртуальные хосты, NAT'ы или другие технологии подобного типа.

В настоящий момент с IPv6 можно только "проиграться", но не более того. Практической отдачи с него никакой, а вот проблем с безопасностью он создает немало. Поэтому, лучше отключить поддержку IPv6, перекомпилировав ядра Linux/BSD. А что же на счет Висты? Увы, _полностью_ отключить поддержку IPv6 нельзя и даже при выключенном IPv6, компоненты IPv6 стека будут по прежнему принимать пакеты, что позволит атаковать систему при наличии в ней дыр (а в том, что такие дыры есть, можно не сомневаться). Хорошо хоть Teredo отключается штатными средствами! А IPv6 можно заблокировать на IPv4 брандмауэре, построенном на базе BSD с "вырезанным" IPv6 стеком.

Конечно, отказ от использования IPv6 – слишком радикальная мера защиты, но на данный момент она, увы, единственная. Все остальные решения — ненадежны.

>>> врезка ключевые особенности протокола IPv6

- расширенные возможности адресации;
- упрощенный формат заголовка;
- изъятие из заголовка поля контрольной суммы;
- поддержка дополнительных заголовков;

- усиленные механизмы аутентификации;
- уменьшение размеров таблиц маршрутизации;
- возможность смены положения узла с сохранением его адреса;

>>> врезка реализация стека IPv6 в разных осях

ось	год реализации IPv6	по умолчанию	примечания
Windows NT 4 SP1	1998	выключен	экспериментальный IPv6 стек
Windows 2000 SP1	2002	выключен	экспериментальный IPv6 стек
Windows XP	2001	выключен	бета-версия IPv6 стека для разработчиков
Windows XP SP1	2002	выключен	версия для локальных корпоративных сетей
Windows Server 2003	2002	выключен	версия для локальных корпоративных сетей
Windows Vista	?	включен	стабильная версия
xBSD	2000	включен	стабильная версия (проект KAME)
Linux 2.1.8	1996	выключен	бета-версия
Linux 2.6.10	?	включен	стабильная версия (Linux IPv6 Development Project)
IBM's AIX 4.3	1997	выключен	стабильная версия
Apple Mac OS X 10.3	2003	включен	стабильная версия

Таблица 1 реализация стека IPv6 в различных операционных системах и его состояние в конфигурации по умолчанию

>>> врезка куда девался IPv5

Хорошо, сначала у нас был IPv4, теперь все готовятся к неизбежному переходу на IPv6, а куда же подевался IPv5?! Действительно, несправедливо делать вид, будто бы такого протокола и вовсе не существует, когда он есть, — известный под именем Streams 2 (ST2) и описанный в RFC 1819, он работает на том же уровне, что и IPv4, используясь (пусть и не очень широко) в приложениях реального времени.