Кафедра радиоэлектроники и защиты информации
(РЗИ)

Е.Ю. Агеев

Защита информационных процессов в компьютерных системах

Учебно-методическое пособие по проведению практических занятий
студентов специальности 090104
«Комплексная защита объектов информатизации»

Томск, 2012

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Томский государственный университет систем управления и радиоэлектроники» (ТУСУР)

Утверждаю:

Зав. каф. РЗИ

_____ Задорин А.С.

Кафедра радиоэлектроники и защиты информации (РЗИ)

Защита информационных процессов в компьютерных системах

Учебно-методическое пособие по проведению практических занятий студентов специальности 090104
«Комплексная защита объектов информатизации»

Разработчик:

доц. каф. РЗИ

_____ Агеев Е.Ю.

Томск, 2012

УДК
681.3.067

Рецензент:
профессор каф. РЗИ Задорин А.С.

Е.Ю. Агеев
Защита информационных процессов в компьютерных системах. Учебно-методическое пособие по проведению практических занятий. / Е.Ю. Агеев. – Томск: ТУСУР, 2012. –35 с

В методическом пособии представлены темы практических занятий по дисциплине «Защита информационных процессов в компьютерных системах» для студентов специальности 090104 «Комплексная защита объектов информатизации». В ходе выполнения практических работ студенты знакомятся с такими инструментами получения информации о сетевой инфраструктуре и возможных уязвимостях компьютерных систем, как сканер сетевых портов и сканер уязвимостей конечных устройств, изучают элементы защиты электронных документов и сообщений электронной почты.

Методические указания предназначены для студентов очной, заочной и дистанционной форм обучения специальности 090104 по дисциплине «Защита информационных процессов в компьютерных системах».

.

УДК
681.3.067

Содержание

Занятие №1 Работа со сканером портов NMAP

Продолжительность- 2 часа
Максимальный рейтинг- 6 баллов

Цель работы

Изучить работу с программой-сканером портов NMAP [1].

Общая информация

Порт – это идентификатор приложения, работающего на компьютере и использующего сетевые соединения. Как для TCP, так и для UDP-сервисов номера портов лежат в диапазоне от 1 до 65535. Номера портов от 0 до 1023 считаются зарезервированными для общеупотребительных приложений (табл. 11), обычно выполняющихся от имени пользователя root или другого привилегированного пользователя. Соответствующие им номера портов называются общеизвестными. Номера портов с 1024 по 65535 могут использоваться остальными приложениями. Они обычно соответствуют определенным сервисам, но не обязательно. Кроме того, существуют недолговечные номера портов, которые операционная система выбирает случайным образом из номеров, превышающих 1024, (обычно - в верхней части диапазона). Они используются для машин, которые произвольным образом устанавливают соединения с другими машинами. Например, для загрузки web-страницы ваша машина обратится к порту 80 web-сервера. Сервер увидит входящее соединение с некоторым случайным номером порта, превышающим 1024. В таком случае сервер будет знать, что это, вероятно, пользователь, а не другое приложение, устанавливающее с ним соединение. Он также использует недолговечный номер порта для отслеживания определенного пользователя и сеанса. Например, если вы параллельно откроете два навигатора, то ваш компьютер для сеанса каждого из них создаст два разных номера порта для установления соединений, которые сервер будет считать различными.

Таблица 11.

Номер порта	Протокол	Сервис
21	FTP	Протокол передачи файлов (управляющий порт)
22	SSH	Защищенный shell
23	Telnet	Telnet
25	SMTP	Почтовый сервис
53	DNS	Разрешение доменных имен
79	Finger	Finger
80	HTTP	Web-сервис
135-139	NetBIOS	Сетевые коммуникации Windows
443	SSL	Защищенный web-сервис

То, что пакет помечен для порта 80, не запрещает ему содержать данные, отличные от web-трафика. Система номеров портов зависит от определенной "честности" машин, с которыми приходится взаимодействовать, и именно отсюда может прийти беда. На самом деле, многие приложения, такие как программы мгновенного обмена сообщениями и одноранговое ПО, которые обычно блокируются межсетевым экраном организации, нарушают эту конвенцию и проскальзывают через порт 80, который согласно конфигурации остается открытым, поскольку пользователям, находящимся позади межсетевого экрана, разрешен web-доступ.

Когда порт на компьютере открыт, он получает весь направляемый в него трафик, законный или незаконный. Посылая некорректно сформированные пакеты, пакеты со слишком большим количеством данных или с некорректно отформатированными данными, иногда можно вызвать аварийное завершение основного приложения, перенаправить поток управления в этом приложении и незаконно получить доступ к машине. Такая атака называется переполнением буфера и составляет большой процент современных уязвимостей.

Переполнение буфера происходит, если прикладные программисты неаккуратно пишут программы и не обеспечивают должную обработку данных, «переполняющих» области памяти, отведенные входным переменным. Когда в программу поступают входные данные, не умещающиеся в отведенный буфер, они могут изменить внутренний ход выполнения программы и в результате предоставить хакеру доступ к ресурсам системного уровня. Раньше это было технически сложной задачей, за которую могли взяться только самые квалифицированные хакеры. Но теперь, чтобы осуществить подобный взлом, уже не нужно быть высококлассным программистом. Доступны программы, которые с одного щелчка мыши автоматически выполняют переполнение буферов. Почти все программы, независимо от размера, содержат ошибки такого рода. Современное программное обеспечение, насчитывающее миллионы строк исходных текстов, - просто-напросто слишком сложное, чтобы избежать подобных ошибок. Возможно, со временем, когда вырастут новые поколения программистов, обученных автоматически писать безопасный код, данная проблема потеряет свою остроту или исчезнет совсем. Пока же необходимо внимательно следить за тем, какие приложения или порты видны в вашей сети. Эти порты являются потенциальными «окнами» в серверах и рабочих станциях, через которые хакеры могут запускать свой вредоносный код в ваш компьютер. Поскольку именно здесь происходит большинство нарушений безопасности, очень важно понимать, что происходит на этом уровне на ваших серверах и других машинах. Этого можно добиться с помощью программного средства, называемого сканером портов.

Сканеры портов последовательно опрашивают порты TCP или UDP и смотрят, не ответит ли приложение. Если ответ получен, это означает, что некоторое приложение слушает порт с данным номером. Имеется 65535 возможных портов TCP и столько же - UDP. Сканеры можно сконфигурировать для опроса всех возможных портов или только общеупотребительных (с номерами, меньшими 1024). Веская причина для полного сканирования состоит в том, что сетевые троянские и другие вредоносные программы, чтобы избежать обнаружения, нередко используют нетрадиционные порты с номерами в верхней части диапазона. Кроме того, некоторые производители не следуют стандартам должным образом и подключают серверные приложения к портам с большими номерами. Полное сканирование охватывает все возможные места, где могут скрываться приложения, хотя и требует больше времени.

Существует большое число программ-сканеров портов от очень сложных с множеством различных возможностей до имеющих минимальную функциональность. В конце концов, можно вручную выполнить сканирование портов, например, с помощью программы Telnet, проверяя порты по очереди. Просто подключайтесь к IP-адресу, добавляя номер порта, например:

```
telnet 192.168.0.1:80
```

Номер после двоеточия (для некоторых реализаций Telnet необходимо просто оставить пробел между IP-адресом и номером порта) говорит Telnet, что для соединения надо использовать порт 80 вместо стандартного для Telnet порта 23. Вместо того чтобы получить от Telnet обычное приглашение вы соединитесь с web-сервером, если он запущен на машине. После нажатия клавиши ввода вы получите первый ответ web-сервера навигатору. Вы увидите информацию из заголовка HTTP, которая обычно обрабатывается навигатором и скрыта от пользователя. Так же можно поступить с любым открытым портом, по сути, именно это и делают сканеры портов: они пытаются установить соединение и ожидают ответ. Некоторые сканеры портов пытаются также идентифицировать операционную систему на другом конце, выявляя так называемые идентификационные метки TCP. Хотя TCP/IP является стандартом сетевых коммуникаций, каждый производитель реализует его немного иначе, чем другие. Эти различия, обычно не мешающие взаимодействию, проявляются в ответах на любое воздействие, такое как эхо-тест или попытка установления TCP-соединения. Сравнивая полученный от машины ответ с базой известных

идентификационных меток TCP, можно сделать разумное предположение об операционной системе на другом конце.

Знание операционной системы и ее версии может послужить хорошей отправной точкой для определения того, какие зацепки и средства проникновения стоит испробовать. Это очень веская причина для регулярного сканирования своей сети, чтобы определить, какие порты в системе оставлены открытыми. Затем следует их просмотреть, закрыть неиспользуемые порты и защитить те, которые должны оставаться открытыми.

Планирование сканирования портов

При планировании сканирования портов любой сети помните, что эта деятельность создает большую нагрузку на сеть. Сканирование за короткое время десятков тысяч портов порождает в сети интенсивный трафик. Если вы используете для сканирования устаревшей сети на 10 Мбит/с мощный компьютер, это может существенно повлиять на сетевую производительность. При сканировании через Интернет данная проблема будет менее острой, так как ограничивающим фактором послужит пропускная способность промежуточных соединений, однако все равно можно снизить производительность загруженного web-сервера или почтового сервера. В крайних случаях ваша активность может даже привести к прекращению работы машин.

Независимо от способа использования, перед сканированием следует получить разрешение владельца сканируемых хостов. Сканирование портов - деятельность на грани законности (в действительности вы не взламываете системы, просто опрашиваете сеть). Однако вашему начальнику может быть не до нюансов, если вы нарушите работу корпоративной сети. Даже при наличии разрешения необходимо принять во внимание предполагаемый эффект сканирования целевой сети. Если это интенсивно используемая сеть, сканирование нужно выполнять ночью или в периоды наименьшей активности. Некоторые сканеры имеют возможность замедлять посылку пакетов, чтобы не очень сильно воздействовать на сеть. Это означает, что сканирование будет выполняться дольше, но в более дружественном для сети режиме.

Некоторые современные устройства, такие как межсетевые экраны и некоторые маршрутизаторы, достаточно интеллектуальны, чтобы распознать сканирование своих портов и отреагировать на него. Брандмауэр UNIX и Linux – Iptables можно сконфигурировать для этого, используя опцию multiport и устанавливая флаг приоритета. Машины могут отвечать на сканирование портов снижением скорости ответа для каждого последующего опроса. В итоге ваше сканирование может растянуться до бесконечности.

Когда вы получите разрешение на сканирование, следует определить, с какой целью вы собираетесь сканировать сеть. Сканеры портов предлагают быстрый способ просмотра диапазона адресов и выявления все активных машин в этом сегменте, позволяя выполнять инвентаризацию сети. Сканер портов покажет все сервисы, запущенные в данный момент на машине. Если это серверная машина, то, вероятно, таковых окажется много, и, возможно, не все из них на самом деле нужны для выполнения основной функции машины. Чем больше работающих сервисов, тем меньше безопасности. И все эти программы могут замедлять работу перегруженного сервера. Ненужные Web-, FTP- и DNS-серверы крадут циклы процессора у основной функции компьютера. Сканирование портов серверов с последующим анализом результатов и оптимизацией может дать немедленное увеличение скорости и сокращение времени реакции. Сканирование позволяет обнаружить вирусные программы, внедренные на компьютеры сети. Эти программы называются шпионским ПО, потому что нередко они пытаются следить за активностью пользователя и могут передавать собранные данные обратно на шпионский центральный сервер. Эти программы обычно не опасны, но их чрезмерное количество может существенно снизить производительность труда пользователя. Кроме того, написаны они зачастую неаккуратно и могут мешать работе других программ или даже вызывать их аварийное завершение. Другим классом вредоносного программного обеспечения являются «троянские» программы. Эти программы

специально созданы для взлома сетей. Подобно троянскому коню из греческой мифологии, эти программы открывают хакерам и взломщикам заднюю дверь в вашу сеть. Обычно их присутствие можно обнаружить только по открытому сетевому порту, а с помощью антивирусных средств выявить их крайне сложно. Оказавшись внутри компьютера, большинство «троянских» программ пытаются вступить во внешние коммуникации, чтобы дать своему создателю или отправителю знать, что они заразили машину на этих портах. В табл. 12 перечислены наиболее распространенные «троянские» программы и номера портов, используемые ими. Сетевые «черви» - особо мерзкий тип вирусов. Зачастую они снабжены сетевыми средствами и открывают порты на зараженном компьютере. Сетевые «черви» используют сеть для распространения и поэтому иногда выявляются при сканировании портов.

Таблица 12.

Номер порта	**IP протокол**	**Известные "троянские" программы, использующие эти порты**
12456 и 54321	TCP	NetBus
23274 и 27573	TCP	Sub7
31335	TCP	Trin00
31337	TCP	Back Orifice
31785-31791	TCP	Hack 'a'Tack
33270	TCP	Trinity
54321	UDP	Back Orifice 2000
60000	TCP	Deep Throat
65000	TCP	Stacheldraht

Программа NMAP

Программа Nmap - вне всяких сомнений, лучший сканер портов. Его создатель - программист с псевдонимом «Fyodor», разработки которого используются во многих других программах и портированы практически на все употребительные операционные системы. Достаточно сказать, что Nmap должен входить в инструментарий каждого администратора безопасности. Перечислим некоторые из основных достоинств Nmap:

- У него есть множество опций. Можно понизить частоту отправки зондирующих пакетов, если вы опасаетесь замедления работы сети, или, наоборот, повысить ее, если имеется запас ширины полосы пропускания. Опции невидимости - еще один элемент репертуара Nmap. Хотя некоторые критикуют эти опции, полагая, что они необходимы только хакерам, для них имеются законные применения. Например, если необходимо проверить, насколько чувствительной является система обнаружения вторжений. Nmap позволяет сделать это, выполняя сканирование с различными уровнями невидимости. Далее, Nmap выходит за рамки простого сканирования портов и осуществляет идентификацию ОС, что полезно при установлении соответствия между IP-адресами и машинами.
- Он легкий, но мощный. Код Nmap невелик и будет выполняться даже на самых старых машинах, он запускается даже на некоторых КПК. В небольшом объеме он концентрирует огромную энергию и без проблем сканирует очень большие сети.
- Он прост в использовании. Хотя существует множество различных способов его запуска, реализуемое по умолчанию базовое сканирование SYN делает все, что требуется большинству приложений. Имеется как режим командной строки, так и графический интерфейс для LINUX и Windows, чтобы удовлетворить запросы как круглых дураков, так и тех, кому необходима графика. Он также очень хорошо документирован и поддерживается большим числом разработчиков и оперативных ресурсов.

Экран графической оболочки программы показан на рис. 1.

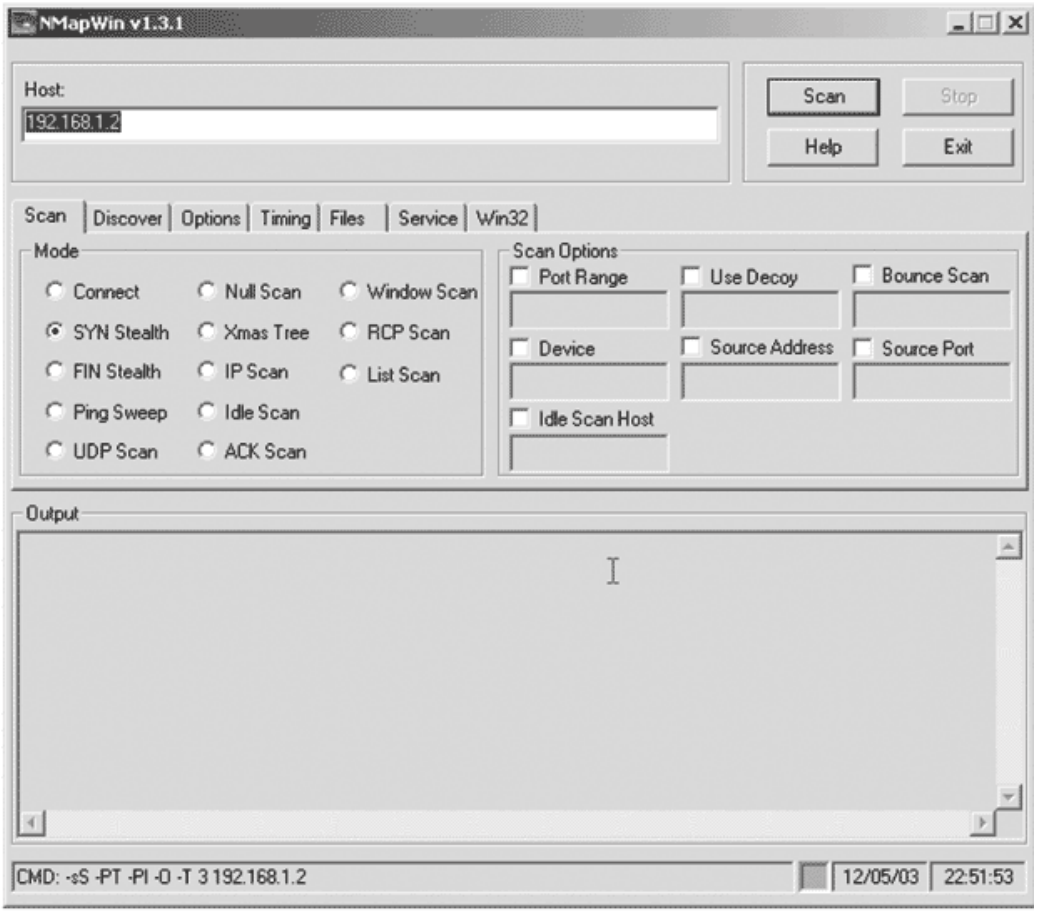

Рис. 1. Графический интерфейс Nmap.

Графический клиент Nmap предоставляет весьма простой интерфейс. Вверху имеется поле для ввода IP-адреса или диапазона IP-адресов, а чтобы начать сканирование, достаточно нажать кнопку Scan. В табл. 13 приведены различные форматы для ввода IP-адресов. Адреса могут также извлекаться из файла, если выбрать пункт Input элемента File основного меню и задать текстовый файл с данными в подходящем для Nmap формате.

Таблица 13.

Формат	**Пример**
Одиночный IP-адрес	192.168.0.1
IP- адреса, разделенные запятыми	192.168.0.1,192.168.0.2
IP-диапазон, разделенный дефисом	192.168.0.1-255
Использование стандартной нотации с косой чертой	192.168.0.1/24 (сеть класса С из 256 адресов)

Nmap можно запустить из командной строки как в LINUX, так и в Windows. Общий формат команды запуска таков:

nmap параметры IP-диапазон

Основные способы сканирования NMAP

Nmap поддерживает множество различных видов сканирования. В табл. 14 перечислены наиболее употребительные. В скобках указаны параметры командной строки.

Таблица 14.

Тип сканирования (параметры командной строки)	**Описание**
SYN (-sS)	Способ сканирования, установленный по умолчанию, пригодный для большинства целей. Он менее заметен, чем TCP Connect, то есть не будет фиксироваться большинством простых средств протоколирования. В этом режиме в каждый возможный порт посылаются одиночные TCP-пакеты с установленным флагом SYN. Если в ответ возвращается пакет SYN ACK, то Nmap делает вывод, что здесь запущен сервис. Если ответа нет, то предполагается, что порт закрыт SYN-сканирование не завершает трехходовое квитирование установления связи в TCP, так как не возвращает целевой машине пакет с установленным флагом ACK. С точки зрения сканируемой системы действующие соединения не устанавливаются. Однако, удаленная система будет удерживать эту «половинку сокета» открытой, пока не пройдет максимально допустимое время ответа. Некоторые современные серверы и программы выявления вторжений достаточно интеллектуальны, чтобы уловить подобные действия, но для большинства машин SYN-сканирование будет невидимым
TCP-соединение: Connect (-sT)	Это тип сканирования напоминает SYN за исключением того, что трехходовое квитирование установления связи в TCP выполняется до конца и устанавливается полноценное соединение. Подобное сканирование не только шумно, но и создает дополнительную нагрузку на сканируемые машины и сеть. Однако, если скрытность или экономия полосы пропускания не являются приоритетными, то сканированием Connect, по сравнению с SYN, можно порой получить более точные результаты. Кроме того, если у вас нет привилегий администратора или суперпользователя на машине Nmap, вы не сможете воспользоваться никаким другим типом сканирования, поскольку создание построенных особым образом пакетов для других типов сканирования требует низкоуровневого доступа к ОС
Эхо-тестирование: Ping Sweep (-sP)	Выполняется простое эхо-тестирование всех адресов, чтобы увидеть, какие из них ответят на ICMP-запрос. Если вас на самом деле не интересует, какие сервисы запущены, и вы просто хотите знать, какие IP-адреса активны, то данный тип позволит достичь цели много быстрее, чем полное сканирование портов. Однако некоторые машины могут быть сконфигурированы так, чтобы не отвечать на ping (например, новый межсетевой экран XP), но, тем не менее, выполнять некоторые сервисы, поэтому Ping Sweep - менее надежный метод, чем полное сканирование портов
UDP-сканирование: UDP Scan (-sU)	Этот тип сканирования проверяет наличие слушаемых UDP-портов. Так как UDP, в отличие от TCP, не отвечает положительным подтверждением, а отвечает на входящий пакет, только когда порт закрыт, данный тип сканирования

	может иногда приводить к ложным срабатываниям, однако он способен выявить троянские программы, использующие UDP-порты с большими номерами и скрытые RPC-сервисы. Он может быть весьма медленным, так как некоторые машины намеренно замедляют ответы на этот тип трафика, чтобы избежать перегрузки. Однако машины, выполняющие ОС Windows, не реализуют замедления, поэтому вы сможете использовать UDP для нормального сканирования хостов Windows.
FIN-сканирование: FIN Stealth (-sF)	Это скрытное сканирование, аналогичное SYN, но использующее пакеты TCP FIN. Большинство компьютеров, но не все, ответят пакетом RST, поэтому сканирование FIN сопряжено с ложными срабатываниями и пропуском положительных результатов, но может осуществляться под наблюдением некоторых программ выявления вторжений и при наличии других контрмер
NULL-сканирование: NULL Scan (-sN)	Еще одно весьма скрытное сканирование, при котором все флаги заголовка TCP сброшены (или пусты). Подобные пакеты обычно некорректны, и некоторые хосты не знают, что с ними делать. Операционные системы Windows входят в эту группу, так что их сканирование в режиме Null будет давать недостоверные результаты. Однако для серверов не под Windows, защищенных межсетевым экраном, оно может стать способом проникновения
XMAS-сканирование: XMAS Tree (-sX)	Аналогично сканированию NULL, за исключением того, что все флаги в заголовке TCP установлены, а не сброшены (отсюда и название - пакет расцвечен, как рождественская елка). Машины Windows, ввиду особенностей реализации на них стека TCP, не отвечают на подобные пакеты
Сканирование через отражатель: Bounce Scan (-n FTP_HOST)	Этот хитроумный тип сканирования использует лазейку в протоколе TCP для "отражения" сканирующих пакетов от сервера FTP во внутреннюю сеть, которая обычно недоступна. Зная IP-адрес сервера FTP, который подключен к локальной сети, вы можете проникнуть через межсетевой экран и сканировать внутренние машины. Стоит проверить и свою собственную сеть на наличие данной уязвимости. В большинстве современных серверов FTP эта дыра в защите ликвидирована. Примечание: В дополнение к сканируемым IP-адресам вы должны задать действующий сервер FTP, имеющий доступ к сети
RPC-сканирование: RPC Scan (-sR)	Этот особый тип сканирования ищет машины, отвечающие сервисам удаленного вызова процедур (RPC). Сервис RPC, при определенных условиях позволяющий удаленным командам выполняться на машине, сопряжен со значительным риском. Так как сервисы RPC могут выполняться на многих различных портах, то по результатам обычного сканирования выявить эти порты трудно. RPC-сканирование зондирует найденные открытые порты с помощью команд, показывающих имя программы и версию сервиса RPC. Неплохо время от времени проводить подобное сканирование, чтобы узнать, работают ли, и где именно, RPC-сервисы

Window-сканирование: Window Scan (-sW)	Данный тип сканирования полагается на аномалию в ответах на пакеты ACK в некоторых операционных системах, чтобы обнаружить порты, которые предположительно фильтруются. Известно, что к числу операционных систем, уязвимых для подобного сканирования, принадлежат некоторые версии AIX, Amiga, BeOS, BSDI, Cray, DG/UX, Digital LINUX, FreeBSD, HP/UX, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, OpenVMS, OS/2, QNX, Rhapsody, SunOS 4.X, Tru64 LINUX, Ultrix, VAX и VxWorks
Реактивное сканирование: Idle Scan (-sI хост-зомби:используемый_порт)	Данный тип сканирования появился в Nmap версии 3.0. Это сверхскрытный метод, при применении которого пакеты сканирования отражаются от внешнего хоста. Необязательно иметь контроль над этим хостом, но он должен работать и удовлетворять некоторым требованиям. Вы должны ввести IP адрес хоста-зомби и номер используемого порта. Хотя это сканирование крайне трудно проследить до исходной точки, оно вряд ли особенно полезно для большинства администраторов, сканирующих свои собственные сети. Это одна из самых спорных опций Nmap, так как на практике она применима только для злоумышленных атак

Опции времени Nmap

Nmap предоставляет средства для повышения или понижения частоты, с которой посылаются пакеты сканирования. Если вас беспокоит слишком большой сетевой трафик (или вы пытаетесь действовать скрытно), то можно понизить частоту. Помните только, что чем реже посылаются пакеты, тем дольше продлится сканирование. Для больших сетей время может вырасти экспоненциально. С другой стороны, если вы торопитесь и не обращаете внимание на некоторый дополнительный сетевой трафик, можно поднять частоту. Различные уровни и частоты пакетов приведены в табл. 15 и 16.

Таблица 15

Опция	**Описание**
TCP + ICMP (-PB)	Настройка Nmap, установленная по умолчанию. В процессе такого сканирования программа использует для определения статуса хоста и ICMP, и TCP-пакеты. Это наиболее надежный и точный способ, так как, если хост активен, то хотя бы по одному методу ответ, как правило, будет получен. К сожалению, это также самый шумный способ, который, скорее всего, приведет к регистрации каким-нибудь устройством сканируемой сети
Эхо-тестирование TCP (-PT)	Для обнаружения хостов используется только метод TCP. Многие межсетевые экраны и некоторые маршрутизаторы отбрасывают пакеты ICMP, возможно, с протоколированием. Если вы пытаетесь остаться невидимым, то метод TCP - это наилучший вариант. Однако для некоторых экзотических типов сканирования (FIN, XMAS, NULL) какие-то хосты могут остаться незамеченными.
Эхо-тестирование ICMP (-PE)	Использовать для раскрытия сети только пакеты ICMP. Это не лучший вариант, если вы сканируете сеть извне через межсетевой экран, так как большинство ваших пакетов будет, вероятно, отброшено. Однако внутри сети данный метод вполне надежен, хотя вы можете пропустить свой межсетевой экран и некоторые сетевые устройства, которые не отвечают на ICMP-пакеты

Без эхо-тестирования (-P0)	Если задается эта опция, то Nmap не будет пытаться сначала выяснить, какие хосты активны, а будет вместо этого посылать пакеты по каждому IP-адресу заданного диапазона, даже если по этому адресу машины нет. Это расточительно как с точки зрения полосы пропускания, так и времени, особенно когда сканируются большие диапазоны. Однако это может быть единственным способом просканировать хорошо защищенную сеть, которая не отвечает на ICMP-пакеты.

Таблица 16.

Уровень частоты	Параметр командной строки	Частота пакетов	Пояснения
Параноидальный	-F 0	Раз в 5 минут	Не используйте эту опцию при сканировании большого числа хостов, иначе сканирование никогда не закончится.
Исподтишка	-F 1	Раз в 15 секунд	
Вежливый	-F 2	Раз в 4 секунды	
Нормальный	-F 3	Со скоростью работы ОС	Используется по умолчанию
Агрессивный	-F 4	То же, что и Normal, но максимальное время ожидания пакета сокращено до 5 минут на хост и до 1,25 секунды на зондирующий пакет.	
Безумный	-F 5	Время ожидания 0,75 секунды на хост и 0,3 секунды на зондирующий пакет.	Этот метод не будет хорошо работать, если только вы не находитесь в очень быстрой сети и не используете очень быстрый сервер Nmap. Даже в этом случае есть риск потерять данные.

Другие опции Nmap

В табл. 17 перечислены некоторые другие опции Nmap, которые управляют, например, разрешением доменных имен, идентификацией ОС и т.д., и не попадают в другие категории

Таблица 17.

Опция	Описание
Не выполнять разрешение имен (-n)	Обычно Nmap пытается разрешать доменные имена для всех сканируемых IP-адресов. Это может существенно затягивать сканирование, поэтому если вас не интересуют имена хостов, разрешение имен можно отключить Помните, однако, что знать имена хостов полезно, особенно при сканировании сетей с DHCP, где IP-адреса могут меняться
Быстрое сканирование (-F)	Эта опция вызывает сканирование только портов, перечисленных в файлах употребительных портов Nmap. По умолчанию это общеупотребительные серверные порты с номерами, меньшими 1024. Данные файлы можно отредактировать и добавить в список другие порты. Подобное сканирование может оказаться значительно более быстрым, но оно не выявит троянские

	программы и сервисы, использующие порты с большими номерами.
Диапазон портов (-p диапазон_портов)	По умолчанию Nmap сканирует все 65535 возможных портов TCP. Однако, если вы хотите просканировать только определенный диапазон, можно задать его в качестве аргумента опции -p. Это полезно, если вы хотите просканировать только один тип серверов, например, порт 80 для Web-серверов, или только верхние диапазоны, чтобы найти необычные сервисы и потенциальные троянские программы.
Использование приманок (-D адрес_приманки_1, адрес_приманки_2...)	Эта опция создает видимость, что хосты, указанные в качестве приманок, участвуют в сканировании целевых машин. Последние будут наблюдать потоки данных из нескольких источников, и им будет трудно определить, какой из них является реальным сканирующим хостом. Это еще одна опция сверхскрытности, не обязательная для большинства добропорядочных применений и создающая, кроме того, существенно более высокую нагрузку на сеть. Следует учитывать также, что использование хостов в качестве приманок может привести к блокированию их доступа к сканируемой машине. На вас может обрушиться гнев людей, которых вы таким образом "подставили".
Фрагментация (-f)	Данная опция вызывает фрагментацию отправляемых пакетов сканирования. Это - средство обеспечения скрытности, которое можно применять, чтобы избежать обнаружения сканирования. Пакеты будут собираться на другом конце получающей их машиной, но фрагментированные пакеты могут обмануть системы обнаружения вторжений и межсетевые экраны, которые зачастую проверяют соответствие конкретным шаблонам
Запрашивать информацию Identd (-I)	Служба Identd функционирует на некоторых (обычно - LINUX) машинах и предоставляет при запросе дополнительную информацию о хосте, например, тип операционной системы. Следует учитывать, что Nmap автоматически выполняет идентификацию ОС с помощью идентификационных меток TCP, поэтому данная опция менее полезна, чем кажется на первый взгляд. Если в вашей сети нет систем LINUX, то применение этой опции вообще теряет смысл
Разрешать все адреса (-R)	При использовании данной опции Nmap пытается разрешать все адреса в диапазоне, даже когда они не отвечают. Это может быть полезно, например, в сети поставщика Интернет-услуг, где целый диапазон записей о хостах может быть присвоен потенциальным IP-адресам для пула коммутируемого доступа, но в каждый момент времени возможно использование только определенной части из них.
Идентификация ОС (-O)	Эта опция включена по умолчанию. Как упоминалось ранее, каждая реализация стека TCP имеет свои особенности. При сравнении точной идентификационной метки ответов с базой данных известных идентификационных меток TCP, Nmap, как правило, может с высокой достоверностью (иногда - вплоть до диапазона версий) идентифицировать ОС, с которой общается. Изредка попадается что-то незнакомое, и тогда ответ TCP печатается внизу отчета. Если вы обнаружите неопределенную сигнатуру, вы сможете помочь в построении базы данных идентификационных меток ОС. Если вы точно знаете, чему она

	соответствует, скопируйте ее и отправьте по электронной почте на адрес группы разработчиков Nmap. Они добавят ее в базу данных, чтобы в будущем при сканировании машины такого типа ее можно было правильно идентифицировать. Все известные Nmap идентификационные метки TCP содержатся в файле nmap-os-fingerprints в каталоге Data установки Nmap.
Отправить через интерфейс (-e имя_интерфейса)	Эта опция заставляет пакеты сканирования отправляться через определенный интерфейс. На практике это необходимо только на машине с несколькими сетевыми платами, или если Nmap не опознает ваш сетевой интерфейс автоматически.

Вывод результатов Nmap

Nmap генерирует отчет, содержащий каждый обнаруженный IP-адрес, выявленные слушающие порты по этим адресам и соответствующие общеизвестные имена сервисов (при наличии таковых). Отчет также показывает, является ли порт открытым, фильтруемым или закрытым. Тот факт, что Nmap получил ответ из порта 80 и напечатал в отчете «http», еще не означает, что на компьютере запущен Web-сервер, хотя, скорее всего, это так. Всегда можно проверить любой подозрительный открытый порт, подключаясь с помощью telnet к нужному IP-адресу с указанием номера порта и анализируя полученный ответ. Если там выполняется web-сервер, то обычно получают ответ, вводя команду GET / HTTP. Должна быть выдана подразумеваемая домашняя страница в необработанном HTML-виде (а не как красивая Web-страница), что послужит подтверждением функционирования сервера. То же самое можно проделать с другими сервисами, такими как FTP и SMTP. Отметим, что в LINUX-версии Nmap кодирует цветом найденные порты в соответствии с их ролью (табл. 18).

Таблица 18.

Цвет	**Описание**
Красный	Данный номер порта присвоен сервису, который предлагает некоторую форму прямого входа в систему (как, например, Telnet или FTP). Зачастую эти сервисы оказываются наиболее притягательными для хакеров
Голубой	Этот номер порта представляет почтовый сервис, такой как SMTP или POP. Подобные сервисы также часто являются объектами хакерских атак
Жирный черный	Эти сервисы могут предоставлять некоторую информацию о машине или операционной системе (как, например, finger, echo и т.д.)
Простой черный	Любые другие идентифицированные сервисы или порты

На рис. 2, показан формат вывода программы. Он позволяет быстро просмотреть отчет и определить, есть ли какие-то сервисы или порты, о которых следует побеспокоиться. Это не означает, что нужно игнорировать все необычные номера, которые не выделены цветом или шрифтом (в версиях LINUX). Троянские программы и ПО для общения часто отображаются как неизвестные сервисы.

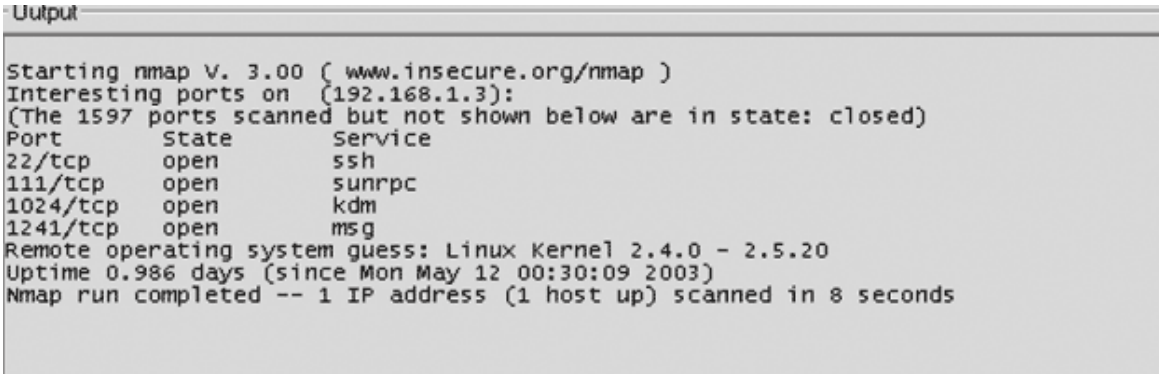

Рис. 2. Вывод Nmap

Журналы Nmap можно сохранять в различных форматах, включая обычный или машиночитаемый текст, и импортировать их в другую программу. Однако, если этих возможностей недостаточно, то существует программное средство обработки отчетов Nlog.

Контрольные вопросы

1. Что такое порт.
2. Какие номера портов закреплены за стандартными сетевыми сервисами. Назовите некоторые номера портов и соответствующие им сервисы.
3. Зачем нужны сканеры портов.
4. Могут ли сканеры портов использоваться в сетях не использующих протоколы TCP/IP.
5. Что означает SYN-сканирование.
6. Назовите автора-разработчика программы NMAP.
7. Какая программа обрабатывает отчеты NMAP.
8. Что означает красный цвет в отчете NMAP, черный.
9. Какой режим сканирования задается командой nmap -PB.

Выполнение работы

1. Выполнить сканирование учебной локальной сети. Определить количество активных компьютеров и всю доступную информацию по IP-адресам и MAC-адресам, установленной операционной системе.
2. Выбрать три целевых компьютера в учебной локальной сети и выполнить для каждой цели полное сканирование сетевых портов. Определить работающие на целевых компьютерах сервисы, их версии, доступную дополнительную информацию.
3. Пояснить в отчете как влияет на результат сканирования работа межсетевого экрана.

Занятие №2 Изучение принципа работы сканера уязвимостей на примере программы Nessus

Продолжительность- 2 часа
Максимальный рейтинг- 6 баллов

Цель работы

Познакомиться с программой-сканером безопасности Nessus [2]. Изучить интерфейс клиентского приложения программы, выполнить пробное сканирование внутренней сети.

Общая информация

Nessus - это инструмент для автоматизации проверки и обнаружения известных уязвимостей и брешей в защите. Обычно кто-то - хакерская группа, компания по разработке ПО для обеспечения безопасности или даже простой пользователь обнаруживает новую уязвимость в каком-либо ПО. Эта уязвимость может быть найдена случайно или же путём тщательного поиска. Nessus – это программа со множеством возможностей, однако она довольно сложна в использовании. Nessus распространяется по лицензии GPL (General Public License). Сайт программы www.nessus.org. Платная техническая поддержка Nessus всегда доступна на www.tenablesecurity.com. Nessus разрабатывался большим коллективом разработчиков, руководимым Renaud Deraison. Сравнительное тестирование программ-сканеров безопасности показывает, что возможности Nessus равны или даже превосходят возможности некоторых продуктов, стоящих тысячи долларов. Важная особенность Nessus – используемая в программе технология клиент-сервер. Сервер может быть размещён в самых разнообразных стратегических точках сети, позволяя осуществлять проверки разных участков сети. Центральный клиент или многочисленные клиенты могут контролировать сервер. Сервер может быть запущен практически на всех разновидностях Unix систем. Он даже может быть запущен на MAC OS X и IBM/AIX, но проще всего его установить на Linux системах. Клиентом может быть как Windows, так и UNIX. Nessus сервер осуществляет действительную проверку безопасности, в то время, как клиент обеспечивает удобный для работы интерфейс.

Программа Nessus использует другие программы: **NMAP** – программу сканер портов, **Hydra** – программу тестер паролей, и наконец **Nikto** – CGI сканер. Хотя эти программы не обязательны для установки, но они очень сильно увеличат работоспособность Nessus. Рекомендуются именно они, потому что эти программы считаются лучшими утилитами в своем классе под UNIX системы.

Настройка программы

После успешной установки сервера, требуется выполнить некоторые базовые шаги. Первым шагом, завершающим установку, будет добавление нового пользователя. Новый пользователь может быть добавлен командой «nessus-adduser», при этом программа задаст вам вопрос о способе аутентификации. Аутентификация может быть осуществлена различными способами, однако, аутентификация при помощи пароля является самым простым и в тоже время самым надёжным способом. Следующим вопросом будет частичная блокировка возможностей некоторых пользователей. Во время использования Nessus пользователь может быть полностью блокирован или допущен к сканированию только некоторых IP адресов. Следующим шагом необходимо сгенерировать сертификат для шифрования трафика между клиентом и сервером. Эту задачу выполняет команда «nessus-mkcert».

Прежде чем начинать сканирование, следует обновить сценарии сканирования. Сценарии сканирования Nessus можно сравнить с коллекцией отпечатков вирусов обычного антивируса. Каждый сценарий посвящен отдельной уязвимости. Сценарии могут быть написаны как для непосредственной эксплуатации уязвимостей, так и для простого

определения версии уязвимого ПО. Сценарии могут быть написаны практически на любом языке, но обычно для этого используется Nessus Attack Scripting Language (NASL). NASL – это собственный язык Nessus, специально разработанный для написания сценариев проверки уязвимостей. Сценарии NASL обычно тестируются, посылая на проверяемый хост особый код и сравнивая результаты с результатами уязвимых значений. Есть лишь небольшое число сценариев, написанных не на NASL. Это C и Perl скрипты, осуществляющие особые цели, которые не так просто создать при помощи NASL. Обновление сценариев желательно делать ежедневно. Новые уязвимости обнаруживаются и распространяются постоянно. Обычно, после появления новой уязвимости кто-то из разработчиков Nessus пишет NASL сценарий и публикует его реализацию на www.nessus.org. Далее это сценарий рассматривается разработчиками и помещается в список одобренных сценариев. Из-за высокого риска, серьёзные сценарии часто выходят в один день с появлением информации о новой критической уязвимости. Обновление сценариев осуществляется командой «nessus-update-plugins». На выполнение этой команды имеет право только пользователь root. Однако, обычные пользователи не ограничены никакими средствами от просмотра списков сценариев на www.nessus.org. Новые или предназначенные для какой-либо особой цели сценарии могут быть написаны на NASL, так что вы можете создавать ваши собственные сценарии.

Для запуска сервера необходимо выполнить команду «nessusd –D». Для использования программы нужно установить клиента. Существует три основных клиента Nessus. Стандартная Unix GUI версия автоматически устанавливается в процессе установки сервера. Nessus можно также контролировать из командной строки. Третий вариант – это существующая версия под Windows, которая называется NessusWX.

Подключение клиента

Так, как Nessus - это технология клиент-сервер, то, запуская клиента, необходимо подключиться к серверу. В стандартном клиенте введите IP адрес сервера, имя пользователя и пароль (пользователя можно создать командой «nessus-adduser») и нажмите login. В NessusWX процесс проще, но использует меню подключения соединения. Клиент подключается к серверу через SSL соединение и загружает список установленных на данный момент сценариев. Эта проверка гарантирует, что и в дальнейшем вы будете держать связь именно с предназначенным сервером. Рисунки 3 и 4 демонстрируют подключение, используя Unix и Windows GUI средства. Рисунок 5 показывает аутентификацию пользователя, с использованием NessusWX клиента.

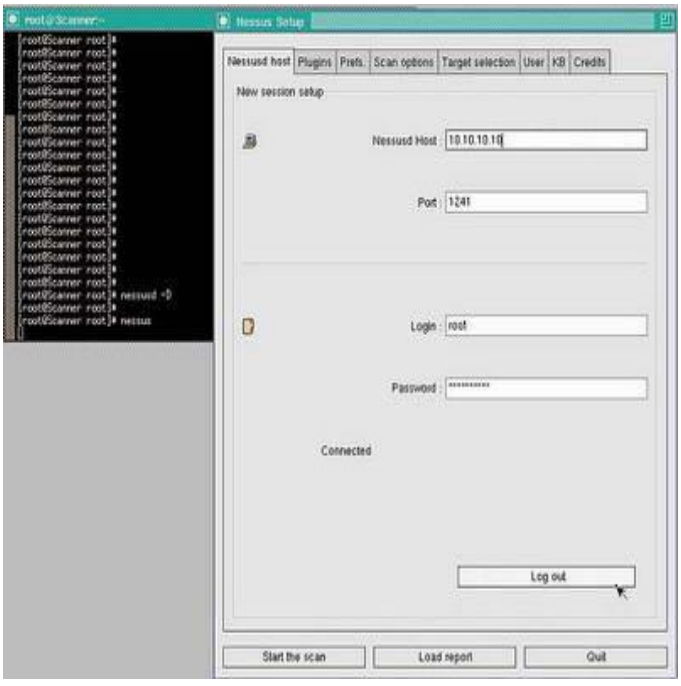

Рис. 3. Запуск Nessus в Unix GUI.

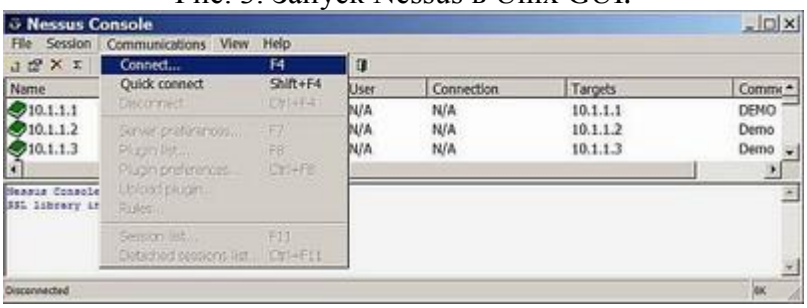

Рис. 4. Запуск Nessus в Windows.

Рис. 5. Аутентификация пользователя с использованием NessusWX клиента.

Использование Nessus

Самой главной и сильной особенностью Nessus являются сценарии. Выбор сценариев – это залог успеха всего сканирования. Сценарии делятся на категории несколькими способами. Один из методов группировки – группировка по степени опасности применения сценариев сканирования. Различают опасные и неопасные сценарии. Некоторые сценарии считаются опасными, так как могут вызвать отказ в обслуживании (DoS) и аварийное завершение работы системы, уязвимой для DoS атаки. Разумеется, такие сценарии не стоит вслепую запускать на функционирующей системе. Они не вызывают серьёзного ущерба, но требуют, как минимум, перезагрузки системы. Стандартный клиент обозначает опасные сценарии предупреждающим треугольником. NessusWX не имеет никакого специального предупреждения об опасных сценариях, за исключением функции Enable Non-DoS. И никакого другого предупреждения, что даже безопасный сценарий может повлечь за собой крах и неработоспособность системы, вы не найдёте. Так, как сценарии посылают нестандартный набор данных, риск во время их использования существует всегда, хотя реальная угроза бывает довольно редко. Поэтому каждый, кто сканирует систему, должен понимать, что это может повлечь за собой повреждение, даже, когда используются на первый взгляд безопасные сценарии. Рисунок 6 показывает выбор сценариев с использованием Unix GUI. Рисунки 7 и 8 показывают выбор сценариев в NessusWX для Windows.

Рис. 6. выбор сценариев с использованием Unix GUI.

Рис. 7. Выбор plug-ins с использованием Windows NessusWX клиента.

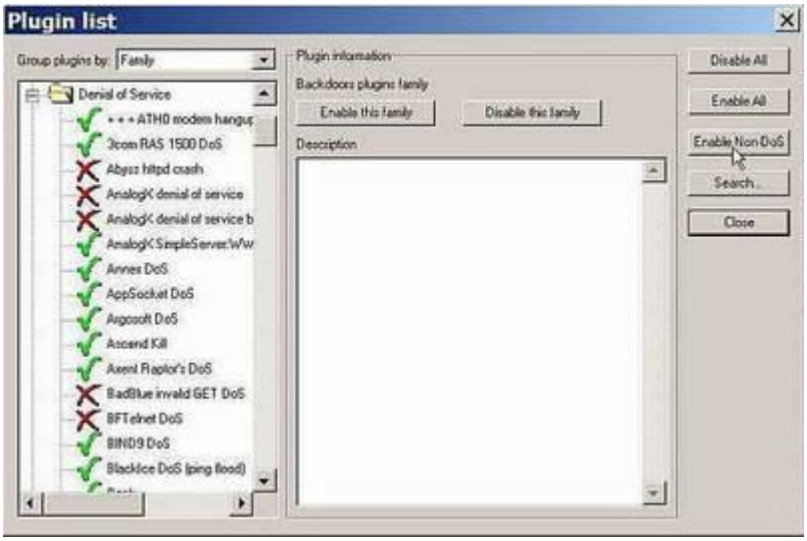

Рис. 8. Включение безопасных plug-ins в Windows NessusWX.

Безопасные проверки не могут нанести вреда, они используют лишь пассивные сценарии и направлены, например, на определение версии используемого программного обеспечения. Однако безопасные проверки не всегда бывают корректными. Иногда они могут выдавать ложные предупреждения или пропускать опасные уязвимости. Режим безопасного сканирования нужно выставить на панели Scan options. Рисунок 9 изображает безопасную проверку в NessusWX.

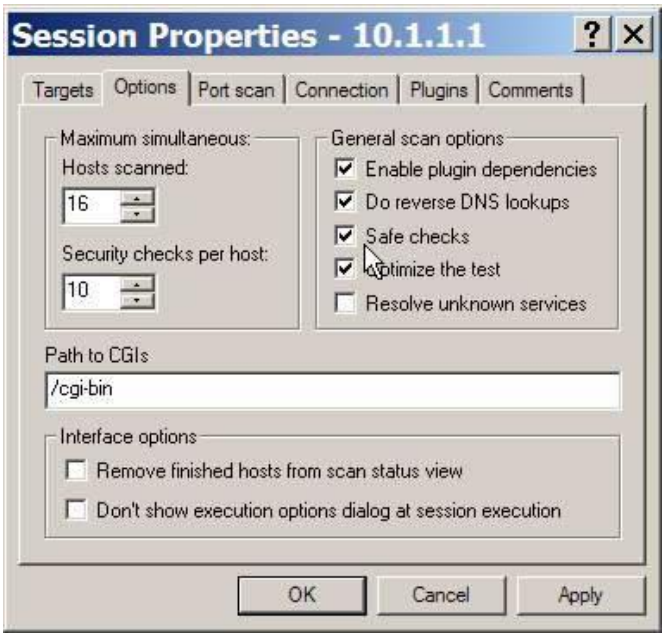

Рис. 9. Выбор безопасных проверок

Другой способ организации сценариев – это организация по таким категориям, как Windows, FTP, SNMP, SMB, Cisco, и т.д. У этого способа организации есть недостатки. Например, куда следует отнести сценарий FTP применимый только для Windows, - к категории Windows или FTP? Ведь выбор категории делает автор сценария, а не кто-то другой. Для облегчения работы был создан механизм фильтрации сценариев. Механизм фильтрации помогает изолировать лишние сценарии. Фильтр может быть установлен на имя сценария, номер сценария и т.п. Кликните на категорию сценария, и вы получите подробности работы и действия этого сценария. Если вам нужна более подробная информация о нём, например NASL код сценария, то вы можете его найти на cgi.nessus.org/plugins/. Заметьте, что сценарии категории «DoS» и сценарии категории «опасные / DoS» это разные вещи. «Опасные / DoS» категории сценариев используют известную уязвимость, в то время как

сценарии категории «DoS» чаще всего только проверяют на наличие уязвимости. Если требуется открыто просканировать систему на наличие уязвимостей, не опасаясь за последствия, то наилучшим решением является выбор всех сценариев. Однако если целью является скрытое сканирование или происходит сканирование рабочей системы, то выбор сценариев становится нелёгким делом.

Сканирование портов – одна из важных частей процесса сканирования. Это процесс, во время которого определяются активные порты для определённого IP адреса. Каждый порт связан с определённым приложением. Nessus – это разумный сканер, и он использует сценарии только в том случае, если будет найдена программа для проверки на уязвимость. Например, сценарии для веб-сервера будут использованы только в случае, если будет найден веб-сервер. Так как довольно часто порты используются не по их стандартному назначению, для того, чтобы определить их, Nessus использует сценарии, называемые сервисами (services). Сервисные сценарии стараются определить программу, запущенную на каждом порте. Как только программа определена, против неё запускаются сценарии, выбранные пользователем.

Nessus позволяет использовать несколько различных способов сканирования портов. Первый - это классический NMAP сканнер, который получил широкую известность. В программе есть и встроенный сканер портов и сканер на основе программы ping. Насколько важен грамотный выбор сценариев, настолько же важен и выбор способа сканирования, зависящий от ситуации. Рисунки 10 и 11, показывают обычное SYN сканирование, используя NessusWX и Unix GUI клиента, соответственно:

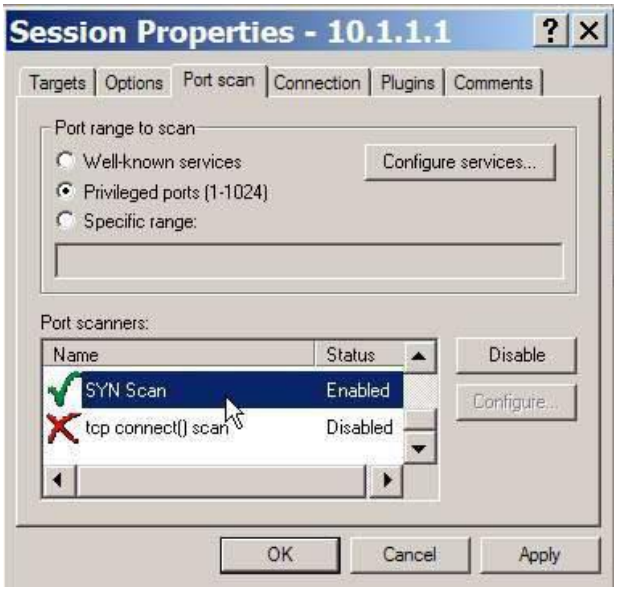

Рис. 10. Выбор SYN сканирование в NessusWX.

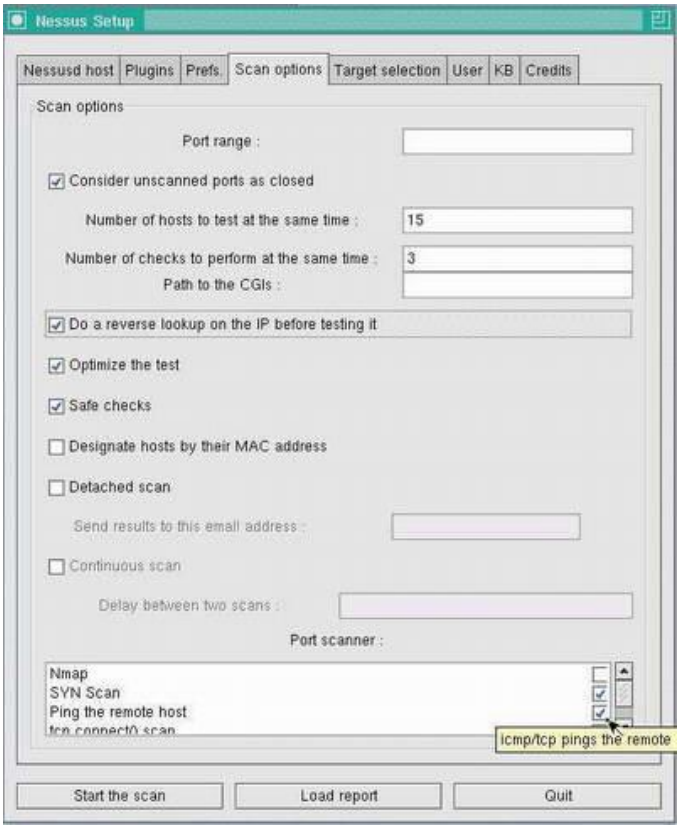

Рис. 11. Выбор SYN сканирование в Unix клиенте.

Конечная задача – определить объекты сканирования. Цель сканирования указывается на панели Target Selection. Объектом сканирования может быть как простой IP адрес, так и подсеть или даже диапазон IP адресов. Рисунки 12 и 13 показывают, как выбрать цель в клиентском приложении.

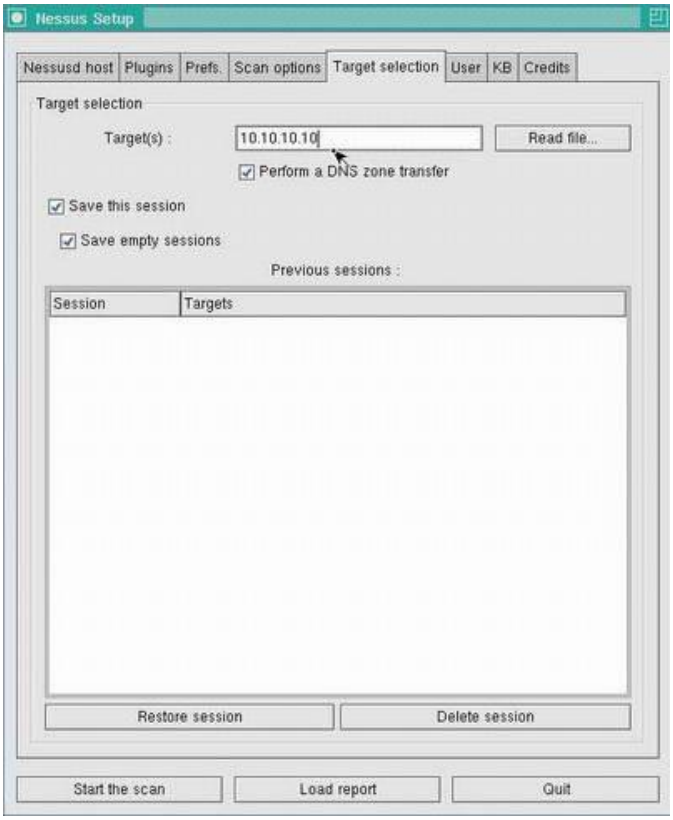

Рис. 12. Выбор цели сканирования в Unix GUI.

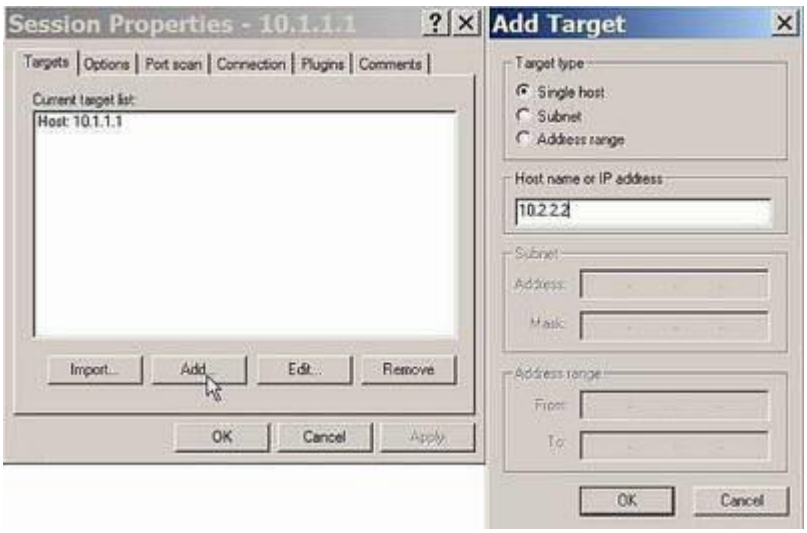

Рис. 13. Выбор цели сканирования в NessusWX.

Контрольные вопросы

1. Какие задачи решает программа Nessus.
2. В чем преимущества программы Nessus.
3. Чтотакое NASL.
4. Почему некоторые сценарии, выполняемые программой, считаются опасными.
5. Какой командой запускается серверная часть программы.
6. Доступ к программе осуществляется через специальное клиентское приложение, а в нем необходимо ввести регистрационную информацию, что делать если вы еще не зарегистрировались, сможете ли вы воспользоваться программой.
7. Может ли Nessus использовать другие программы, какие.

Выполнение работы

1. Запустить серверную часть программы.
2. Запустить клиентскую часть, зарегистрироваться.
3. Выполнить пробное сканирование в учебной локальной сети.
4. Выполнить сканирование на уязвимости одной из выбранных целей в учебной локальной сети.

Занятие №3 Создание защищенных документов в текстовом процессоре OpenOffice.Org Writer

Продолжительность- 2 часа
Максимальный рейтинг- 6 баллов

Цель работы

Изучить технологию создания защищенных документов и документов, подписанных цифровой подписью в текстовом процессоре OpenOffice.org Writer.

Создание защищенного документа

Все документы, которые сохраняются в XML-формате (формат используемый в OpenOffice.org Writer по-умолчанию, расширение .odt), можно сохранять с паролями [3, 4]. Документы, сохраненные с паролем, нельзя открыть без пароля. Содержимое защищается таким образом, чтобы его нельзя было читать с помощью какого-либо внешнего редактора. Шифруются текстовое содержимое, графические изображения и объекты OLE. Информация, введенная в диалоговом окне **Файл - Свойства** не шифруется. Она включает имя автора, дату создания, количество слов и символов. Для включения защиты необходимо выбрать команду **Файл - Сохранить как** и в открывшемся диалоговом окне (рис. 14) установить флажок **Сохранить с паролем**.

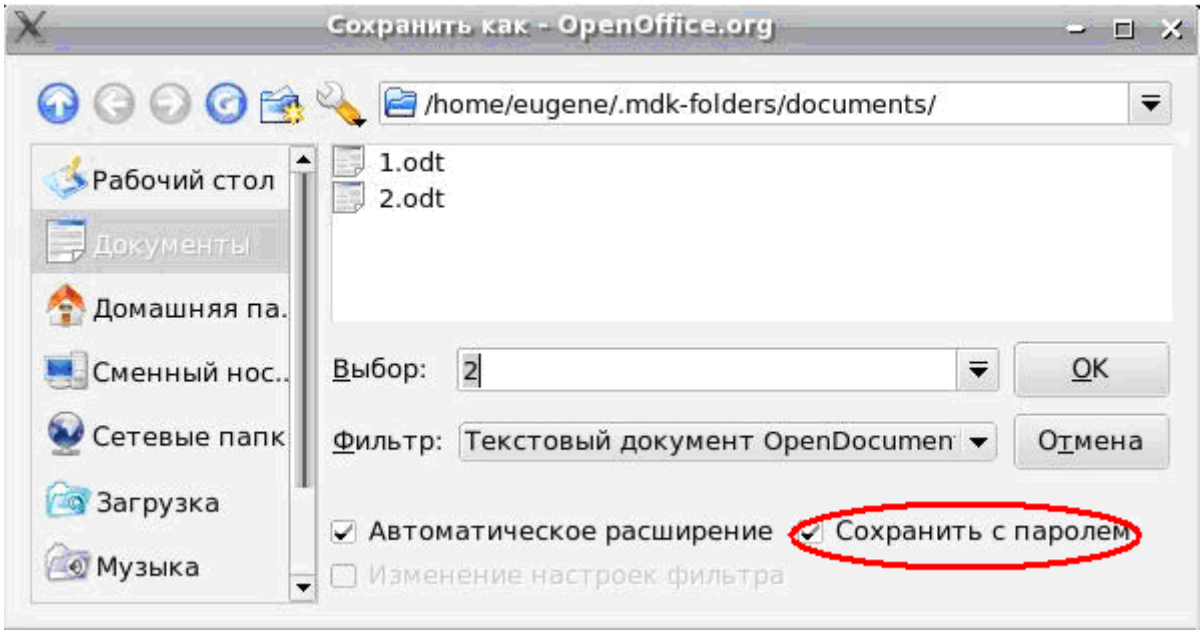

Рис. 14. Сохранение документа в зашифрованном виде.

Если поле установки флажка нажать клавишу «ОК», то появится окно с запросом пароля, рис. 15., используемого как ключ шифрования.

Рис. 15. Окно введения пароля для шифрования документа.

Для отключения защиты сначала необходимо открыть документ, введя правильный пароль.

Затем, выбрав команду **Файл - Сохранить как**, вновь попасть в окно сохранения документа (рис. 14) и снять флажок **Сохранить с паролем.** При попытке открыть защищенный документ появляется окно запроса пароля (рис. 15).

Создание документа с цифровой подписью

Прежде, чем подписывать документы OpenOffice.org, необходимо получить (или создать) личный цифровой сертификат (digital certificate) и установить его на свою машину. Цифровой сертификат это защищенный паролем файл в котором хранится различная информация - имя владельца, его e-mail адрес, ключ шифрования, а также наименование организации выдавшей этот сертификат и дату, после которой цифровой сертификат считается недействительным (expiration period). Большинство организаций выдающие сертификаты делают это на коммерческой основе и требуют оплаты за их выдачу. Но существуют и некоммерческие организации, например ассоциация **CACert** (http://www.cacert.org), выдает цифровые сертификаты бесплатно. Можно создать и так называемый самоподписанный цифровой сертификат (self-signed certificate).

Цифровой сертификат (Digital certificate) - небольшой файл, содержимое которого уникальным образом идентифицирует пользователя или сайт, показывая, что вы можете доверять определённой информации. Цифровой сертификат содержит открытый ключ владельца сертификата, его персональную информацию, а также цифровую подпись сертифицирующей организации.

Цифровая подпись (Digital signature) информация об организации или личности, зашифрованная с помощью закрытого ключа этой организации. Удостоверяет, что документ исходит от того лица, чья цифровая подпись прилагается. Подпись нельзя подделать, т.к. закрытый ключ недоступен. Кроме того, внесение изменений в документ приводит к разрушению цифровой подписи, т.е. наличие цифровой подписи в документе не только подтверждает личность отправителя, но и свидетельствует о том, что документ никем кроме отправителя не изменялся.

Для получения сертификата от CACert достаточно зарегистрироваться на сайте этой организации. Заполнив поля электронной формы с запросами о вашей персональной информации, через некоторое время вы получаете на свою электронную почту ссылку на подтверждение запроса. Переход по этой ссылке приводит к генерированию персонального сертификата и вы можете установить его на вашу машину.

> *Для выполнения указанных процедур требуется поддержка Java, поэтому в вашем браузере должна быть разрешена ее работа.*

Сертификат CACert устанавливается в программу-браузер, после установки его необходимо экспортировать в виде отдельного файла, который можно сохранить в любой доступный каталог.

Для создания самоподписываемого сертификата можно использовать бесплатную программу SELFCERT от фирмы Abylonsoft (http://www.abylonsoft.com/).

После получения сертификата можно подписать документ цифровой подписью. Для этого нужно выполнить следующие шаги:

1. В меню **Файл** выбрать пункт **Цифровые подписи.**
2. В окне сообщения выводится рекомендация сохранить документ. Нажмите кнопку **Да**, чтобы сохранить файл.
3. После сохранения будет открыто диалоговое окно «Цифровые подписи». Нажмите кнопку **Добавить,** для создания цифровой подписи.
4. В диалоговом окне «Выбор сертификата» выберите сертификат и нажмите кнопку "OK".
5. Будет снова открыто диалоговое окно «Цифровые подписи», в котором можно добавить любые необходимые дополнительные сертификаты. Нажмите кнопку «ОК», чтобы добавить общий ключ к сохраненному файлу.

Для подписанного документа в строке состояния отображается значок. Чтобы просмотреть сертификат, можно дважды щелкнуть этот значок, рис. 16.

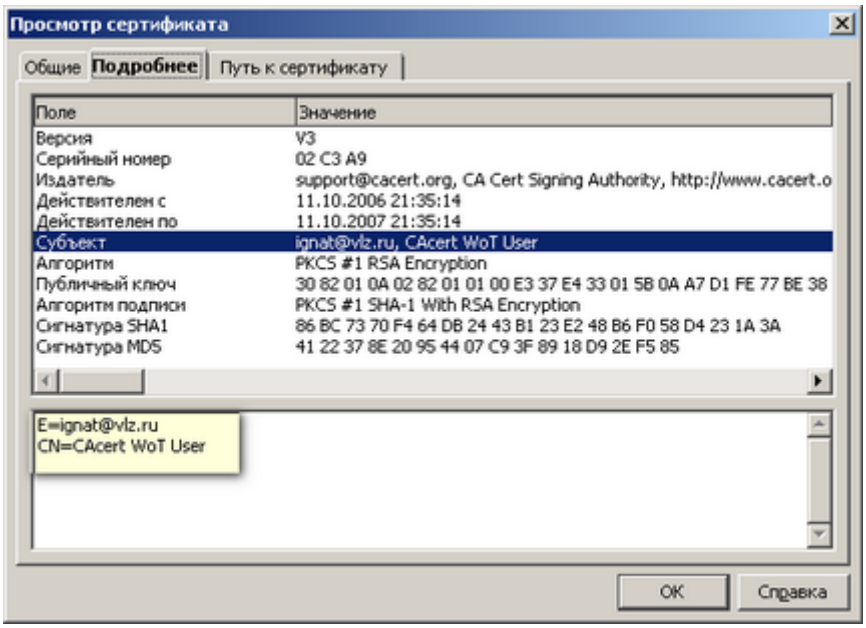

Рис. 16. Окно просмотра свойств сертификата.

Контрольные вопросы

8. Можно ли в текстовом процессоре OpenOffice.org Writer создать зашифрованный документ.
9. Что такое цифровой сертификат.
10. Обеспечивает ли цифровая подпись защиту документа.
11. Откуда можно взять сертификат. Можно ли его изготовить самостоятельно.
12. Может ли текстовый процессор OpenOffice.org Writer снимать защиту с зашифрованных документов Microsoft Word.
13. Какой формат используется для хранения текстовых документов в OpenOffice.org Writer.
14. Можно ли создать в OpenOffice.org Writer защищенный паролем документ формата .doc.

Выполнение работы

1. Создать в OpenOffice.org Writer защищенный паролем документ. Открыть этот документ в простом редакторе (например, встроенном в Midnight Commander), проанализировать его содержимое. Какую информацию можно извлечь из зашифрованного файла.
2. Получить сертификат от CACert. Создать в OpenOffice.org Writer документ с цифровой подписью на основе полученного сертификата. Что происходит с документом при попытке внесения в него изменений.

Занятие №4 Шифрование почтовых сообщений в программе Mozilla Thunderbird

Продолжительность- 2 часа
Максимальный рейтинг- 6 баллов

Цель работы

Изучить технологию шифрования и дешифровки зашифрованных почтовых сообщений на примере почтовой программы Mozilla Thunderbird.

Введение

Известно [5], что высокоэффективное шифрование отдельных файлов электронных документов, почтовых сообщений, каталогов и т.п. обеспечивает программа PGP (Pretty Good Privacy – довольно хорошая секретность), разработанная Филиппом Циммерманом (Philip Zimmermann) и использующая схему шифрования с асимметричными ключами. Такая схема основана на математических алгоритмах, позволяющих шифровать сообщение одним ключом, а расшифровывать другим. Программа, использующая асимметричный алгоритм шифрования, генерирует пару связанных ключей. Особенность алгоритма шифрования в том, что расшифровать сообщение можно только парным ключом, соответствующим ключу с которым выполнялось шифрование, но нельзя ключом которым выполнялось шифрование. Ключи стали именоваться открытый и закрытый, т.к. один из них (открытый) может распространяться свободно, тогда как закрытый хранится в надежно защищенном месте. Для вычисления парного ключа на основе имеющегося, например, открытого требуются слишком большие вычислительные мощности и огромные временные периоды, поэтому нет никакой опасности в свободном распространении одного из ключей. Отправив всем своим адресатам открытый ключ мы даем им возможность шифровать этим ключом сообщения, предназначенные для нас. Никто другой расшифровать их не сможет, потому что закрытый ключ – второй из пары ключей – необходимый для расшифровки хранится у нас и мы его никому не передаем. Даже сами наши адресаты, зашифровав сообщения нашим открытым ключом не смогут их затем расшифровать, т.к. не имеют закрытого ключа.

Симметричность процесса шифрования позволяет выполнять и обратную операцию – шифрование с помощью закрытого ключа. В этом случае расшифровка должна выполняться открытым ключом, т.е. расшифровать такое сообщение могут все, у кого есть открытый ключ, потенциально все желающие, т.к. открытый ключ распространяется свободно и может быть общедоступен. Такое шифрование применяется для однозначной идентификации отправителя и называется **цифровой подписью**. Отправитель шифрует своим закрытым ключом свои идентификационные данные – имя и фамилию, название компании и т.п. Поскольку только у него храниться закрытый ключ шифрования, расшифровав персональную информацию отправителя открытым ключом, мы убеждаемся, что отправитель именно тот человек (банк, компания) за кого он себя выдает.

Ранние версии программы PGP были бесплатными, однако ее современные реализации являются коммерческими. В качестве альтернативного программного пакета в рамках FSF (Free Software Foundation) и под лицензией GPL (General [GNU] Public License) распространяется и получает все большую популярность пакет GnuPG (GPG), реализующий алгоритм шифрования, аналогичный PGP. Шифрование почтовых вложений и самих писем с помощью этого пакета поддерживается популярным почтовым клиентом Mozilla Thunderbird. Программа GPG работает из командной строки и выполняет операции шифрования или дешифровки зашифрованных файлов. Для встраивания программы в почтовый клиент и автоматизации операций шифрования/дешифровки разработан программный модуль Enigmail, делающий задачу отправки закрытой почты простой и удобной дополнительной функцией почтового клиента.

В ходе практического занятия отдельно будут рассмотрены процедуры шифрования персональной информации с помощью закрытого ключа, иначе говоря, создание цифровой подписи электронного сообщения и шифрования с помощью открытого ключа, полученного от адресата, сообщения для этого адресата.

Генерации связанной пары ключей (открытого и закрытого), является необходимым условием выполнения описываемых процедур, но в данной работе не рассматривается и должна быть изучена самостоятельно, так же как и обмен открытыми ключами.

Создание электронного письма с цифровой подписью.

Для создания письма с цифровой подписью необходимо создать новое сообщение. На первом этапе рекомендуется в качестве получателя указать свой обратный адрес, т.е. отправлять письмо самому себе.

> *В этом случае для расшифровки сообщения используется собственный открытый ключ, заботится о получении которого не приходится (ключи всегда генерируются парами). При подготовке сообщения для другого адресата необходимо предварительно передать ему ваш открытый ключ шифрования.*

Введите произвольный текст, под которым хотите подписаться и после того как текст будет готов, кликните мышкой на пиктограмме замка «OpenPGP», рис. 17.

Следует отметить, что эта пиктограмма появляется в Mozilla Thunderbird только после установки программного модуля Enigmail [6].

В раскрывшемся окне необходимо установить флажок в чек-боксе «Sing Message» - подписать сообщение.

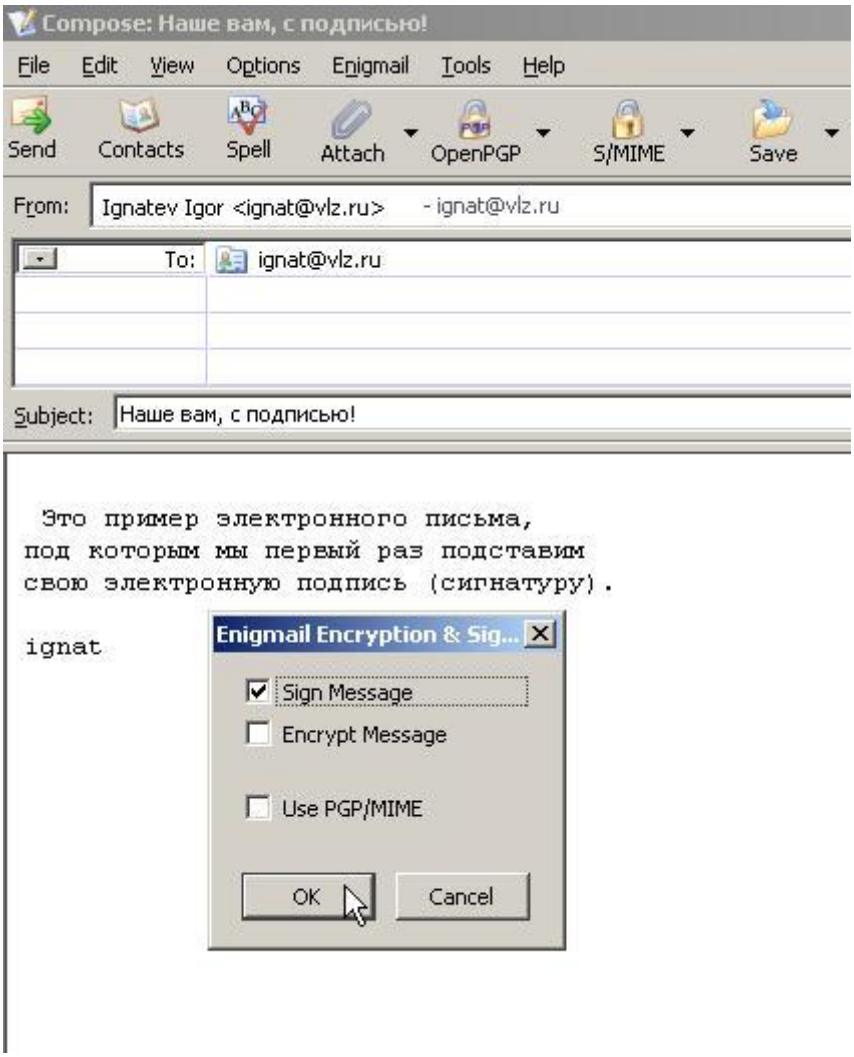

Рис. 17. Формирование почтового сообщения с цифровой подписью.

После этого отправка сообщения на почтовый сервер, вызывает ответный запрос парольной фразы для закрытого ключа шифрования, рис. 18. Такой запрос возникает всегда при обращении к закрытому ключу (ключ хранится на вашем компьютере). Если необходимо отправить несколько почтовых сообщений с цифровой подписью, то, чтобы не вводить каждый раз пароль, в окне запроса можно установить флажок, разрешающий Enigmail запоминать пароль на некоторое время и автоматически его использовать. В примере, показанном на рис.2 это время составляет 5 минут.

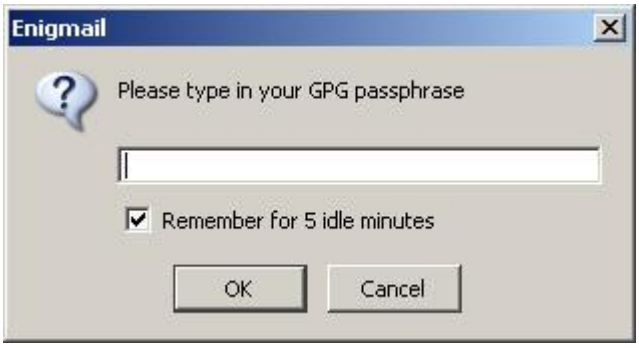

Рис. 18. Запрос парольной фразы для доступа к закрытому ключу.

После ввода парольной фразы модуль Enigmail вычислит и подставит цифровую подпись прямо в текст сообщения. На рис. 19 показано какой вид получает подписанное электронное письмо.

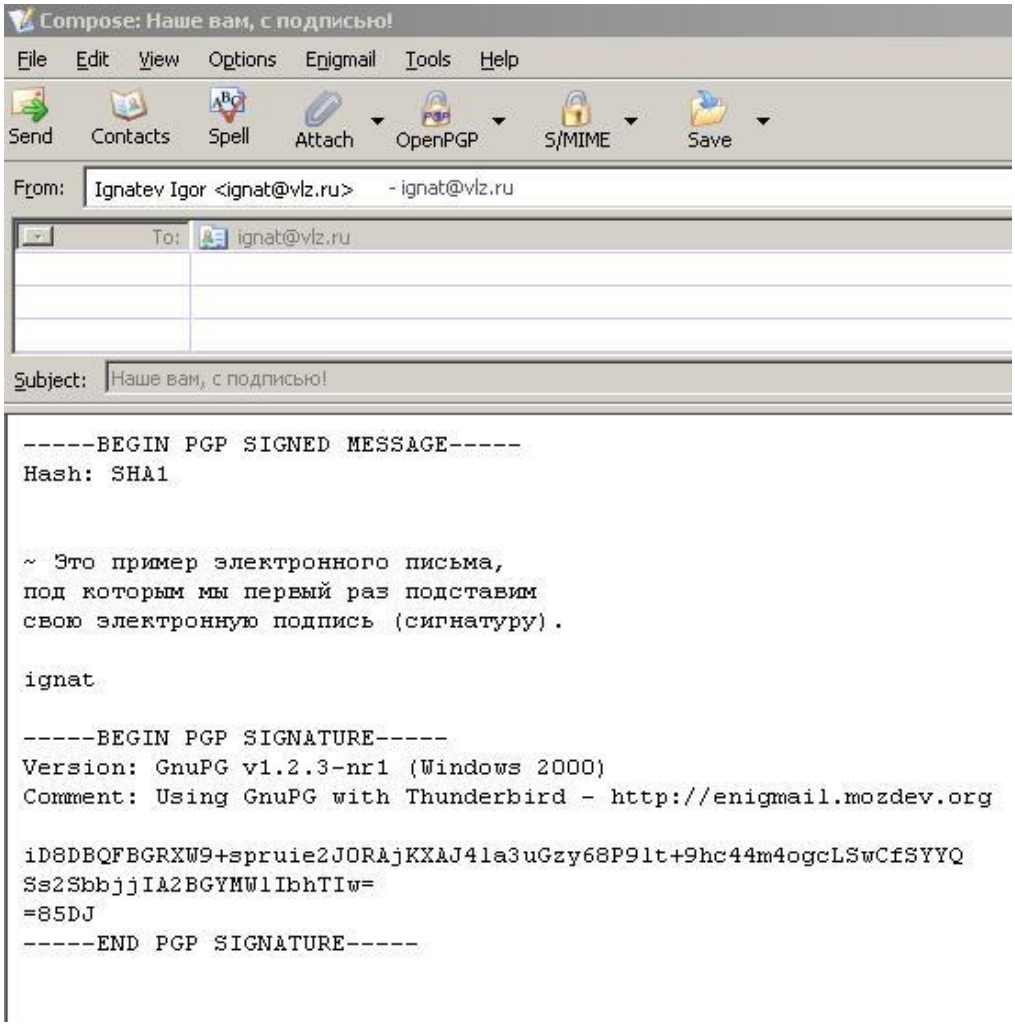

Рис. 19. Электронное письмо с цифровой подписью, подготовленное к отправке.

Заголовок в начале почтового сообщения «BEGIN PGP SIGNED MESSAGE», предупреждает о том, что ниже него содержится текст подписанный с помощью PGP. Сама цифровая подпись располагается ниже текста сообщения, между заголовками «BEGIN PGP SIGNATURE» и «PGP SIGNATURE».

Получение подписанного электронного письма.

При получении подписанного цифровой подписью электронного сообщения модуль Enigmail автоматически определяет тот факт, что письмо содержит цифровую подпись, об этом свидетельствует пиктограмма карандаша в заголовке письма (рис. 20) и сообщение о результате проверки цифровой подписи в статусной строке.

Разумеется, проверка завершится успешно только в том случае, когда модуль Enigmail найдет на локальных дисках соответствующий открытый ключ шифрования для отправителя сообщения. Этот ключ должен быть получен и установлен заранее. Пиктограмма в нижнем правом углу письма может принимать несколько типов окраски. Зеленый цвет свидетельствует о том, что проверка завершена успешно. В статусной строке Enigmail, выводится информация о том, что полученное письмо подписано «Ignatev F Igor», идентификатор открытого (публичного) ключа, хранящийся в системе, для которого – KeyID =0xE89ED89ED.

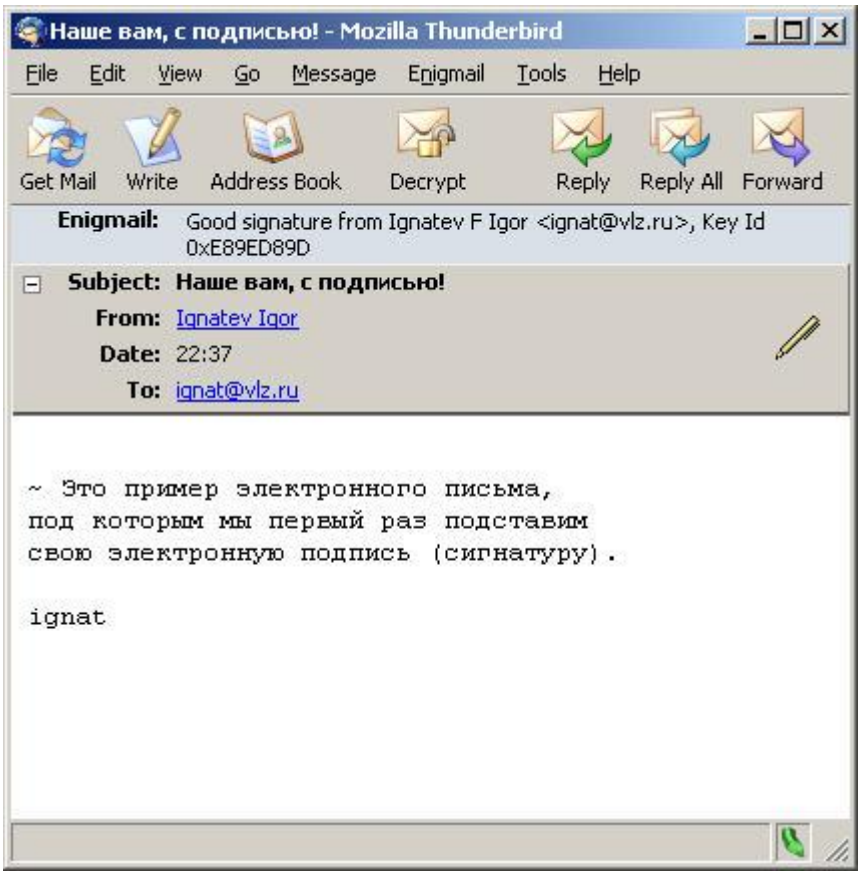

Рис. 20. Письмо с цифровой подписью, полученное адресатом.

Если письмо будет содержать неправильную цифровую подпись, его внешний вид меняется:

- В статусной строке Enigmail выводится предупреждение о том, что цифровая подпись неверная;
- Картинка в заголовке письма превращается в изображение сломанного карандаша;
- Пиктограмма в нижнем правом углу окашивается в красный цвет.

Это – признак того, что вас хотят обмануть. Возможно полученное письмо во время пересылки (случайно или намеренно) подверглось изменению или человек, поставивший подпись, не тот, за кого он себя выдает. Наконец, результат проверки может быть

отрицательным, если отправитель использовал секретный ключ не соответствующий хранящемуся у получателя открытому ключу. Какова бы ни была причина, доверять полученному электронному сообщению нельзя.

Подготовка зашифрованного электронного письма.

Процедура создания зашифрованного электронного письма отличается от создания письма с цифровой подписью только тем, что после нажатия на пикограмму «OpenPGP» в окне рис. 17 необходимо выбрать пункт «Encrypt Message» - зашифровать сообщение.

Модуль Enigmail при подготовке зашифрованного письма предлагает три варианта шифрования, рис. 21:

- Зашифровать только текст сообщения, но не присоединенный файл (файлы);
- Зашифровать только присоединенный файл, оставив текст письма в открытом виде;
- Зашифровать письмо полностью – и текст письма, и присоединенный файл.

Рис. 21. Выбор способа шифрования электронного письма.

В случае выбора полного шифрования используется кодирование по стандарту PGP/MIME. Не все почтовые программы поддерживают этот стандарт. Об этом говорит текст предупреждения в окне выбора способа шифрования. На платформе ОС Windows лишь несколько программ могут работать с кодированными PGP/MIME сообщениями. Это Enigmail, Sylpheed, Pegasus и Mulberry. Ни одна из этих программ не является широко используемой в России. Для избежания проблем с кодировкой рекомендуется выбирать второй вариант и использовать английский алфавит для именования файлов. Если точно известно, что адресат имеет программу, поддерживающую стандарт PGP/MIME (например, Enigmail), то следует выбрать третий вариант. В этом случае имена файлов могут быть и в русской кодировке.

Еще раз отметим важность шифрования электронного сообщения для вашего адресата открытым ключом, полученным от этого адресата и никаким другим. Для запоминания последовательности действий при шифровании можно потренироваться, отправляя зашифрованное сообщение самому себе. В этом случае шифровать его нужно собственным открытым ключом. Программа запрашивает каким из открытых ключей мы собираемся выполнять шифрование, если в хранилище ключей присутствует больше одного ключа. Процесс шифрования выполняется достаточно быстро и конечным его результатом будет письмо в таком виде, как оно показано на рисунке 22.

При получении зашифрованного письма модуль Enigmail автоматически определит его как закодированное. Поскольку процесс расшифровки требует доступа к закрытому ключу, программа попросит ввести пароль. После правильного пароля выполняется

декодирование сообщения и оно становится доступно в исходном виде, рис. 23. На рисунке показаны признаки зашифрованного сообщения – пиктограмма ключа, и результата выполнения его расшифровки – зеленый замочек.

Возможность запоминания пароля на некоторое время позволяет более комфортно работать с большим числом писем от одного адресата.

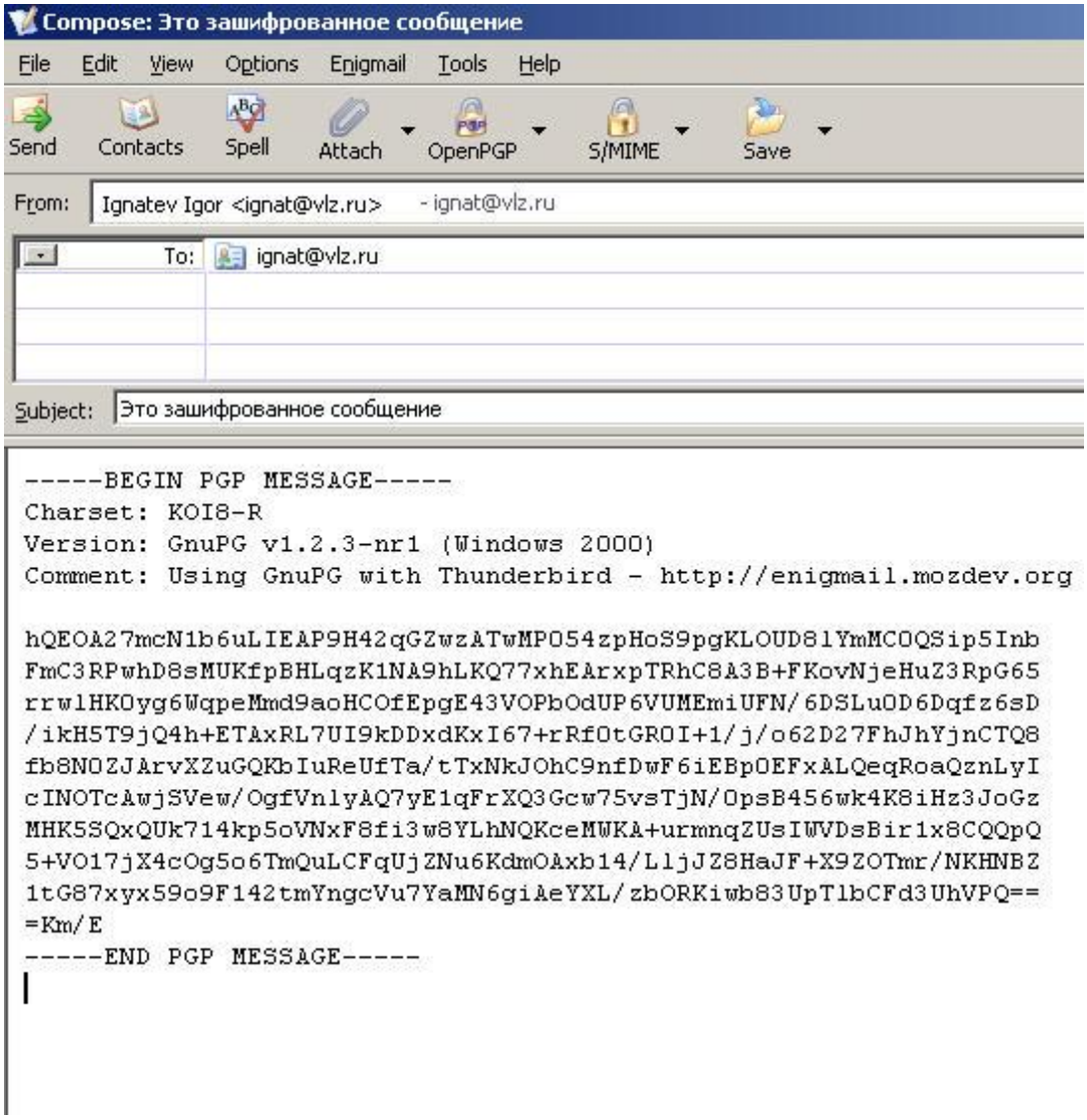

Рис. 22. Зашифрованное электронное письмо.

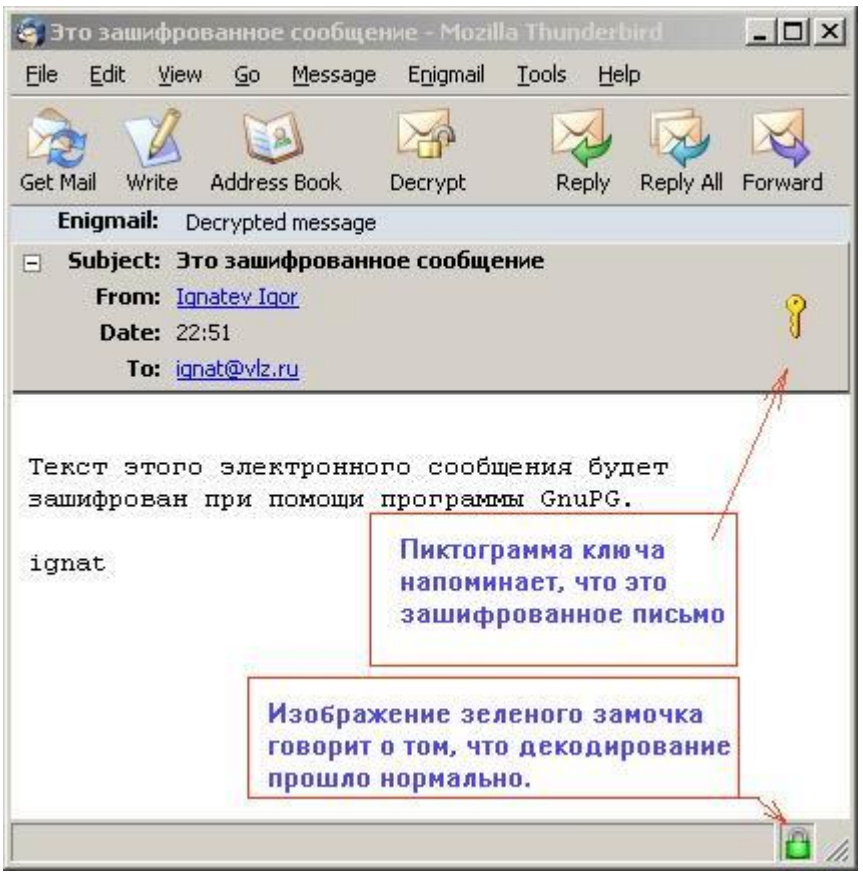

Рис. 23. Расшифрованное сообщение.

Контрольные вопросы

1. Почему для шифрования сообщений электронной почты применяется механизм асимметричного шифрования.
2. Что такое открытый ключ, в каких случаях он применяется.
3. Поясните понятие «цифровая подпись». Зачем нужна такая подпись.
4. Расшифруйте аббревиатуру PGP, GPG, в чем отличия программ, описываемых этими сокращениями.
5. Как называется программный модуль, обеспечивающий шифрование почтовых сообщений в программе Mozilla Thunderbird.
6. Существуют ли программы, взламывающие зашифрованные по ассимметричному алгоритму электронные документы.
7. Что в почтовом сообщении показывает факт присутствия цифровой подписи.
8. Почему мы не можем отправить сообщение адресату, зашифровав его своим открытым ключом.
9. Для чего нужна функция запоминания пароля.
10. Какие проблемы могут возникнуть при отправке зашифрованных сообщений с файлами, имеющими в названии кириллические символы.

Выполнение работы

Работа выполняется в парах отправитель – получатель. Каждый студент выступает для другого отправителем, сам же является получателем сообщений.

1. Убедиться, что в программе установлены необходимые ключи шифрования. Если их нет сгенерировать или получить их.
2. Подготовить для адресата письмо с цифровой подписью. Обменяться сообщениями. Убедиться в их успешном декодировании. Продемонстрировать результат преподавателю.
3. Подготовить для адресата зашифрованное сообщение без вложений. Обменяться сообщениями. Убедиться в их успешном декодировании. Продемонстрировать результат преподавателю.
4. Подготовить для адресата зашифрованное сообщение с вложением. Зашифровать только письмо. Обменяться сообщениями. Убедиться, что вложение доступно без дешифрования. Расшифровать сообщение. Убедиться в его успешном декодировании. Продемонстрировать результат преподавателю.
5. Подготовить для адресата зашифрованное сообщение с вложением. Зашифровать его полностью. Обменяться сообщениями. Расшифровать сообщение. Убедиться в его успешном декодировании. Продемонстрировать результат преподавателю.

Литература

1. Gordon “Fyodor” Lyon. Официальное руководство по использованию программы NMAP - The Official Nmap Project Guide to Network Discovery and Security Scanning. 2011. [Электронный ресурс]. Режим доступа: http://nmap.org/book/toc.html, свободный. Заглавие с экрана.
2. Renaud Deraison, Haroon Meer, Roelof Temmingh, Charl van der Walt, Raven Alder, Jimmy Alderson, George A. Theall. Nessus Network Auditing. Издательство Syngress Publishing, Inc., 2004. — 545 с.
3. Защита содержимого в LibreOffice [Электронный ресурс]. Режим доступа: http://help.libreoffice.org/Common/Protecting_Content_in/ru, свободный. Заглавие с экрана.
4. Защита содержимого в LibreOffice Writer. [Электронный ресурс]. Режим доступа: http://help.libreoffice.org/Writer/Protecting_Content_in_Writer/ru, свободный. Заглавие с экрана.
5. PGP. Википедия, свободная энциклопедия. [Электронный ресурс]. Режим доступа: http://ru.wikipedia.org/wiki/PGP, свободный. Заглавие с экрана.
6. Enigmail. Установка и базовая настройка / OpenSource в заметках. Путь к пониманию. [Электронный ресурс]. Режим доступа: http://www.ashep.org/2010/enigmail-ustanovka-i-bazovaya-nastrojka/, свободный. Заглавие с экрана.