МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

ПРОГРАММНО-АППАРАТНАЯ ЗАЩИТА ИСПОЛНЯЕМОГО МОДУЛЯ ОТ НЕСАНКЦИАНИРОВАННОГО ДОСТУПА

Методические указания по выполнению практических работ для студентов укрупненной группы специальностей и направлений подготовки 10.00.00

Курск 2023

УДК 004.424

Составители: А.В. Митрофанов, А.А. Чеснокова

Рецензент

Кандидат технических наук, доцент кафедры «Информационная безопасность» А.Л. Марухленко

Программно-аппаратная защита исполняемого модуля от несанкционированного доступа: методические указания по выполнению практической работы по дисциплине «Технологии и методы программирования» / Юго-Зап. гос. ун-т; сост.: А.В. Митрофанов, А.А. Чеснокова. Курск, 2023. 11 с., Библиогр.: с. 11.

Содержат основные теоретические и практические сведения о программно-аппаратной защите исполняемого модуля от несанкционированного доступа. Указывается порядок выполнения практической работы, правила оформления и содержание отчета.

Методические указания по выполнению практических работ по дисциплине «Технологии и методы программирования», предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00

Текст печатается в авторской редакции

Подписано в печать _____. Формат 60×84 1/16.
Усл.печ.л. . Уч.-изд.л. . Тираж 50 экз. Заказ ____. Бесплатно
Юго–Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ

Цель практической работы – предложить программно-аппаратную защиту исполняемого модуля от несанкционированного доступа.

2. ЗАДАНИЕ

Ознакомиться с теоретическим материалом. Ознакомиться с примерами решения. Предложить программно-аппаратную защиту исполняемого модуля от несанкционированного доступа.

3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Получить задание.
2. Изучить теоретическую часть.
3. Предложить программно-аппаратную защиту исполняемого модуля от несанкционированного доступа.

4. СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист.
2. Краткая теория.
3. Предложения по программно-аппаратной защите исполняемого модуля от несанкционированного доступа.
4. Вывод.

5. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Программно-аппаратная защита используется для защиты программного обеспечения от несанкционированного (неавторизованного) доступа и нелегального использования. Защитный механизм программным образом опрашивает специальное устройство, используемое в качестве ключа, и работает только в его присутствии. Таким образом, механизм программно-аппаратной защиты содержит две составляющие:

1) аппаратное устройство (аппаратная часть);

2) программный модуль (программная часть).

Поэтому обычно говорят о системах программно-аппаратной защиты. Очевидно, что стоимость такого механизма превышает стоимость программной защиты, причем стоимость аппаратной части, как правило, превышает стоимость программной части. По этой причине программно-аппаратная защита считается привилегией корпоративных заказчиков, так как для индивидуального пользователя часто неприемлема с экономической точки зрения. Обратим внимание на то, что по существу программно-аппаратная защита не является защитой программ от нелегального распространения и использования. Не станет заказчик программы оплачивать дорогую аппаратуру только ради соблюдения авторских прав разработчика. Но если программный продукт снабжен модулем, предназначенным для защиты от несанкционированного доступа к данным и информации пользователя, то заказчик, как правило, готов платить за

аппаратуру, повышающую надежность такой защиты. Система защиты от несанкционированного доступа к данным реализована таким образом, что осуществляет проверку легальности пользователя при работе с программным обеспечением и тем самым косвенно препятствует и незаконному использованию программы. Кроме того, современные аппаратные устройства (ключи), помимо информации о законном пользователе, могут содержать также информацию о программном продукте. А системы программно-аппаратной защиты, кроме аутентификации пользователя, могут производить аутентификацию приложения. Поэтому системы программно-аппаратной защиты от несанкционированного доступа могут служить в то же время и для защиты авторских прав разработчиков программ. Системы программно-аппаратной защиты широко используются на практике и многими пользователями признаются надежным средством.

Защита от несанкционированного доступа Рассмотрим основные моменты защиты информации от несанкционированного доступа. Речь идет о таком порядке работы, при котором

1) доступ к информации имеет только тот пользователь, который имеет разрешение; будем называть такого пользователя законным;

2) каждый законный пользователь работает только со своей информацией и не имеет доступа к информации другого законного пользователя;

3) каждый законный пользователь может выполнять только те операции, которые ему разрешено выполнять.

Для организации такого порядка работы прежде всего необходимо обеспечить распознавание законного пользователя. Этот процесс часто называют авторизацией пользователя. Авторизация пользователя включает три этапа.

1. Идентификация пользователя.

2. Аутентификация пользователя.

3. Непосредственно авторизация пользователя.

Идентификация пользователя (identification) - это, с одной стороны, присвоение пользователю идентификатора - некоторого уникального признака (или нескольких); с другой стороны, процесс, во время которого пользователь указывает присвоенный ему идентификатор. Другими словами, идентификация - это процесс, при котором пользователь называет себя. Аутентификация пользователя (от англ. autentication - установление подлинности) - установление подлинности пользователя на основе сравнения с эталонным идентификатором. Авторизация пользователя - установление прав пользователя. Авторизованный пользователь (авторизованное лицо) - пользователь (лицо), который получил определенные права на работу с информацией. В процессе авторизации для законного пользователя определяются права пользователя, то есть определяются данные, с которыми ему разрешено работать; операции, которые ему разрешено выполнять и т.п

6. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

Правила связи программных модулей по информации

1. Информация зон глобальных переменных доступна дня использования любым модулем, входящим и комплекс программ или группу программ в соответствии с областью действия зоны глобальных переменных, т.е. глобальные переменные, могут быть доступны не для всего комплекса программ, а лишь для указанной в описании группы модулей.

2. Локальные переменные доступны лишь в пределах того модуля, в котором они определены или объявлены.

3. Для взаимодействия вызываемых и вызывающих модулей создаются зоны обменных переменных, информация из которых доступна лишь модулям, непосредственно связанным по управлению.

4. После окончания работы вызываемого модуля считается, что регистры не содержат информации, являющийся результатом его работы. Запрещается их использовать в вызывающем модуле.

5. Информация, находящаяся в регистрах вызывающего модуля, при вызове должна быть сохранена на период выполнения вызываемого модуля и восстановлена при возврате управления в вызывающий модуль. Сохранение регистров может осуществлять как вызывающий, так и вызываемый модуль.

Правила связи программных модулей по управлению

1. Передача управления вызываемому модулю всегда осуществляется через его начало, т.е. через первый оператор или команду.

2. Если необходимо исполнить модуль с некоторой внутренней точки, то вызов осуществляется стандартным образом (через первый оператор), а точка начала задается в виде параметра, при этом в начале вызываемого модуля должен стоять переключатель, который обеспечивает передачу управления к внутренним точкам входов по параметру, указанному при обращении.

3. Выход из вызываемого модуля всегда происходит через его естественное окончание, т.е. после нормального его завершения.

4. По окончании исполнения вызываемого модуля управление передастся в вызывающий модуль на оператор, следующий непосредственно за оператором вызова.

5. Модули низших уровней или одного уровня иерархии могут вызываться для исполнения только моделями высших уровней, т.е. модули низших уровней не могут вызывать модули высших уровней, а модули одного уровня - вызывать друг друга.

6. В каждом модуле должна быть предусмотрена возможность подключения контрольных и отладочных средств; операторы, реализующие эти средства, обычно сосредоточиваются в конце модуля.

7. СПИСОК ВОПРОСОВ

1. Что такое программный модуль?
2. Что такое защищенное программирование?
3. Что такое объектный модуль?
4. Что такое загрузочный модуль?
5. Правила связи программных модулей по управлению.
6. Правила связи программных модулей по информации.

8. БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Дж.Хьюз, Дж.Мичтом. Структурный подход к программированию. М.: Мир, 1980. - С. 29-71.

2. В.Турский. Методология программирования. - М.: Мир, 1981. - С. 90-164.

3. Е.А.Жоголев. Технологические основы модульного программирования//Программирование,1980, #2. - С. 44-49.

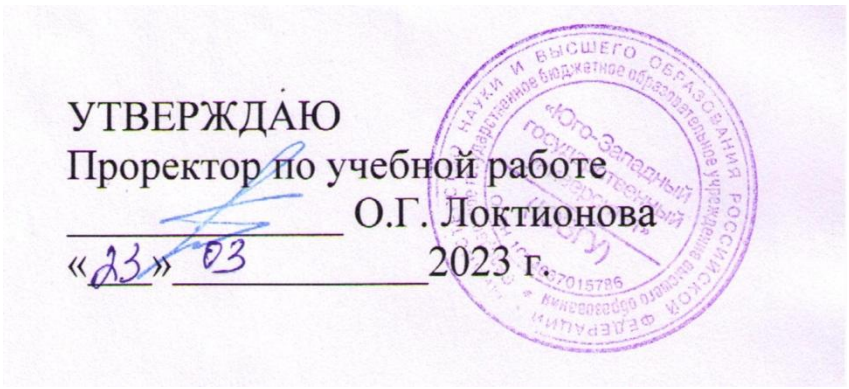
4. R.C.Holt. Structure of Computer Programs: A Survey//Proceedings of the IEEE, 1975, 63(6). - P. 879-893.

5. Г.Майерс. Надежность программного обеспечения. М.: Мир, 1980. - С. 92-113.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности



ЗАЩИТА СЕТЕЙ С ПРИМЕНЕНИЕМ МЕЖСЕТЕВЫХ ЭКРАНОВ

Методические указания по выполнению практических работ для студентов укрупненной группы специальностей и направлений подготовки 10.00.00

Курск 2023

УДК 004.725

Составитель: А.В. Митрофанов

Рецензент

Кандидат технических наук, доцент кафедры «Информационная безопасность» А.Л. Марухленко

Защита сетей с применением межсетевых экранов: методические указания к выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем» / Юго-Зап. гос. ун-т; сост.: А. В. Митрофанов. Курск, 2023. 14 с. Библиогр.: с. 14.

Указываются необходимые теоретические сведения, порядок выполнения практической работы, содержание отчета.

Методические указания по выполнению практических работ по дисциплине «Проектирование защищенных телекоммуникационных систем», предназначены для студентов укрупненной группы специальностей и направлений подготовки 10.00.00

Текст печатается в авторской редакции

Подписано в печать _____. Формат 60×84 1/16.
Усл.печ.л. . Уч.-изд.л. . Тираж 50 экз. Заказ ____. Бесплатно
Юго–Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ

Овладеть навыками работы с сетевой программой ATGuard

2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Межсетевой экран (firewall или брандмауэр) является программно-аппаратным средством осуществления сетевой политики безопасности в выделенном сегменте IP-сети.

В сфере компьютерных сетей межсетевой экран представляет собой барьер, защищающий от вторжения злоумышленников во внутреннюю локальную сеть для того, чтобы скопировать, изменить или стереть информацию либо воспользоваться памятью или вычислительной мощностью работающих в этой сети компьютеров. Межсетевой экран призван обеспечить безопасный доступ к внешней сети и ограничить доступ внешних пользователей к внутренней сети.

Название «брандмауэр», может относиться к одному устройству или одной программе. Термин «межсетевой экран» был принят для обозначения совокупности компонентов, которые находятся между вашей сетью и внешним миром и образуют защитный барьер.

Брандмауэр не может защитить от:

– вирусов. Хотя некоторые брандмауэры и способны распознавать вирусы в проходящем через них трафике, существует множество способов спрятать вирусы в программе. Если даже в описании вашего брандмауэра заявлена функция антивирусной проверки, не выключайте проверку вирусов на отдельных компьютерах в сети;

– «троянских коней». Как и в случае с вирусами, блокировать проникновение в сеть «троянских коней» (Trojan horses) достаточно сложно. Пользователь нередко поддается искушению загрузить программу из Internet или открыть прикрепленный к сообщению электронной почты файл, проложив тем самым путь в систему вредоносной программе;

– «социальной инженерии». Термин «social engineering» возник недавно и применяется для описания методов получения хакерами информации от доверчивых пользователей. Часто люди готовы сообщить свой пароль любому, кто позвонил по телефону и отрекомендовался представителем службы безопасности, что-нибудь «проверяющим». Межсетевой экран не в состоянии остановить невоздержного на язык сотрудника

– некомпетентности. Плохо подготовленные сотрудники или небрежное руководство приводят к ошибкам в настройках локальной сети и межсетевого экрана. Если сотрудники не понимают, как работает брандмауэр и как правильно его настраивать, не исключено, что это будет способствовать возникновению проблем;

– атаки изнутри. Межсетевой экран не может предотвратить злонамеренные действия внутри вашей сети. Это одна из причин, по которой безопасность компьютеров в сети остается важной проблемой и после установки брандмауэра.

3. ПОРЯДОК ВЫПОЛНЕНИЕ ПРАКТИЧЕСКОЙ РАБОТЫ

1) Установка и описание программы AtGuard

После инсталляции программы и перезагрузки компьютера Вы обнаружите в системном трее (system tray) иконку запущенного AtGuard’a , а вверху экрана его же панель (dashboard). Это означает, что инсталляция и первый запуск прошли успешно. Двойной щелчок на иконке открывает окно настроек.

Установка флажка Enable web filters включает блокирование, опции секретности и активные установки фильтров, определенные в диалоговом окне Web (HTTP) Filters. Уберите этот флажок, если Вы хотите выключить все web-фильтры.

Enable web filters действует как главный переключатель, который позволяет вам отменять индивидуальные установки фильтра в диалоговом окне

Web (HTTP) Filters и отключать всю фильтрацию веб-трафика. Когда Вы отключаете web-фильтры, программное обеспечение прекратит фильтровать любые HTTP данные, входящие или исходящие от рабочей станции. Если Вы устанавливаете или снимаете флажок Enable web filters, то эти изменения начинают действовать сразу.

Опции в этом диалоговом окне позволяют Вам включать или выключать индивидуальные web-фильтры, cookie и Java/ActiveX мониторы. Вы можете также модифицировать список портов, которые AtGuard контролирует для HTTP связи, когда web-фильтры включены. Установите флажки Ad Blocking, Privacy, Active Content и Cookie Assistant в этом диалоговом окне - этого будет достаточно. Флажок Java/ActiveX Assistant можете не устанавливать, иначе AtGuard будет каждый раз задавать ненужные вопросы.

HTTP Port List - сетевые сервисы (типа HTTP или FTP) используют специфические порты на вашем компьютере. Например, HTTP-связь обычно проводится через порт 80. Web-фильтры AtGuard’a контролируют весь HTTP-трафик, посланный и полученный через порты, которые указаны в Port List, применяя блокирование, секретность и другие опции, которые Вы определили. Ваша рабочая станция может соединяться с Интернетом посредством proxy-сервера, при этом весь HTTP-трафик проходит через порт, используемый этим proxy-сервером. Или Вы можете использовать приложение, которое инициирует HTTP-связь через нестандартный порт. Если HTTP-трафик идет через нестандартный порт, вы должны добавить номер этого порта в Port List. Изменения в настройках фильтров вступают в действие сразу после нажатия кнопки "Применить".

Add Site - нажмите эту кнопку, чтобы открыть диалоговое окно New Site/Domain, которое используется для добавления нового сайта или домена к иерархическому списку сайтов в левом окне. Напечатайте имя web-сайта или имя домена и нажмите OK. После добавления сайта Вы можете выбрать его в списке. Используйте установки Ad Blocking, Privacy, Active Content, чтобы

определить правила и набор блокировок, которые AtGuard использует только когда Вы посещаете конкретный web-сайт.

Ad Blocking - эти установки позволяют поддерживать блокирующий список по умолчанию и специфические для конкретных сайтов блокирующие списки, которые используются, чтобы указать, чего не нужно отображать на веб-страницах. Когда блокирующий фильтр включен, все HTML-страницы просматриваются на предмет наличия HTML-строк, специфичных для конкретного сайта, указанного в списке для блокирования, плюс значения по умолчанию, определенные для всех сайтов. Любой HTML-код, который содержит разрешенную к блокированию строку, будет удален из web-страницы AtGuard'ом прежде, чем эта страница будет интерпретирована и показана браузером.

Privacy - установки секретности позволяют определять правила, управляющие тем, как Ваш браузер обрабатывает запросы о различных типах информации, сделанных сайтами, которые Вы посещаете.

Cookies - это информация, которую web-серверы сохраняют на вашем компьютере для более позднего использования. Web-серверы могут читать cookies, чтобы следить, сколько раз вы их посетили, когда и какую информацию вы просматривали. Они могут даже использовать cookies, чтобы передать эту информацию другим web-серверам, типа серверов рекламы. Положительная сторона cookies в том, что они могут использоваться, чтобы сохранить вашу собственную конфигурацию web-сайта, запоминать, что вы поместили в вашу "покупательскую корзину" в интерактивном магазине или сохранять имя пользователя и пароль для сайтов подписки. Чтобы обеспечить максимальную секретность, разрешите использование cookies только проверенным сайтам, которым вы доверяете.

Referer - позволяет вам определить, узнают ли третьи сайты о том, из какого места поступил запрос данных с этих серверов.

Refer field - эти поля используются, чтобы обеспечить "третьи" сайты информацией относительно сайта, с которого поступил запрос данных из их

сервера. Refer поля позволяют веб-серверам знать, где вы только что были. Вполне возможно, что вы не захотите, чтобы эта информация становилась известной. Иногда это опасно. Например, некоторые онлайновые почтовые службы подставляют пароль просто в сетевой путь, который отображается в браузере. Если вам пришло письмо, содержащее ссылку на какой-нибудь сайт, и вы последовали по ссылке прямо из web-mail, то в статистике сайта на, который вы пришли, будет зафиксирован адрес страницы, содержавшей ссылку на него - refferer. А этот самый refferer может содержать ваш логин и пароль к вашему почтовому ящику. Некоторые сайты не позволяют заходить на них с включенным Block refer fields.

Browser (User-agent) - позволяет определить, обеспечиваются ли сайты информацией, какой браузер вы используете.

E-mail (From) - позволяет определить, получат ли сайты адрес электронной почты, который использует ваш браузер, чтобы идентифицировать вас как отправителя почты.

Active Content - эти установки позволяют Вам предотвращать выполнение следующих типов программ: JavaScript, Java applets, ActiveX controls. Кроме того, можно установить, чтобы анимационные изображения проигрывались только один раз. Когда блокирование активного содержимого включено, все HTML страницы просматриваются, и любой HTML-код, который активизирует нежелательное содержание, будет удален из страницы AtGuard'ом прежде, чем страница интерпретируется и отобразится веб-браузером.

Установки файрвола определяют, должен ли AtGuard запретить или разрешить приложениям на вашем компьютере посылать или получать информацию по TCP/IP. Для этого имеется список правил, которые описывают, какие типы сетевой активности разрешаются, и через какие сервисы приложения могут связываться. Вы можете добавлять, изменять, или удалять правила.

Включите опции Enable firewall, Enable RuleAssistant (interactive learning mode)

Для временного отключения какого-либо правила уберите флажок напротив соответствующей строки в списке правил.

Если правило "Блокировать" (Block) или "Разрешить" (Permit) указано, все оставшиеся правила игнорируются. Другими словами если вы, например, закрыли порт номер N, а ниже прописано правило, разрешающее использование этого порта, то оно будет проигнорирована и соединение по этому порту будет закрыто.

Если правило "Игнорировать" (Ignore) указано, тип связи, которая была предпринята, регистрируется в лог-файле firewall'a и затем обработка продолжается, пока не произойдет какого-либо другого соответствия. Если не найдется никакого правила, связь или блокируется (по умолчанию) или вызывается RuleAssistant. Чтобы перемещать правило по списку, выделите соответствующую строку и затем используйте кнопки "стрелка вверх" или "стрелка вниз" для помещения правила в соответствующую позицию.

Любое TCP/IP соединение, для которого нет firewall правила, блокируется по умолчанию. Если Вы хотите выборочно блокировать или разрешать соединение, для которого нет правила, установите флажок Enable RuleAssistant (интерактивный режим изучения).

Если флажок RuleAssistant включен, вам будет автоматически задан вопрос запретить (Block) или разрешить (Permit) соединение всякий раз, когда приложение на вашей рабочей станции или какое-то приложение извне делает попытку установить связь, для которой не описано никаких правил в firewall. В результате вашего решения AtGuard разрешает или блокирует сетевую связь и может создавать правило firewall, которое применяется в дальнейшем для данного типа сетевого соединения.

Direction. Inbound связь включает пакеты, посланные вашему компьютеру. Outbound связь включает пакеты, посланные вашим компьютером. Either - связь в любом направлении.

Protocol. Определяет, к какому протоколу связи применяется правило: TCP, UDP, или TCP и UDP, ICMP...

TCP - стандартный протокол Интернета транспортного уровня, обеспечивает надежную полнодуплексную связь. Программное обеспечение, реализующее протокол TCP, обычно постоянно находится в операционной системе и использует IP протокол, чтобы передать информацию. Примеры TCP приложений и сервисов - FTP, web-браузер, email и IRC.

UDP - транспортый уровень в TCP/IP сетях. UDP - низкоуровневый протокол, который использует IP, чтобы доставить пакеты. Примеры сервисов и приложений, которые используют UDP - DNS, NetBIOS.

ICMP - протокол межсетевых управляющих сообщений.

Application. Эта опция позволяет определять, применяется ли правило к конкретному приложению или к любому приложению, которое делает попытку сетевой связи, определенной правилом.

Service. Позволяет определять, применяется ли правило к локальным или удаленным сервисам и применяется ли это к одиночному определенному сервису или к любому сервису, который делает попытку сетевой связи, определенной правилом.

Time Active. Используйте эти установки, чтобы определить время когда, правило будет действовать.

Logging. Определяет, что событие регистрируется в лог-файле, когда устанавливается описанное правилом соединение.

Show taskbar icon - при запущенном AtGuard показывать его иконку в панели задач.

Show dashboard window - при запущенном AtGuard показывать dashboard.

Enable password protection - если выбрано, то как только вы попытаетесь открыть диалоговое окно AtGuard Settings, окно Dashboard Properties, Event Log, или окно статистики, вы будете должны ввести пароль.

StartUp Options / Run at network startup. Когда эта опция выбрана, AtGuard запускается автоматически, если вы открываете сетевое соединение, и останавливается также автоматически, когда вы закрываете ваше сетевое соединение.

Сразу после инсталляции, зайдите в настройки Firewall. Снимите флажки Default Inbound ICMP, Default Inbound DNS, Default Inbound Bootp, Default Inbound NetBIOS

Для дальнейшей настройки Inbound DNS необходимо узнать DNS адрес вашего провайдера. Затем Settings -> Firewall -> Add. В поле "Name" впишите DNS, поля "Action" и "Directon" изменять не нужно. Поле "Protocol" установите "TCP or UDP 3". Здесь же нажмите закладку "Service" и установите "Remote service" в "Single service", в появившееся поле впишите 53 (номер порта домена). Теперь выберите закладку "Address" и поставьте "Remote address" = "Host address" и в появившееся поле впишите адрес DNS вашего провайдера, нажмите OK. Теперь повторите эту же операцию только измените "Directon" на "Inbound". Повторите то же самое и для остальных адресов DNS, если они есть. Все, настройка DNS закончена.

Как можно защититься от рекламных баннеров. Правой кнопкой мыши нажимаем по баннеру. В всплывающем меню выбираем "Копировать ярлык". Вызываем AtGuard Settings -> Web. Выбираем в списке (Defaults), Ad Blocking -> Add. В появившемся окошке нажимаем правой кнопкой мыши, выбираем «вставить». Например, для linkexchange будет такая строка http://www.linkexchange.ru/users/091164/goto.map Ее нужно отредактировать, чтобы получилось linkexchange.ru/users/ ибо удаленная часть может изменяться от сайта к сайту. Всё. Еще пример. http://www.reklama.ru/cgi-bin/href/myclub?3353573 После правки: reklama.ru/cgi-bin/ .

Есть еще один более простой способ. Если у вас запущен Dashboard, то в правом углу будет Trashcan ("Мусорная корзина") AtGuard'a. Чтобы переместить рекламу в мусорку при использовании MSIE 4.0, выберите рисунок и мышкой перетащите его в Trashcan. При использовании Netscape или MSIE

3.0, щелкните правой кнопкой мыши на баннере. Чтобы заблокировать все подобные ссылки, выберите пункт Copy link location (если картинка грузится с того же сервера, что и страница). Если баннер грузится с сервера рекламодателя (например, это могут быть баннеры сетей reklama.ru, linkexchange), то выберите пункт Copy image location. Затем щелкните правой кнопкой мыши на иконке Trashcan и выберите пункт Paste (Вставка) из всплывающем меню.

Как говорилось выше, нашу задачу сильно облегчает то, что вся реклама объединяется или уже объединена в рекламные (баннерные) сети. Поэтому закрыв для себя AtGuard'ом один рекламный сайт, вы избавитесь от сотен и сотен рекламных баннеров. Это сохранит вам деньги, нервы и высокую скорость соединения.

Обязательно установите все флажки в окне AtGuard Settings -> Web -> Active Content.

В окне AtGuard Settings -> Firewall включите Enable Rule-Assistant для интерактивного обучения вашего стража. Если в процессе вам встретится непонятное на первый взгляд сообщение о каком-либо соединении, запретите его, потом всегда можно посмотреть в лог-файлах.

2) Защита от атак WinNuke

Чтобы защититься от атаки WinNuke, нужно поставить соответствующий фильтр. Атака WinNuke заключается в посылке OOB-данных на 139 порт. Таким образом, достаточно будет заблокировать TCP-соединения с 139 портом. Однако 139 порт используется для NetBIOS и потому при работе в локальной сети его перекрывать не следует. Но если вы заходите в Сеть с домашнего компьютера, то блокируйте смело.

В настройке Firewall добавляем новое правило – “Add”. Назовем “WinNuke”, действие – “Block”, направление только входящие – “Inbound”, протокол “TCP”. Далее на закладках: Any Application. Service: remote - "Any", local – single service 139 порт. Остальные настройки можно оставить по умолчанию. Включите протоколирование, чтобы можно было видеть, что вы подверглись атаке. По аналогии можно настроить и другие фильтры.

3) Задание на практическую работу

– Изучить функции программы, пользуясь описанием программы.

– Установить 2 виртуальные машины. Настроить локальную сеть между двумя виртуальными машинами, если требуется.

– Установить обе виртуальные машины AtGuard.

– Осуществить обмен пакетами запрещенного типа при включенном и при выключенном AtGuard с другим компьютером сети.

– Создать правило запрещающее получение доступа к компьютеру с удаленного компьютера (с конкретного IP-адреса или с определенного имени компьютера), попытаться обратиться с запрещенного компьютера и отследить реакцию AtGuard.

– Составить отчет о проделанной работе.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) Шувалов В.П., Величко В.В., Субботин Е.А., Ярославцев А.Ф. Телекоммуникационные системы и сети. Том 3. Мультисервисные сети (2005)

2) Петраков А.В. Основы практической защиты информации. 2-е изд. Учебн. пособие. – М.: Радио и связь. 2000. – 368 с.

3) Цифровые и аналоговые системы передачи: Учебник для вузов/ В.И.Иванов, В.Н.Гордиенко, Г.Н.Попов и др.; Под ред. В.И.Иванова. – 2-е изд. – М.: Горячая линия – Телеком, 2003. – 232 с.

4) Гольдштейн Б.С., Соколов Н.А., Яновский Г.Г. Сети связи: Учебник для ВУЗов. - СПб.: БХВ-Петербург, 2010. - 400 с.

5) Башарин Г.П. Лекции по математической теории телетрафика: Учеб. пособие. Изд. 3-е, испр. и доп. - М.: РУДН, 2009. - 342 с.

6) Буч Г. Объектно-ориентированный анализ и проектирование. – М.: Вильямс, 2008.

7) Леоненков А.В. Самоучитель языка UML. – СПб.: БХВ-Петербург, 2004.

8) Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов. – М.: ДМК Пресс, 2002.

9) А.В. Росляков. Виртуальные частные сети. Основы построения и применения. - М.: Эко-Трендз, 2006. - 242 с.

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра «Информационная безопасность»

СЕТЕВЫЕ ФИЛЬТРЫ

Методические указания по выполнению лабораторных и практических работ для студентов укрупненной группы специальностей 10.00.00.

Курск 2019

УДК 004
Составители: И.В. Калуцкий, К.Г. Верютина

Рецензент
Кандидат технических наук, доцент кафедры
информационной безопасности *А.Г. Спеваков*

Сетевые фильтры: методические указания по выполнению лабораторных и практических работ для студентов укрупненной группы специальностей 10.00.00 / Юго-Зап. гос. ун-т; сост.: И.В. Калуцкий, К.Г. Верютина, Курск, 2019. 21 с.: ил. 21, табл. 1. Библиогр.: с. 21.

Содержат сведения по вопросам настройки и управления фильтрами для защищенных и открытых сетей, а также основные правила фильтрации в ViPNet4. Указывается порядок выполнения лабораторной работы, правила оформления, содержание отчета.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям и направлениям подготовки «Информационная безопасность телекоммуникационных систем», «Информационная безопасность».

Предназначены для укрупненной группы специальностей 10.00.00.

Текст печатается в авторской редакции

Подписано в печать 21.01.19. Формат 60х84 1/16.
Усл. печ. л. 1,22. Уч. –изд. л. 1,1. Тираж 50 экз. Заказ 27.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Нельзя не сказать об угрозах защищаемому компьютеру из сети Internet: как входящие на компьютер данные могут нанести вред ему, так и выходящая за предел информация может потерять свою ценность. Однако можно найти решение ряд проблем с безопасностью в Интернете, или, по крайней мере, сделать их менее опасными, если использовать существующие и хорошо известные технологии и меры защиты на уровне хостов. Брандмауэр может значительно повысить уровень безопасности сети организации и сохранить в то же время доступ ко всем ресурсам Интернете.

Несмотря на то, что VPN сети считаются более защищенными по сравнению с обычной сетью, в них так же существует проблема контролирования входящих и исходящих данных. Если решением для обычно сети будет брандмауэр, то каким решением будет для виртуальной сети?

Для виртуальной сети ViPNet такое решение есть – сетевой фильтр. Сетевой фильтр позволяет контролировать данные, циркулирующие в сети, а также блокировать данные с неизвестных источников. Так же в сетевых фильтрах ViPNet возможна гибкая настройка, что позволяет создавать фильтры для любых целей, включая блокирование данных с определенного порта или работу сетевого фильтра по расписанию.

Однако, для того, чтобы настраивать сетевые фильтры необходимо иметь представления о правилах фильтрации в сети ViPNet. Именно об этом и пойдет речь в лабораторной работе.

ЦЕЛЬ РАБОТЫ

Цель лабораторной работы – усвоить правила фильтрации и способы настройки сетевых фильтров, которые в дальнейшем позволить защитить сеть ViPNet от сетевых атак и несанкционированного доступа.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Получить задание
2. Изучить теоретическую часть
3. Описать со скриншотами предметную область
4. Написать вывод

СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист
2. Задание в соответствии с вариантом
3. Описание предметной области со скриншотами
4. Вывод

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Фильтрации подвергается весь трафик, который проходит через сетевой узел:

- открытый (нешифрованный) трафик;
- защищенный (зашифрованный) трафик;
- туннелируемый трафик.

Рис. 1 – Виды IP-трафика

Наибольшую опасность представляет трафик из открытой сети, поскольку в случае атаки достаточно сложно обнаружить ее источник и принять оперативные меры по ее пресечению.

И открытый, и защищенный трафик может быть локальным или широковещательным. Под локальным трафиком понимается входящий или исходящий трафик конкретного узла (то есть когда сетевой узел является отправителем или получателем IP-пакетов). Под широковещательным трафиком имеется в виду передача узлом IP-пакетов, у которых IP-адрес или MAC-адрес назначения является широковещательным адресом (то есть передача пакетов всем узлам определенного сегмента сети).

Кроме этого, через Координатор может проходить транзитный трафик. Координатор не является ни отправителем, ни получателем транзитных IP-пакетов, которые следуют через Координатор на другие узлы.

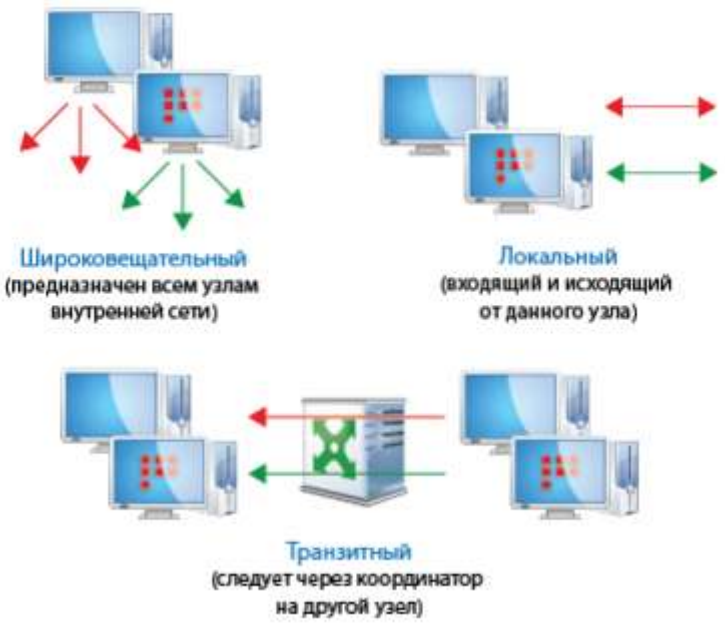


Рис. 2 – Виды защищенного и открытого трафика

С помощью фильтров для открытой сети на защищенном узле можно разрешить либо запретить обмен IP-пакетами с открытыми узлами, то есть с узлами, на которых не установлено программное обеспечение ViPNet с функцией шифрования трафика.

С помощью фильтров защищенной сети можно ограничить обмен IP-трафиком с защищенными узлами ViPNet, с которыми данный узел имеет связь. Фильтры для туннелируемого трафика определяют правила для IP-пакетов, передаваемых между туннелируемыми узлами и узлами сети ViPNet, с которыми данный координатор имеет связь.

ВЫПОЛНЕНИЕ РАБОТЫ

Фильтры открытой сети

Сначала необходимо задать IP-адреса: для узла *Главный администратор* – 192.168.4.1/24 (VM1), для открытого узла – 192.168.4.2/24 (VM2).

Убедимся, что сейчас команда ping между этими двумя узлами выполняется.

Для этого перейдите на открытый узел, вызовите командную строку и выполните команду *ping 192.168.4.1*. В результате чего будет успешно отправлено 4 пакета на узел *Главный администратор.*

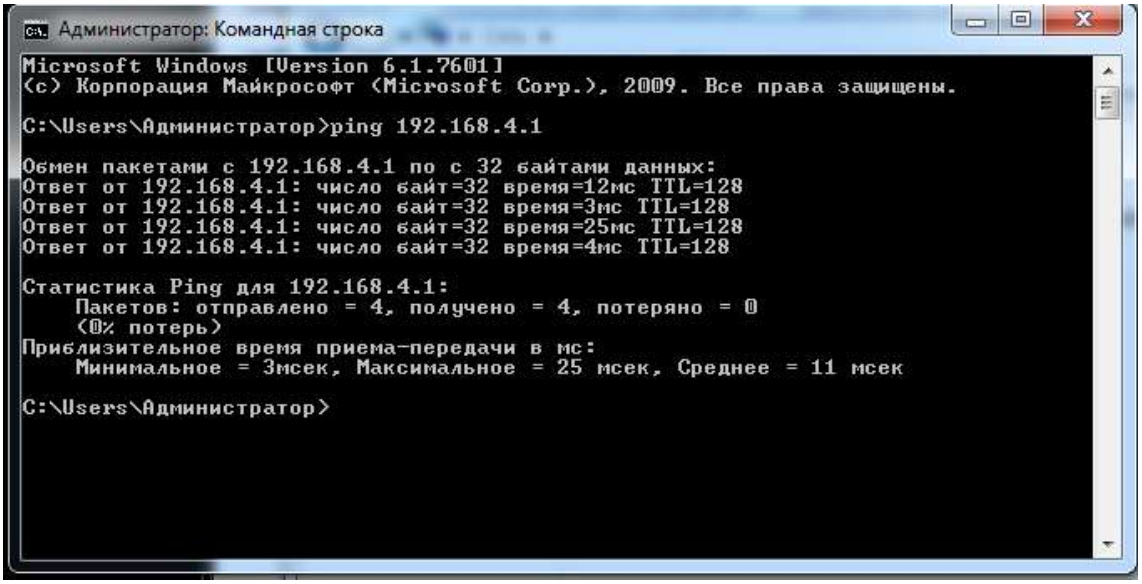


Рис. 3 – Выполнение команды *ping* на открытом узле

Для запрета доступа по протоколу ICMP (ICMP отвечает за обмен сообщениями и за выполнение команды *ping*) с открытого узла на узел *Главный администратор* выполните следующие действия в окне программы ViPNet Монитор на узле *Главный администратор*:

1. В окне программы ViPNet Монитор на панели навигации выберите раздел *Сетевые фильтры > Фильтры открытой сети*.

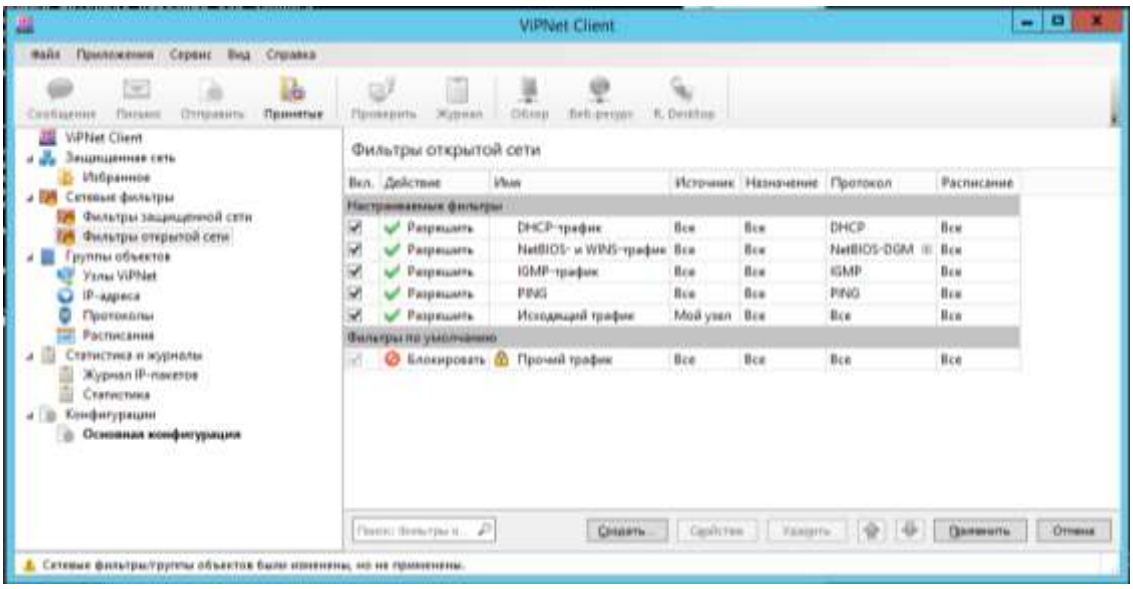


Рис. 4 – Фильтры открытой сети

2. На панели просмотра нажмите кнопку *Создать*, после чего в появившемся окне задайте параметры нового фильтра.

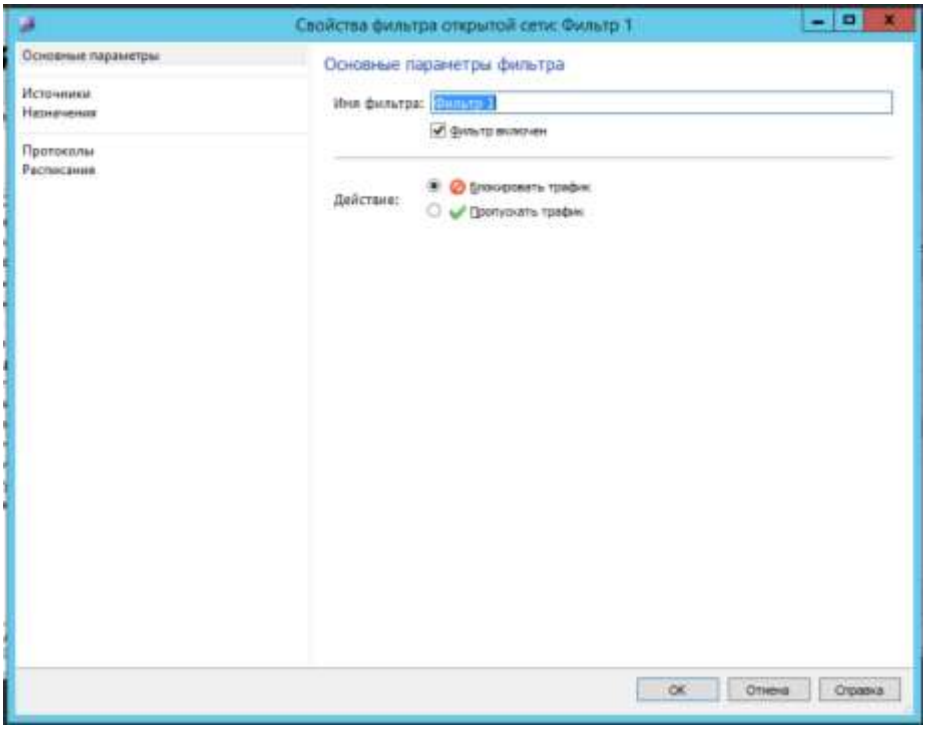


Рис. 5 – Новый фильтр открытой сети

3. В разделе *Основные параметры* укажите имя фильтра и его действие: *блокировать*. В разделе *Источники* задайте IP-адрес открытого узла – 192.168.4.2.

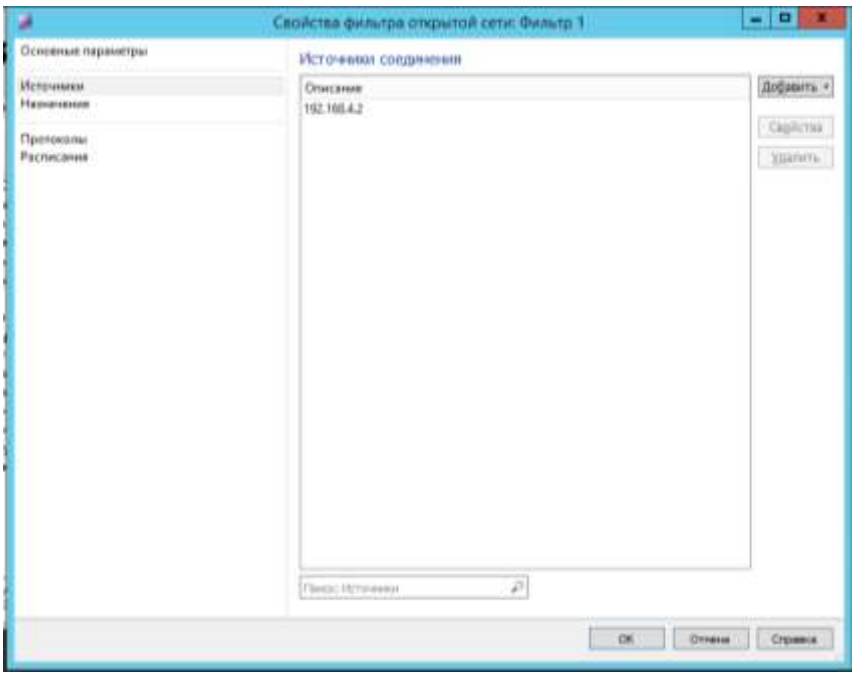


Рис. 6 – Задание источника нового фильтра

4. В разделе *Назначения* выберите узел *Мой узел*.

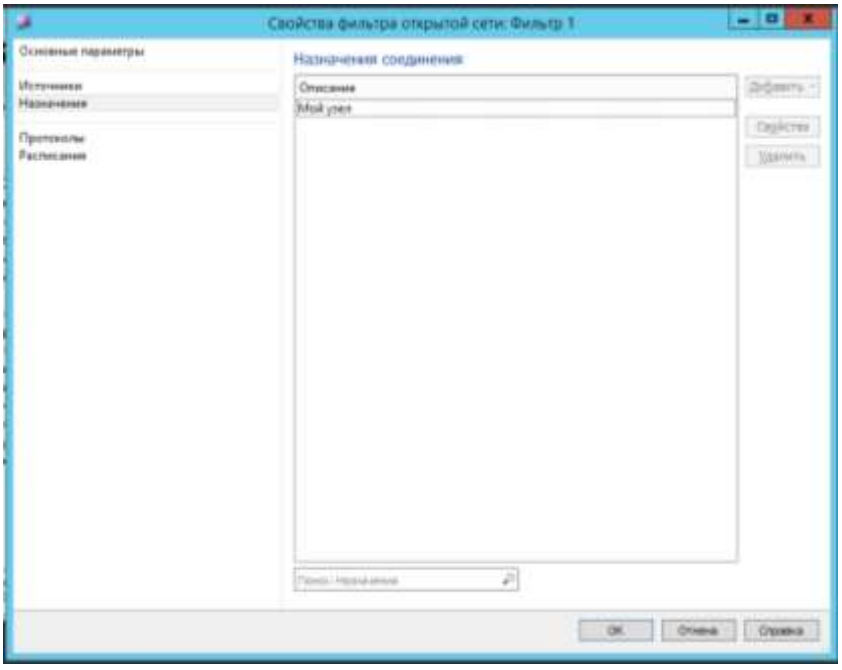


Рис. 7 – Задание назначения нового фильтра

5. В разделе *Протоколы* укажите *протокол ICMP*, для этого нажмите кнопку *Добавить* > *IP-протокол* > *ICMP*.

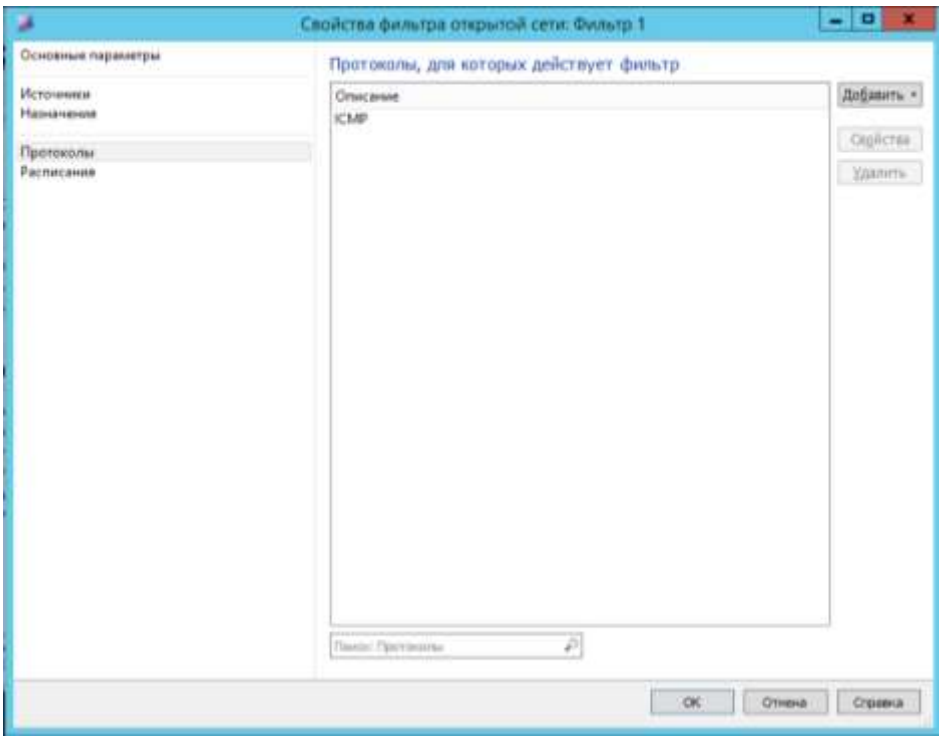


Рис. 8 – Добавление IP-протокола ICMP

6. В разделе *Расписания* в случае необходимости можно указать дату и время работы данного фильтра.

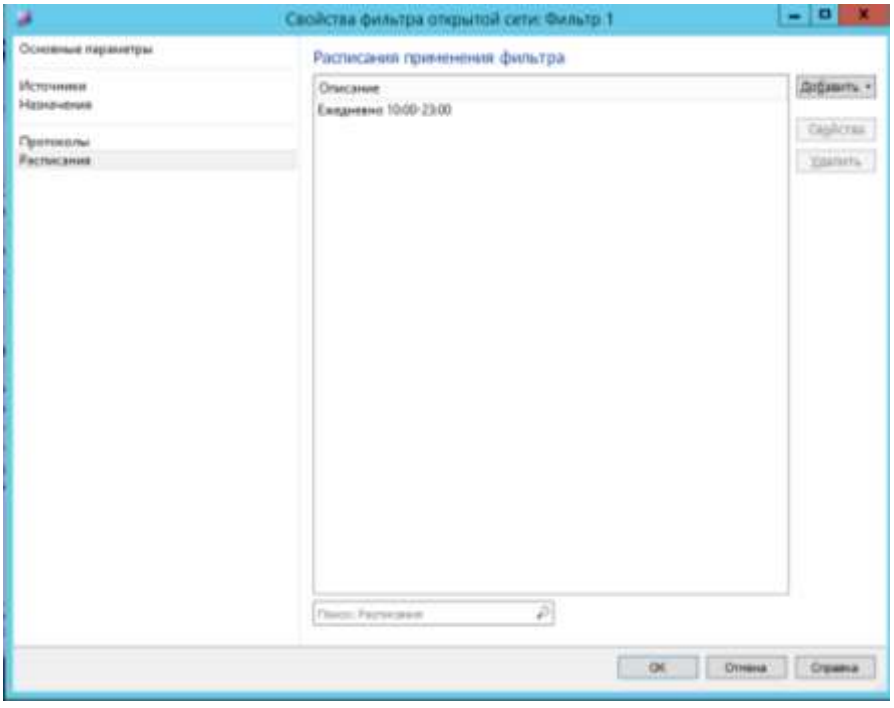


Рис. 9 – Добавление расписания применения фильтра

7. Нажмите кнопку *ОК*. В результате в списке на панели просмотра появится новый фильтр.

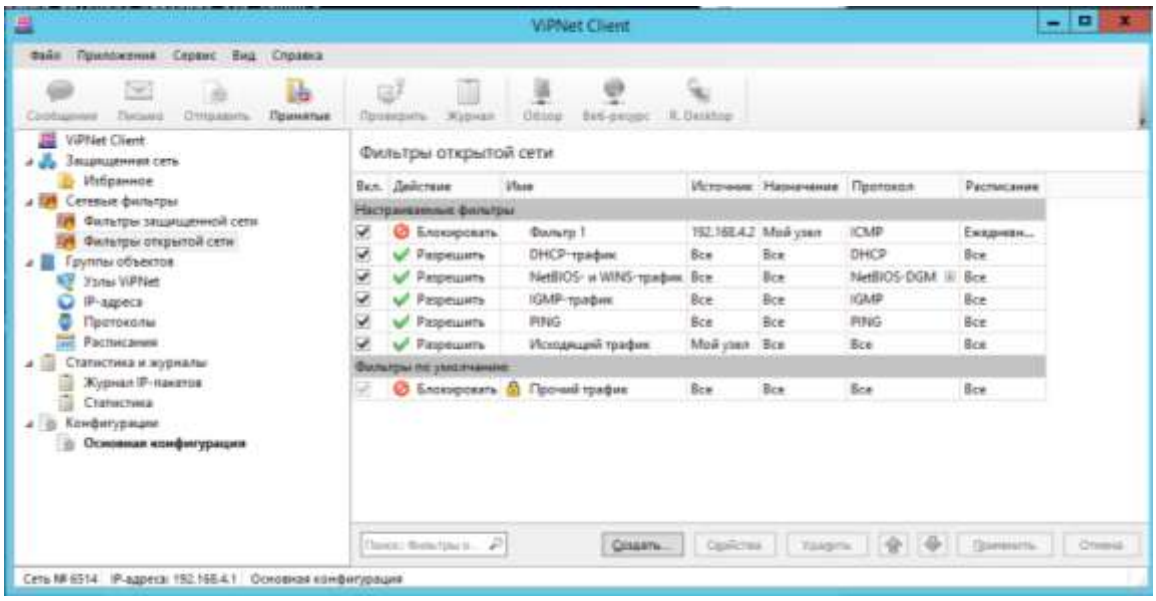


Рис. 10 – Фильтры открытой сети после добавления нового фильтра

8. Проверьте появление нового фильтра в *разделе Сетевые фильтры > Фильтры открытой сети* и нажмите кнопку *Применить*. В случае, если кнопка *Применить* не будет нажата, новый фильтр не будет использоваться при работе программы.

9. Перейдите на открытый узел, вызовите командную строку и выполните команду *ping 192.168.4.1*. В результате чего пакеты на узел *Главный администратор* отправляться не будут.

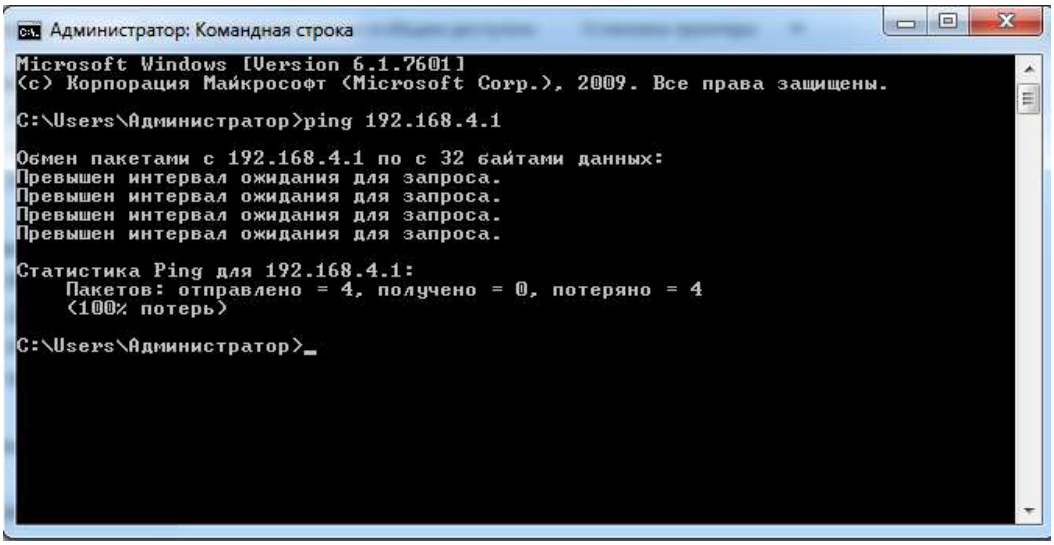


Рис. 11 – Выполнение команды *ping* на открытом узле после применения фильтра

Фильтры защищенной сети

Сначала необходимо включить удаленное подключение к рабочему столу в настройках системы. Также необходимо задать IP-адреса: для узла *Главный администратор* – 192.168.4.1/24, для узла *Помощник глав админа* – 192.168.4.2/24.

Убедимся, что сейчас подключение к удаленному рабочему столу *Помощника глав админа* с узла *Главного администратора* работает.

Для этого выберите узел *Помощник глав админа* в окне *Защищенная сеть* на узле *Главный администратор* и подключитесь к нему с помощью удаленного рабочего стола, после чего появится окно авторизации.

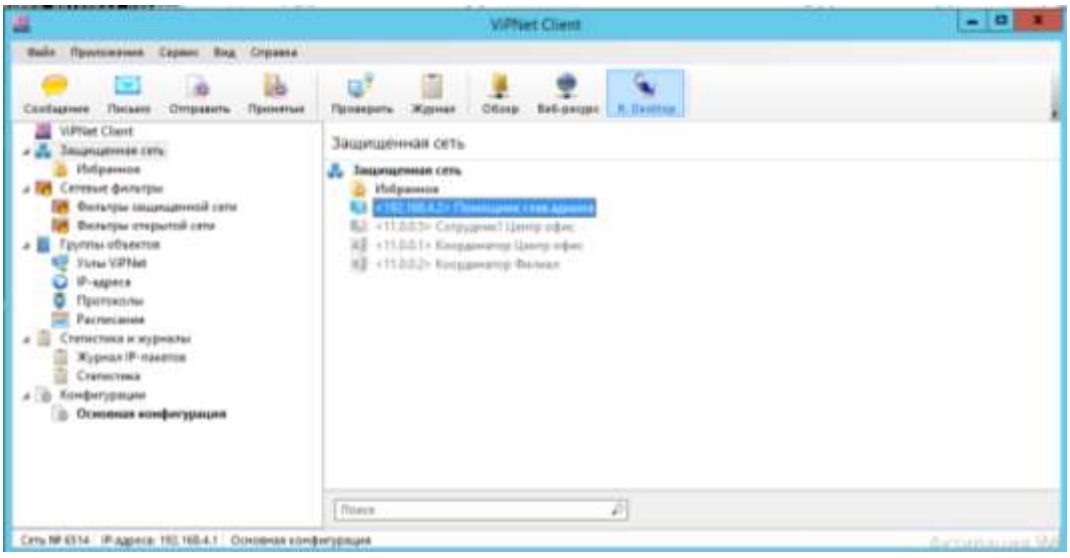


Рис. 12 – Подключение к удаленному рабочему столу

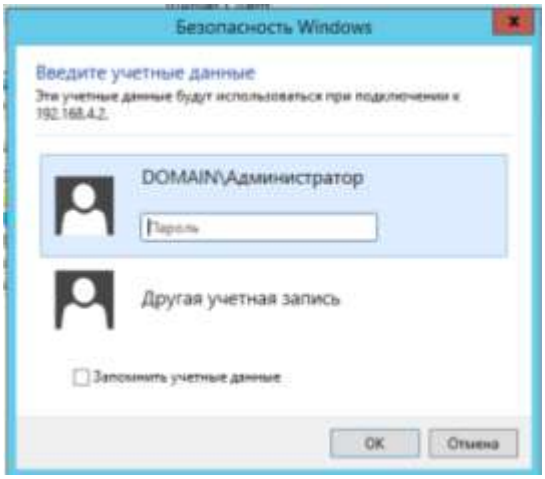


Рис. 13 – Окно авторизации

Для запрета доступа по протоколу RDP к узлу *Помощник глав admина* выполните следующие действия в окне программы ViPNet Монитор на узле *Главный администратор*:

1. В окне программы ViPNet Монитор на панели навигации выберите раздел *Сетевые фильтры* > *Фильтры защищенной сети*.

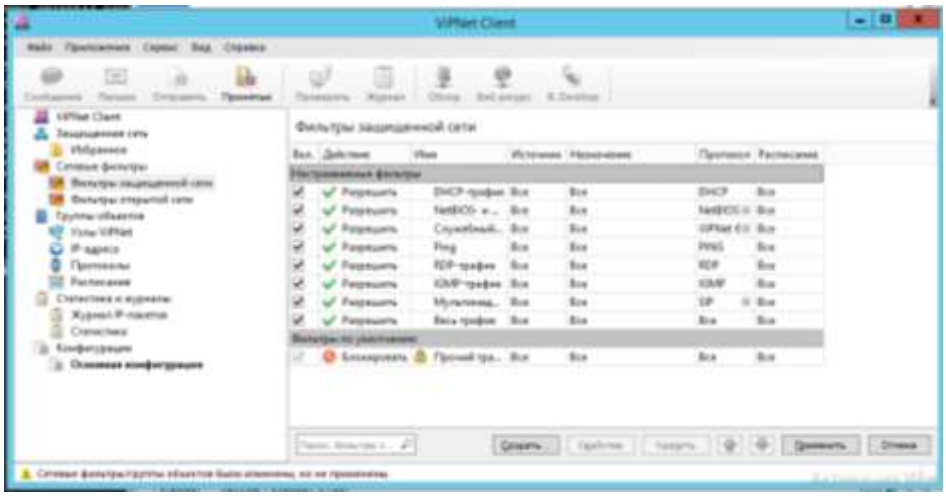


Рис. 14 – Фильтры защищенной сети

2. На панели просмотра нажмите кнопку *Создать*, после чего в появившемся окне задайте параметры нового фильтра.

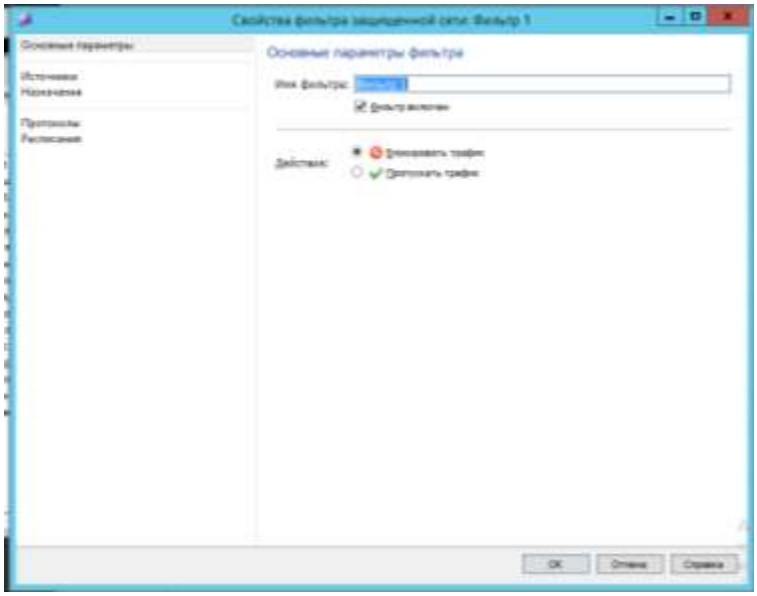


Рис. 15 – Новый фильтр защищенной сети

3. В разделе *Основные параметры* укажите имя фильтра и его действие: *блокировать*. В разделе *Источники* выберите *Мой узел*.

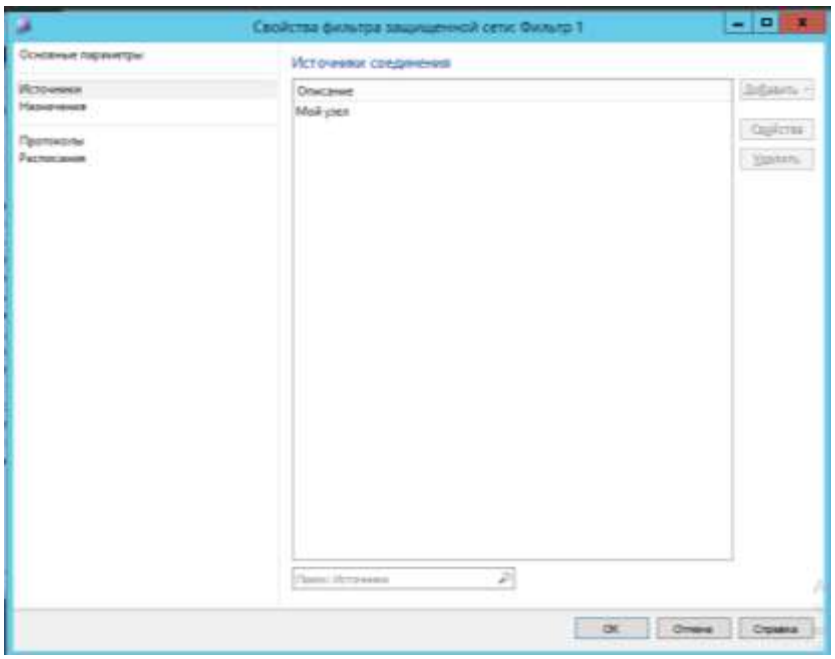


Рис. 16 – Задание источника нового фильтра

4. В разделе *Назначения* выберите узел *Помощник глав admina*.

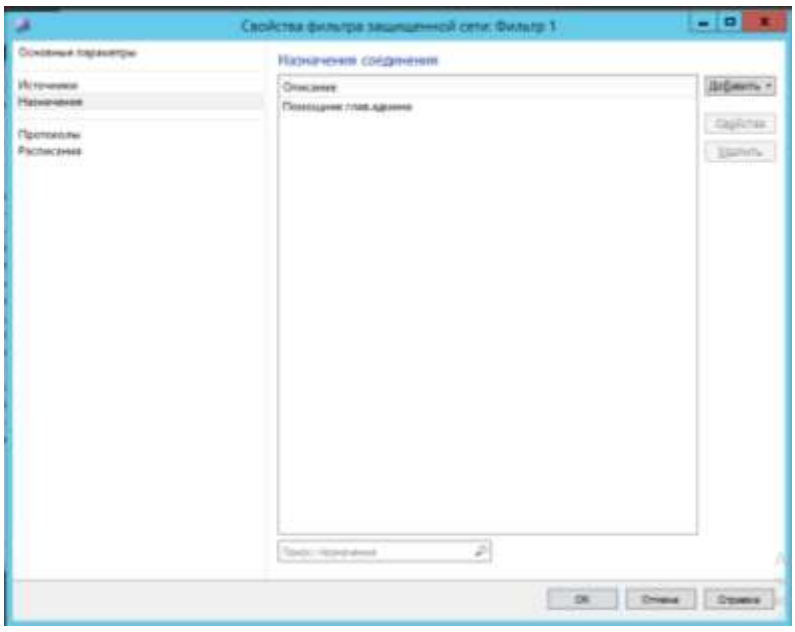


Рис. 17 – Задание назначения нового фильтра

5. В разделе *Протоколы* укажите *протокол RDP*, для этого нажмите кнопку *Добавить* > *Группы протоколов* > *RDP*.

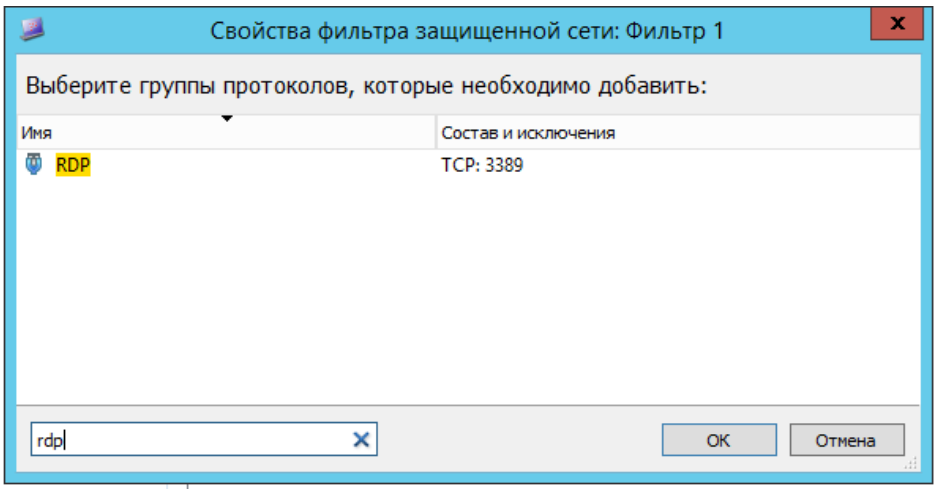


Рис. 18 – Добавление группы протоколов RDP

6. В разделе *Расписания* в случае необходимости можно указать дату и время работы данного фильтра.

7. Нажмите кнопку *ОК*. В результате в списке на панели просмотра появится новый фильтр.

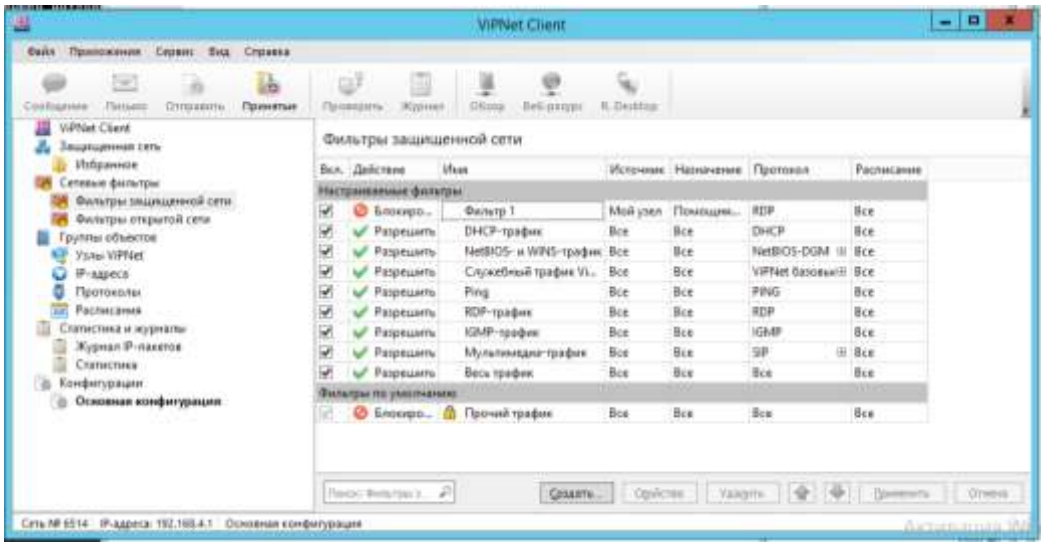


Рис. 19 – Фильтры защищенной сети после добавления нового фильтра

8. Проверьте появление нового фильтра в *разделе Сетевые фильтры > Фильтры защищенной сети* и нажмите кнопку *Применить*. В случае, если кнопка *Применить* не будет нажата, новый фильтр не будет использоваться при работе программы.

9. Выберите узел *Помощник глав адмиа* в окне *Защищенная сеть* и подключитесь к нему с помощью удаленного рабочего стола, после чего окно авторизации на удаленном узле не должно открыться.

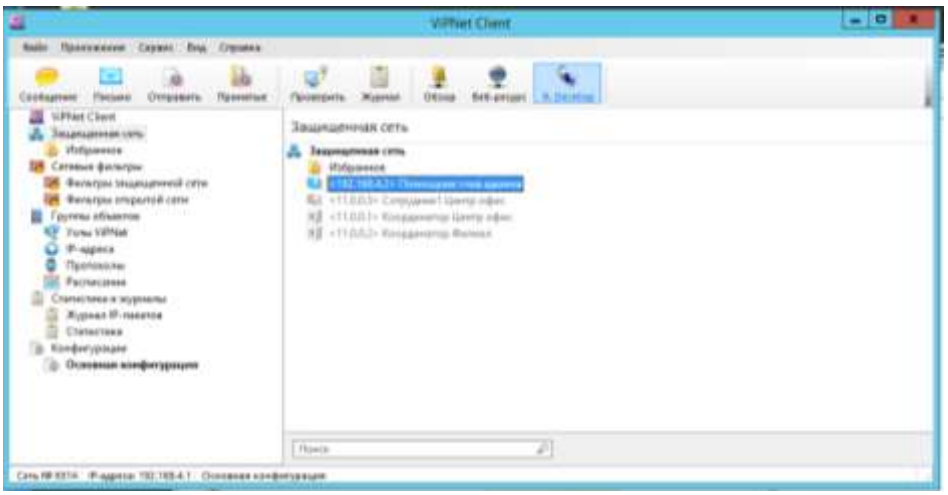


Рис. 20 – Подключение к удаленному рабочему столу

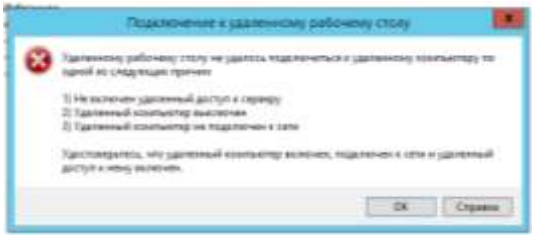


Рис. 21 – Ошибка при подключении к удаленному рабочему столу

Проделав данное действие, мы увидели, что настроили фильтр правильно и окно авторизации на удаленном узле не открылось.

ВАРИАНТЫ ЗАДАНИЙ

При выполнении лабораторной работы необходимо выполнить 1 и 2 задании в соответствии с вариантом, выданным преподавателем (таблица 1). Проверить работу фильтров.

Таблица 1 – Варианты заданий

№	1 задание	2 задание
1	Создать фильтр для защищенной сети. Фильтр должен блокировать исходящие соединения по протоколу TELNET. Настроить время работы с 12 до 18	Создать фильтр протоколов для открытой сети. Фильтр должен блокировать любые соединения протокола ICMP. Настроить время работы с 9 до 18
2	Создать фильтр для открытой сети. Фильтр должен блокировать любые соединения протокола TCP. Настроить время работы фильтра с 8 до 18	Создать фильтр для защищенной сети. Фильтр должен блокировать входящие соединения по протоколу SSH. Настроить время работы с 10 до 15
3	Создать фильтр для защищенной сети. Добавить фильтр протоколов для блокирования входящего соединения протокола TCP и настроить время работы с 13 до 14	Создать фильтр протоколов для открытой сети. Фильтр должен блокировать любые соединения протокола ICMP. Настроить время работы с 9 до 18
4	Создать фильтр протоколов для открытой сети. Фильтр должен блокировать любые соединения протокола ICMP. Настроить время работы с 9 до 18	Создать фильтр для защищенной сети. Фильтр должен блокировать входящие соединения по протоколу SSH. Настроить время работы с 10 до 15
5	Создать фильтр для защищенной сети. Фильтр должен блокировать входящие соединения по	Создать фильтр для защищенной сети. Фильтр должен блокировать исходящие соединения по протоколу

№	1 задание	2 задание
	протоколу SSH. Настроить время работы с 10 до 15	TELNET. Настроить время работы с 12 до 18
6	Создать фильтр для открытой сети. Фильтр должен блокировать любые соединения протокола TCP. Настроить время работы фильтра с 8 до 18	Создать фильтр протоколов для открытой сети. Фильтр должен блокировать любые соединения протокола ICMP. Настроить время работы с 9 до 18
7	Создать фильтр для защищенной сети. Фильтр должен блокировать исходящие соединения по протоколу TELNET. Настроить время работы с 12 до 18	Создать фильтр для защищенной сети. Добавить фильтр протоколов для блокирования входящего соединения протокола TCP и настроить время работы с 13 до 14

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Для чего предназначены сетевые фильтры?

2. Какие виды трафиков существуют?

3. Что такое локальный трафик?

4. Что такое широковещательный трафик?

5. Что такое транзитный трафик?

6. Координатор является отправителем или получателем транзитных IP-пакетов?

7. Каким сетевым фильтром в ViPNet Client обрабатывается пакет по протоколу *icmp* от узла ViPNet, с которым установлена связь?

8. Каким сетевым фильтром в ViPNet Client обрабатывается пакет по протоколу *icmp* от узла ViPNet, с которым не установлена связь?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Чаплыгин В.Е., Чефранова А.О., Алабина Ю.Ф Администрирование системы защиты информации ViPNet версии 4. – М.: Горячая линия – Телеком, 2017 г. – 188 с.

2. Чефранова А.О., Алабина Ю.Ф Технология построения VPN ViPNet: курс лекций. – М.: Горячая линия – Телеком, 2017 г. – 338 с.

3. Комплект документации на ПО ViPNet Client 4 [Электронный ресурс] – https://infotecs.ru/downloads/all/vipnet-client-.html?arrFilter_93=3868181484&set_filter=Y (Дата доступа 26.12.2018).

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра информационной безопасности

Интеграция механизмов защиты в программное обеспечение

Методические указания для выполнения лабораторных и практических работ студентами групп специальностей 02.03.03, 09.00.00, 10.00.00, 11.00.00, 12.03.04, 38.05.01, 45.03.03

Курск 2022

УДК 004.056.55

Составители: О.А. Демченко

Рецензент

Кандидат технических наук, доцент *А.Л. Марухленко*

Интеграция механизмов защиты в программное обеспечение: методические указания для выполнения лабораторных и практических работ студентами групп специальностей 02.03.03, 09.00.00, 10.00.00, 11.00.00, 12.03.04, 38.05.01, 45.03.03 / Юго-Зап. гос. ун-т; сост. О.А. Демченко Курск, 2022. - 20 с.

Содержат сведения по вопросам информационной безопасности. Указывается порядок выполнения лабораторной работы, пример выполнения работы, правила оформления, содержание отчета, варианты заданий.

Методические указания разработаны для изучения дисциплин, связанными с безопасностью эксплуатации телекоммуникационных систем.

Текст печатается в авторской редакции

Подписано в печать Формат 60х84 1/16.
Усл.печ. л._____ . Уч.-изд.л _____. Тираж 30 экз. Заказ_____.

Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94.

СОДЕРЖАНИЕ

ЦЕЛЬ РАБОТЫ

Цель работы: реализовать механизм защиты программного обеспечения от анализа кода программы с помощью метода шифрования исполняемого файла.

ЗАДАНИЕ

Задание**:** овладеть практическими навыками составления шифра вручную, разработки и программирования вычислительного процесса шифрования комбинированным методом с использованием методов перестановки, моно- и поли- алфавитной подстановки, метод гаммирования, сделать отчет и написать вывод.

СОДЕРЖАНИЕ ОТЧЕТА

1. Титульный лист
2. Краткая теория
3. Выполненное задание
4. Вывод

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Защита программного обеспечения — комплекс мер, направленных на защиту программного обеспечения от несанкционированного приобретения, использования, распространения, модифицирования, изучения и воссоздания аналогов.

Основными инструментами для исследования программ являются дисассемблеры и отладчики.

Дизассемблирование — это получение из исполняемого кода программы код на языке ассемблера.

Дизассемблер - программа, осуществляющая дисассемблирова-ние.

Интерактивный дизассемблер - программа, тесно взаимодействующая с пользователем в процессе дисассемблирования.

Отладчик - программа, предназначенная для анализа поведения другой программы, обеспечивающая остановку в указанных точках и позволяющая просматривать (редактировать) содержимое ячеек памяти, регистров процессора и команды программы.

Эмулирующий отладчик - отладчик, который самостоятельно интерпретирует и выполняет команды программы (без использования реального процессора).

Существует также множество программ-утилит, предназначенных для вспомогательных операций по изучению логики работы механизма защиты. Широко используются

- шестнадцатеричные просмотрщики - редакторы;
- редакторы таблиц экспорта/импорта;
- файловые мониторы, позволяющие отслеживать операции работы с файлами;
- мониторы реестра, создающие протокол обращений к реестру
- и многие другие.

Например, с помощью файлового монитора (FileMonitor) взломщик

может отследить работу защищенной программы с файлами и обнаружить ключ (пароль), хранящийся в некотором файле. Произведя анализ протокола обращений к реестру с помощью монитора реестра (RegMon), взломщик может обнаружить ключ (пароль), хранящийся в системной базе данных Registry.

Инструментарий современного хакера настолько развит, что все попытки авторов защит противодействовать исследованию программ, с точки зрения высококвалифицированных хакеров, считаются безрезультатными.

С помощью современных версий интерактивных дизассемблеров и эмулирующих отладчиков может быть обнаружена практически любая защита.

Тем не менее, отказываться от использования приемов и методов защиты от дизассемблирования кода программы и ее работы под отладчиком нельзя.

Напомним, что абсолютной защиты вообще не бывает. Эффективной защитой считается такая защита, на взлом которой необходимы материальные и трудовые затраты, во много раз превышающие затраты на покупку программного обеспечения. Поэтому затруднение взлома защиты любыми путями и методами является оправданным. Если преодолеть защиту не сможет молодой неопытный взломщик, и заказчику придется обращаться к высококвалифицированному специалисту, - это уже плюс.

Считая защиту от дизассемблирования и отладки, рассчитанной на взломщика средней квалификации, назовем такую защиту затруднением анализа программ. Рассмотрим ее основные моменты.

Универсальным методом противодействия дизассемблированию программы является шифрование. Очевидно, что дизассемблирование зашифрованного кода бесполезно.

Применяя шифрование кода программы для противодействия дизассемблированию, следует учитывать распространенные ошибки

реализации данного метода. Напомним их.

Во-первых, неэффективной является такая реализация, когда исполняемый код в полном объеме и однократно шифруется/дешифруется (так как легко найти момент после дешифрования). Во-вторых, необходимо осуществить выбор эффективного ключа и, если необходимо, надежно хранить ключ. И в-третьих, следует учесть, что для защиты программ от дизассемблирования, не рекомендуется использование симметричных криптографических алгоритмов.

Рекомендуется использовать шифрование с открытым ключом (алгоритм RSA, шифр Эль-Гамаля и др.). В этом случае возможная удачная попытка расшифровать код и понять логику работы защитного механизма не позволит внести изменения в защищенный код, так как для полноценной последующей работы программы эти изменения необходимо внедрить в код в зашифрованном виде. А нарушителю доступен лишь ключ для расшифровки. Возможная атака в данном случае - нахождение «секретного» ключа с помощью трудоемких математических вычислений в зависимости от используемого алгоритма шифрования.

Усиливает защиту динамическое шифрование и многопроходная расшифровка кода.

На практике неплохо зарекомендовали себя и методы, использующие вместе с шифрованием архивирование программного кода. К достоинствам данного метода относят и уменьшение размера исполняемого файла. Однако следует учитывать, что алгоритмы работы широко используемых архиваторов известны многим взломщикам.

Широко распространены на практике методы, основанные на динамическом изменении кода программы в процессе выполнения. Суть этих методов сводится к получению истинных исполнимых команд на этапе выполнения программы путем некоторого преобразования первоначальных кодов. Часто этот способ защиты называют самогенерируемыми, или самомодифицирующимися кодами.

Авторы предлагают различные преобразования, например,

- перемещения участков кода;
- всевозможные функции от истинного кода (контрольной суммы истинного кода);
- или для генерации кода одного участка используют коды предыдущего (или какого нибудь другого) участка программы (так называемая обратная связь).

Интересный прием защиты от дизассемблирования - использование нестандартной структуры программы. В этом случае дизассемблер «не поймет» нестандартную сегментацию программы.

Для того чтобы лучше понять методы борьбы с отладчиками и пути увеличения эффективности этих методов, напомним суть процесса работы программы под отладчиком.

Существует два отладочных механизма:

1) контрольные точки останова и
2) трассировка программы.

Контрольные точки останова

Идея механизма заключается во внесении в програм-мный код специального однобайтового кода (0xCC) - так называемой контрольной точки останова.

Заметим, что можно внести в про-граммный код любое количество таких точек.

Напомним, во время выполнения программы при достижении контрольной точки останова возникает исключительная ситуация - прерывание int 3. В этот момент процессор останавливает работу программы для дальнейших распоряжений пользователя. Для того, чтобы позже продолжить работу программы с точки останова, в стеке запоминаются значения регистра флагов, регистра CS (указатель текущего кодового

сегмента) и регистра IP (указатель на следующую выполнимую команду). При этом сбрасывается флаг трассировки.

Итак, для анализа программы с помощью отладчика в исполняемый код программы необходимо внести контрольные точки останова, следовательно, изменить код!

Этим фактом успешно пользовались авторы защит: достаточно было обнаружить модификацию кода и либо удалить точку останова, либо прекратить дальнейшее выполнение программы.

Для обнаружения модифицированного кода традиционно применялись следующие методы:

- подсчет контрольных сумм критических участков;
- использование контрольной суммы всего кода для расшифровки некоторого фрагмента;
- многопроходная расшифровка кода с ключом, вычисляемым на основе контрольной суммы всего кода либо критического участка;
- использование корректирующих кодов, позволяющих определить местоположение контрольного байта;
- контроль времени выполнения критического участка по сравнению с эталонным временем;
- контроль относительного времени выполнения участка программы (относительно другого участка)
- и другие.

Интересные методы противодействия отладчикам реального (!) режима основаны на перехвате необходимого отладчику прерывания int 3. Предлагается заменить команды обработчика прерывания int 3 «мусором» либо использовать обработчик прерывания в целях защиты, например, для

расшифровки кода. Тогда в первом случае отладчик будет просто нейтрализован, а во втором - возникнет конфликт между механизмом защиты и отладчиком.

Существует целая группа приемов, основанных на использовании отладчиком стека.

Одним из таких приемов является следующий. При выполнении критического участка необходимо присвоить указателю стека нулевое значение. Таким образом, стек считается полным и не может быть использован для сохранения необходимых регистров. Операционная система в таком случае завершает работу приложения.

Другой прием - хранить в стеке (полностью используя стек) данные, необходимые для работы программы. Отладчик, при использовании стека для записи значений необходимых регистров, удалит (затрет) критические данные, и программа станет неработоспособной.

Рассмотрим второй отладочный механизм - трассировку кода программы.

Трассировка - это пошаговое выполнение программы.

Напомним, установка специального флага трассировки (TF) приводит к генерированию после каждой команды исключительной ситуации - прерывания int 1, называемого трассировочным прерыванием.

При этом, аналогично обработке исключительной ситации int 3, в стеке сохраняются необходимые для дальнейшей работы значения регистра флагов и регистра IP (указатель на следующую выполнимую команду).

Поэтому наряду с простыми проверками - установлен ли флаг трассировки – для противодействия трассировке могут быть использованы и перечисленные выше приемы.

Очевидно, что трассировщик будет стараться следить за флагом TF и корректировать его. Поэтому необходимо для получения оригинального флага трассировки использовать специальные приемы.

Проверять установку флага трассировки можно, исходя из

аппаратных особенностей процессора, например, используя потерю трассировочного прерывания. Этот метод базируется на том, что после команд, изменяющих сегментный регистр SS (до процессора Intel 80386 - любой сегментный регистр), не происходит трассировочное прерывание даже при установленном флаге трассировки. Чтобы получить истинное значение флага трассировки, достаточно перед проверкой регистра флагов переслать сегментный регистр SS.

Для усиления защиты от пошагового выполнения программы авторы успешно используют значение флага трассировки (в совокуп-ности с другими параметрами) для расшифровки критических участ-ков или, что еще сложнее для взлома, в арифметических выражениях.

Современные операционные системы и отладчики позволяют установить аппаратные точки останова.

Аппаратная отладка основана на использовании специальных отладочных регистров (напомним, что их всего 8 регистров) и возможности простановки четырех контрольных точек останова.

Аппаратные точки останова никак не модифицируют код и, следовательно, не могут быть обнаружены традиционными средствами.

Однако существуют приемы, позволяющие, в принципе, обнаружить работу программы под аппаратной отладкой. Для этого можно использовать все отладочные регистры для нужд программы или поместить в отладочные регистры «мусор». Усиливается защита использованием всех четырех контрольных точек останова. Но реализация таких приемов требует высокой квалификации программиста и вряд ли оправдана, так как для взлома защиты нарушитель может воспользоваться другими технологиями и инструментами.

Для взлома механизмов защиты сегодня применяются мощные эмулирующие отладчики. Они самостоятельно (без помощи процессора) интерпретируют и выполняют команды исследуемой программы. Существуют так называемые отладчики с неполной эмуляцией. Эти

отладчики интерпретируют только некоторые команды, а остальные выполняют на реальном процессоре.

Очевидно, что против таких отладчиков бессильны любые приемы противодействия. Поэтому «сегодня уже мало кто решается противодействовать отладчику и включать в свое приложение антиотладочный код. Мода на это давно прошла».

Единственным средством борьбы с отладкой сами хакеры называют эмуляторы процессора.

В рамках примера выполнения практической работы будет реализовано двойное гаммирование. Программа дважды просит пользователя ввести строку-ключ для шифрования. Затем каждый раз введённая строка преобразуется в число, которым инициализируется генератор псевдослучайных чисел. После этого генератор порождает последовательность псевдослучайных чисел такой же длины, что и длина файла и потом каждый байт этой последовательности накладывается по логической операции XOR с соответствующим байтом файла. Для надёжности опять зашифруем уже зашифрованный файл таким же образом. Дешифрование происходит точно также и поэтому одна и та же программа используется для шифрования и для дешифрования.

ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

Воспользуемся методом перестановок. Идея этого метода криптографии заключается в том, что запись открытого текста и последующее считывание шифровки производятся по разным путям внутри некоторой геометрической фигуры (например, квадрата).

Для пояснения идеи возьмем квадратную таблицу (матрицу) размером 8 х 8 (будем записывать текст последовательно по строкам сверху вниз, а считывать -последовательно по столбцам слева направо.

Предположим, что требуется зашифровать сообщение:

НА ПЕРВОМ КУРСЕ ТЯЖЕЛО УЧИТЬСЯ ТОЛЬКО ПЕРВЫЕ ЧЕТЫРЕ ГОДА ДЕКАНАТ

Запишем его в матрицу:

	1	2	3	4	5	6	7	8
1	Н	А		П	Е	Р	В	0
2	М		К	У	Р	С	Е	
3	Т	Я	Ж	Е	Л	0		У
4	Ч	И	Т	Ь	С	Я		Т
5	0	Л	Ь	К	0		П	Е
6	Р	В	Ы	Е		Ч	Е	Т
7	Ы	Р	Е		г	0	Д	А
8		Д	Е	К	А	Н	А	Т

В матрице символом «_» обозначен пробел. В результате преобразований получится шифровка:

НМТЧРЫ_А_ЯИЛВРД_КЖТЬЫЕЕПУЕЬКЕ_КЕРЛСО_ГАРСОЯ_ЧО НВЕ_ПЕДАО_УТЕТ АТ.

Ключом в данном случае является размер матрицы, порядок записи открытого текста и считывания шифрограммы. Естественно, что ключ может быть другим. Например, запись открытого текста по строкам может производиться в таком порядке номеров строк:
4 8 1 2 7 6 5 3 , а считывание криптограммы может происходить по столбцам в следующем порядке: 8 1 3 5 7 6 4 2 .

Будем называть порядок записи в строки матрицы ключом записи, а порядок считывания шифрограммы по столбцам — ключом считывания.

Чтобы дешифровать криптограмму, полученную с помощью матрицы размером п х п, нужно разбить эту криптограмму на группы символов по п символов в каждой группе. Крайнюю левую группу записать сверху вниз в столбец, номер которого совпадает с первой цифрой ключа считывания. Вторую группу символов записать в столбец, номер которого совпадает со второй цифрой ключа считывания, и т. д. Открытый текст считывать из матрицы по строкам в соответствии с цифрами ключа записи.

Рассмотрим пример дешифрации криптограммы, полученной методом перестановок. Известно, что при шифровании использованы матрица размером 6х6, ключ записи 3 5 2 1 4 6 и ключ считывания 4 2 5 3 1 6

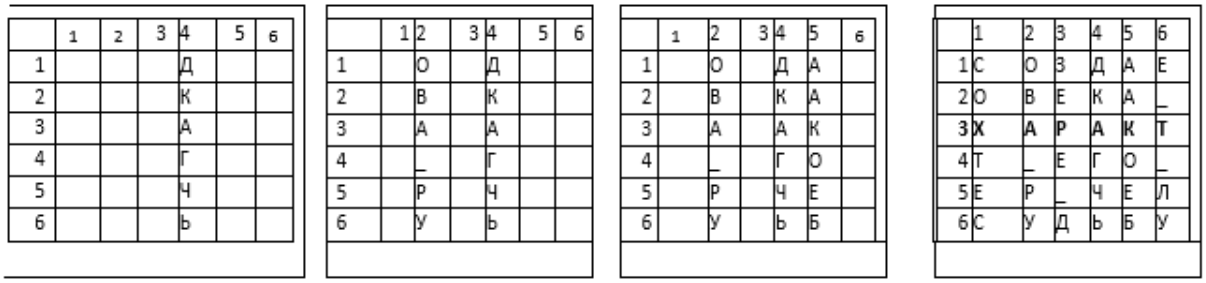
. Текст шифрограммы таков:

ДКАГЧЬОВА_РУААКОЕБЗЕРЕ_ДСОХТЕСЕ_Т_ЛУ

Разобьем шифрограмму на группы по 6 символов:

ДКАГЧЬ ОВА_РУ ААКОЕБ ЗЕРЕ_Д СОХТЕС Е_Т_ЛУ

Первую группу символов запишем в столбец 4 матрицы, так как первая цифра ключа считывания — 4. Вторую группу из 6 символов запишем в столбец 2, третью группу символов — в столбец 5 и т. д.:



	1	2	3	4	5	6
1				Д		
2				К		
3				А		
4				Г		
5				Ч		
6				Ь		

	1	2	3	4	5	6
1		О		Д		
2		В		К		
3		А		А		
4		_		Г		
5		Р		Ч		
6		У		Ь		

	1	2	3	4	5	6
1		О		Д	А	
2		В		К	А	
3		А		А	К	
4		_		Г	О	
5		Р		Ч	Е	
6		У		Ь	Б	

	1	2	3	4	5	6
1	С	О	З	Д	А	Е
2	О	В	Е	К	А	_
3	Х	А	Р	А	К	Т
4	Т	_	Е	Г	О	_
5	Е	Р	_	Ч	Е	Л
6	С	У	Д	Ь	Б	У

Считывание открытого текста в соответствии с ключом записи начинаем со строки 3, затем считываем строку 5 и т. д. В результате дешифрования получаем открытый текст:

ХАРАКТЕР ЧЕЛОВЕКА СОЗДАЕТ ЕГО СУДЬБУ

Естественно, что описанную процедуру дешифрования криптограммы можно произвести автоматически на компьютере с помощью заранее разработанных программ.

Эффективным средством повышения криптостойкости шифров является комбинированное использование нескольких различных способов шифрования с помощью двух или более методов.

Как показали исследования, стойкость комбинированного шифрования Sк не ниже произведения стойкости используемых способов Si , т.е. Sк>= П Si

Комбинировать можно любые методы шифрования и в любом количестве, однако на практике наибольшее распространение получили следующие комбинации:

1. Перестановка + подстановка(замена)
2. Перестановка + гаммирование

3. Гаммирование+ подстановка(замена)

4. Гаммирование+ гаммирование

Типичным примером комбинированного шифра является национальный стандарт США криптографического закрытия данных (DES).

Английский алфавит

A B C D E F G H I J K L M N O P Q P S T U V W X Y Z 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

ВАРИАНТЫ ЗАДАНИЙ

Вариант 1.

Составить программу , которая шифрует сообщение Complete support of RAR and ZIP archives следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу шифром по Таблице Виженера:

1. *3 уровень*

C. Дополнить полученную программу шифром Атбаш, расположив буквы построчно.

Вариант 2.

Составить программу , которая шифрует сообщение Make your own settings by modifying or creating a KBD file следующим образом:

1 уровень

A. Составить программу , которая первым использует шифр перстановок.

2 уровень

B. Дополнить полученную программу, которая использует Аффинные криптосистемы а=5 и в=3

2. *3 уровень*

C. Дополнить полученную программу шифром Квадрата Бьюфорта, расположив буквы построчно.

Вариант 3.

Составить программу , которая шифрует сообщение Use Script to assign a keystroke to a script command or function следующим образом:

1 уровень

A. Составить программу которая использует метод перестановок.

2 уровень

B. Дополнить полученную программу, которая использует Аффинные криптосистемы а=5 и в=7

3. *3 уровень*

C. Дополнить полученную программу шифром по Таблице Виженера, расположив буквы построчно.

Вариант 4.

Составить программу , которая шифрует сообщение *You can map your keyboard the following ways* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу, которая шифрует сообщение по Таблице Бьюфорта

4. *3 уровень*

C. Дополнить полученную программу, которая шифрует сообщение Квадратом

Полибия [6x6]

Вариант 5.

Составить программу , которая шифрует сообщение *In the IDE, the keystroke repeat rate defaults to OFF* следующим образом:

1 уровень

A. Составить программу, которая которая использует метод перестановок

2 уровень

B. Дополнить полученную программу, которая шифрует сообщение шифром Атбаш

5. *3 уровень*

C. Дополнить полученную программу, которая шифрует сообщение квадратом Полибия [6x6]

Вариант 6.

Составить программу , которая шифрует сообщение *Click one of the following buttons to open a KBD file in Notepad* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 *уровень*

B. Дополнить полученную программу, которая шифрует сообщение квадратом Бьюфорта

6. *3 уровень*

C. Дополнить полученную программу, которая шифрует сообщение Таблицей Виженера

Вариант 7.

Составить программу , которая шифрует сообщение *Changing the mapping of your keyboard* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу, которая использует шифр Цезаря

7. *3 уровень*

C. Дополнить полученную программу, которая шифрует сообщение Афинными

Вариант 8.

Составить программу , которая шифрует сообщение *The available Editor SpeedSettings are* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу, которая использует Аффинные криптосистемы а=5 и в=9

8. *3 уровень*

C. Дополнить полученную программу шифром Квадратом Полибия [6x6]

Вариант 9.

Составить программу , которая шифрует сообщение *To customize how text is*

displayed

следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок.

2 уровень

B. Дополнить полученную программу шифром, который при закрытии сообщения

9. *3 уровень*

C. Дополнить полученную программу шифром, который закрывает сообщение Квадратом Полибия [6x6]

Вариант 10.

Составить программу , которая шифрует сообщение *Select the type of text you want* следующим образом:*уровень*

A. Составить программу, которая использует метод перестановок

1 уровень

B. Дополнить полученную программу шифром Атбаш.

10. *3 уровень*

C. Дополнить полученную программу шифром Цезаря.

Вариант 11.

Составить программу , которая шифрует сообщение *The number of available text types* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу шифром, который закрывает сообщение Табл

11. *3 уровень*

C. Дополнить полученную программу шифром Цезаря.

Вариант 12.

Составить программу , которая шифрует сообщение *Choose a font, size, attribute, and color* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу шифром по Таблице Виженера

12. *3 уровень*

C. Дополнить полученную программу шифром, который закрывает сообщение Квадратом Полибия [6x6]

Вариант 13.

Составить программу , которая шифрует сообщение *For example, only one type of text* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу шифром Атбаш.

13. *3 уровень*

C. Дополнить полученную программу шифром, который закрывает сообщение Таблицей Виженера.

Вариант 14.

Составить программу , которая шифрует сообщение *Makes the specified text display in boldface* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу шифром Цезаря.

14. *3 уровень*

C. Дополнить полученную программу шифром Атбаш

Вариант 15.

Составить программу , которая шифрует сообщение *Select a view and the type of text*

следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу шифром Цезаря.

15. *3 уровень*

C. Дополнить полученную программу Таблицей Виженера

Вариант 16

Составить программу , которая шифрует сообщение *Changing the mapping of your keyboard* следующим образом:

1 уровень

A. Составить программу, которая использует метод перестановок

2 уровень

B. Дополнить полученную программу, которая использует шифр Атбаш

16. *3 уровень*

C. Дополнить полученную программу, которая шифрует сообщение Афинными криптосистемами a=9, b=3

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Ярочкин, В.И. Информационная безопасность: учебник для вузов по гуманит. и социал.-экон. спец. / В.И. Ярочкин – М.: Академический Проект: Трикста, 2005. – 543 с. (шифр: 004 Я805).
2. Башлы, П.Н. Информатика: учебное пособие / П.Н. Башлы. – Ростов-н/Д.: Феникс, 2006. – 250 с. (шифр: 004 Б335).
3. Торокин, А.А. Инженерно-техническая защита информации: учебн. пособие для студентов, обучающихся по специальности в области информ. безопасности / А.А. Торокин. – М.: Гелиос АРВ, 2005. – 960 с.
4. Зайцев, А.П. Техническая защита информации: учебн. пособие / А.П. Зайцев, А.А. Шелупанов. – М.: Горячая линия-Телеком, 2007. – 616 с.
5. Хореев, А.А. Техническая защита информации: учеб. пособие для студентов вузов. В 3 т. Том 1. Технические каналы утечки информации / А.А. Хореев. – М.: НПЦ «Аналитика», 2008. – 436 с.