• Разработка планов оценки
• Проведение оценки и анализ результатов
• Анализ итогового отчета и последующие действия.

Оценка осуществляется экспертом или экспертной группой.

На подготовительном этапе рассматривается ряд вопросов, касающихся стоимости, расписания и выполнения оценки, проводится определение целей и области оценки, выбор эксперта или экспертной группы, а также сбор необходимых для оценки данных.

План оценки безопасности содержит цели оценки и детальное описание ее проведения. В ходе разработки планов эксперт определяет включаемые в план защитные меры, а также выбирает, адаптирует и оптимизирует надлежащие процедуры оценки.

После утверждения плана оценки безопасности эксперт или экспертная группа приступает к его выполнению. Оценка мер безопасности производится путем применения к объектам оценки методов оценки. Процедура оценки зависит от оцениваемой меры безопасности (NIST 800-53 разделяет все меры на 17 семейств. Описание методов оценки приводится в приложении к стандарту). При использовании мер безопасности, не описанных в NIST 800-53 эксперты указывают это в плане безопасности и разрабатывают свои процедуры оценки. В качестве методов оценки NIST 800-53 выделяет исследование, интервью и тестирование. На основе собранных свидетельств делается вывод о соответствии или несоответствии мер безопасности предъявляемым требованиям. По результатам оценки безопасности составляется отчет и описание мероприятий по устранению несоответствий.

MEHARI 2010

Методология MEHARI позволяет осуществить полный процесс управления информационными рисками в соответствии со стандартом ISO/IEC 27000. В базу знаний методологии включена оценка оценка комплектации средствами защиты по стандарту ISO/IEC 27002:2005, что позволяет организации подготовиться к прохождению аудита на соответствие стандарту.

Согласно предложенной в MEHARI методике, домены безопасности декомпозируются на сервисы безопасности, состоящие из подсервисов, в которые, в свою очередь, сгруппированы меры безопасности.

Процесс оценки сервиса безопасности осуществляется путем распространения среди подразделений специальных вопросников позволяющих оценить качество решений, направленных на снижение риска. Опросники заполняются аудитором в процессе интервьюирования персонала подразделения.

Выделяется три показателя производительности сервиса безопасности:
• эффективность;
• надежность;
• постоянство.

Качество сервиса безопасности оценивается по шкале от 0 до 4, отражающей сложность нарушения работоспособности сервиса. Описание каждого из уровней приведено в документе MEHARI 2010: Evaluation Guide for security services.

Выводы

Выполненный анализ существующих подходов к проведению проверок на соответствие системе требований ИБ позволяет сделать следующие выводы:

• каждая методика содержит свой перечень требований ИБ, сгруппированных по направлениям оценки, являющихся наиболее важными с точки зрения авторов;

• в качестве демонстрации соответствия требованиям ИБ используются метрики, числовое выражение которых позволяет судить о степени выполнения конкретных требований ИБ и направлений в целом;

• методики отличаются в алгоритме расчета итоговых метрик, но в целом являются общими применение функций свертки с использованием весовых коэффициентов требований.

Таким образом, следует отметить, что при внедрении процесса управления требованиями необходимо определить наиболее важные направления требований ИБ, инвентаризировать их, классифицировать по степени важности и утвердить формулу свертки для определения итогового уровня соответствия СОИБ требованиям ИБ.

Список литературы

1. СТО БР ИББС-1.2-2014 Методика оценки соответствия информационной безопасности организаций банковской системы Российской Федерации требованиям СТО БР ИББС-1.0-2014, 2014.
2. Положение Банка России N 382-П О требованиях к обеспечению защиты информации при осуществлении переводов денежных средств, 2012.
3. NIST SP 800-53A Guide for Assessing the Security Controls in Federal Information Systems and Organizations, 2010.
4. MEHARI 2010 knowledge base, 2010.
5. MEHARI 2010: Risk analysis and treatment Guide, 2010
6. MEHARI 2010: Security Stakes Analysis and Classification Guide, 2010.
7. MEHARI 2010: Processing guide for risk analysis and management, 2010.
8. MEHARI 2010: Fundamental concepts and functional specifications, 2010.
9. MEHARI 2010: Evaluation Guide for security services, 2010
10. Управление информационными рисками на основе методологии MEHARI / М.К. Янчин, 2011.

АЛГОРИТМ РАЗРЕШЕНИЯ ПРОТИВОРЕЧИЙ В ПРАВИЛАХ ФИЛЬТРАЦИИ

Семакин А.И., Мордвин Д.В.
Южный федеральный университет, Таганрог, e-mail: neverminden@gmail.com

Обеспечение безопасности компьютерных сетей является важной задачей при построении систем защиты информации. Межсетевые экраны (МЭ) играют большую роль для сетевой безопасности. Межсетевое экранирование – широко распространенный механизм для усиления защищенности сетей организаций. Однако конфигурирование МЭ является сложной задачей, подверженной возникновению ошибок, даже для опытных администраторов. В результате, ошибки конфигурирования МЭ, в частности, противоречия в правилах фильтрации трафика в сети являются достаточно распространенными и серьезными.

Обычно правила фильтрации состоят из полей адреса источника, адреса назначения, порта источника, порта назначения, протокола и действия фильтрации (разрешить или запретить).

Основными видами аномалий для конфигураций МЭ являются несогласованность и неэффективность правил фильтрации трафика.

Конфигурации МЭ отображает намерения администратора, которые должны быть согласованы между собой. Поэтому несогласованность правил в конфигурации часто является хорошим индикатором ошибок конфигурации.

Таблица 1

Пример конфигурационного файла 1

1.	deny tcp 10.1.1.0/25 any
2.	accept udp any 192.168.1.0/24
3.	deny tcp 10.1.1.128/25 any
4.	deny udp 172.16.1.0/24 192.168.1.0/24
5.	accept tcp 10.1.1.0/24 any
6.	deny udp 10.1.1.0/24 192.168.0.0/16
7.	accept udp 172.16.1.0/24 any

На сегодняшний день определены следующие типы несогласованности между правилами:

а) Затенение определяется в случаях, когда трафик, который одно правило намеревается запретить (разрешить) разрешен (запрещен) предшествующим правилом. Это часто свидетельствует о неправильной конфигурации и является серьезной ошибкой. Правило может быть затенено предшествующим правилом, которому удовлетворяет множество пакетов. В таблице 1 запрещающее правило 4 затенено разрешающим правилом 2, потому что каждый UDP с адресом источника из подсети 172.16.1.0/24 и адресом назначения в подсети 192.168.1.0/24 (правило 4) разрешается правилом 2, которому удовлетворяют все пакеты, направленные в подсеть 192.168.1.0/24.

Пусть правило A предшествует правилу B в списке правил. Тогда правило B будет затенено правилом A, если выполняется условие 1.

$$(A \cap B = B) \wedge (action(B) \neq action(A)) \qquad (1)$$

б) Обобщение определяется в случаях, где подмножество пакетов, удовлетворяющих текущему правилу, было исключено предшествующим правилом. Такая ошибка конфигурирования противоположна ранее описанному затенению и случается, когда предшествующее правило соответствует части текущего правила, но задает другую реакцию. В табл. 1 правило 7 является обобщением правила 4, потому что UDP пакеты из подсети 172.16.1.0/24 в направлении 192.168.1.0/24 является подмножеством UDP пакетов с источником 172.16.1.0/24 (правило 7), но принимаемые решение для этого подмножества в правилах 4 и 7 различны.

Пусть правило A предшествует правилу B в списке правил. Тогда правило B будет обобщать правило A, если выполняется условие 2.

$$(A \cap B = A) \wedge (action(B) \neq action(A)) \qquad (2)$$

в) Пересечение определяется в случаях, когда текущее правило пересекается с предыдущими правилами, но определяет иную реакцию. Решение в таких случаях будет зависеть от порядка пересекающихся правил. Правила 2 и 6 в табл. 1 пересекаются друг с другом. Их пересечение можно определить как «udp 10.1.1.0/24 192.168.1.0/24». Предшествующее по порядку правило определяет судьбу пакетов, удовлетворяющих этому пересечению. Для правил A и B в списке правил будет определено пересечение, если выполняется условие 3.

$$(A \cap B =) \wedge (A \neq D \wedge B \neq D) \wedge (action(B) \neq action(A)) \qquad (3)$$

МЭ должны проверять огромное количество пакетов. Поэтому сложно обойти стороной эффективность МЭ. Для увеличения эффективности МЭ очень важно правильное конфигурирование правил МЭ. Эффективно настроенный МЭ должен использовать минимальное количество правил и памяти и помнить и использовать собственные вычисления различных фильтрационных запросов.

Хотя неэффективность прямо не указывает на уязвимость, более быстрый и эффективный МЭ будет способствовать более активному использованию МЭ в сети и тем самым повысит её безопасность.

Неэффективность бывает двух типов:

а) Избыточность можно определить в случаях, когда при удалении правила реакция МЭ не изменится ни для одного пакета. Уменьшение избыточности может уменьшить общее количество правил и следовательно потребление памяти. Правило может быть определено как избыточное, если какое-либо предшествующее правило удовлетворило рассматриваемому запросу и определило ту же самую реакцию. Для примера в табл. 2 правило 3 избыточно, так как правило 2 уже определило ту же реакцию для всех пакетов, удовлетворяющих правилу 3.

Таблица 2

Пример конфигурационного файла 2

1.	accept tcp 192.168.1.1/32 172.16.1.1/32
2.	accept tcp 10.0.0.0/8 any
3.	accept tcp 10.2.1.0/24 any
4.	deny tcp any any
5.	deny udp 10.1.1.0/26 any
6.	deny udp 10.1.1.64/26 any
7.	deny udp 10.1.1.128/26 any
8.	deny udp 10.1.1.192/26 any
9.	deny udp any

Для правил A и B в списке правил правило A будет избыточно, если выполняется условие 4.

$$(A \cap B = A) \wedge (action(B) = action(A)) \qquad (4)$$

б) Многословность определяется в случаях, когда некоторое множество правил может быть преобразовано в меньшее множество правил. Например, правила 5, 6, 7 и 8 в таблице 2 могут быть просуммированы в одно правило «deny udp 10.1.1.0/24 any». Многословность часто случается на практике, когда администраторы дорабатывают правила фильтрации через некоторый период времени.

Решение проблемы

Учитывая проблемы, приведенные выше, нами поставлена задача разработать автоматизированную систему поиска и устранения противоречий в правилах фильтрации сетевого трафика. Система предназначена для системных администраторов и служит для повышения безопасности и производительности сети.

Исходными данными для системы являются конфигурационные файлы МЭ.

В результате работы системы формируется отчет с указанием найденных противоречий в правилах и рекомендациями по их устранению. Для определения типа ошибки между двумя правилами мы предлагаем следующий алгоритм.

Алгоритм определения типа ошибки между правилами

Входные данные: два правила фильтрации.

Выходные данные: тип ошибки между правилами фильтрации.

Основные шаги:

а) Для правила отдельно по всем параметрам определяются ситуации между ними.

б) Если по всем параметрам есть затенение, протоколы совпадают, а действия у правил различные – найдено затенение между правилами.

в) Если по всем параметрам есть затенение, протоколы совпадают, а действия у правил одинаковые – найдена избыточность. Для алгоритма важно, что это затеняющая избыточность.

д) Если по всем параметрам есть обобщение, протоколы совпадают, а действия у правил различные – найдено обобщение.

е) Если по всем параметрам есть обобщение, протоколы совпадают, а действия у правил одинаковые – найдена избыточность. Для алгоритма важно, что это обобщающая избыточность.

ж) Если по всем параметрам есть какая-либо ошибка, протоколы совпадают – найдено пересечение.

и) Ошибки между правилами нет.

Далее рассмотрим предложенный нами алгоритм поиска ошибок конфигурирования МЭ.

Основная идея алгоритма основана на построении дерева правил МЭ по заданному списку правил МЭ. Данное дерево имеет следующие особенности:

– в корне дерева содержится правило, означающее политику разграничения доступа по умолчанию для заданного МЭ;

– каждый узел дерева может иметь произвольное количество сыновей;

– сыном в дереве может быть только такое правило, которое обобщается (затенено) правилом в родительском узле дерева;

– пересекающиеся и не пересекающиеся правила становятся узлами-братьями, если они оба затенены одним и тем же правилом.

Алгоритм определения ошибок конфигурирования

Входные данные: список правил фильтрации.

Выходные данные: список ошибок между заданными правилами, дерево правил.

Основные шаги:

а) В корень дерева помещается политика разграничения доступа по умолчанию для заданного МЭ.

б) Для каждого правила в списке правил МЭ последовательно выполняются следующие действия.

в) Добавляемое в дерево правило последовательно сравнивается с каждым сыном корневого правила на наличие ошибки между ними.

д) Если добавляемое правило обобщает какие-то из правил, с которыми проводилось сравнение. Тогда новое правило добавляется в дерево сыном текущего родителя. А все правила, которые были обобщены добавляемым правилом, становятся его сыновьями в дереве. При этом необходимо сообщить о найденных обобщениях между правилами.

е) Если добавляемое правило затенено каким-либо из правил, с которыми проводилось сравнение. Тогда родителем для добавляемого правила должно стать последнее затеняющее правило в поддереве первого найденного затеняющего. Для того чтобы найти это последнее затеняющее правило, которое должно стать родителем для добавляемого правила необходимо рекурсивно повторять процедуру добавления данного правила, но родителем для него считать текущее найденного затеняющее его правило. Также необходимо сообщить о всех найденных затенениях.

ж) Если добавляемое правило пересекается с каким-либо из правил, оно добавляется сыном текущего родительского узла. Необходимо проверить пересечения с остальными сыновьями текущего родительского правила и сообщить о всех найденных пересечениях. Если ошибок нет, добавляемое правило становится сыном текущего родительского узла.

и) Ошибки избыточности определяются, когда построение дерева закончено. Для этого необходимо обойти дерево и определить случаи, где действие родительского правила совпадает с действием дочернего правила.

Устранение противоречий

После того, как все противоречия найдены, система выдаёт рекомендации по их исправлению.

а) Если найдено затенение между двумя правилами, то система предлагает:

– переместить затеняемое правило выше затеняющего;

– переписать затеняемое правило;

– переписать затеняющее правило;

– удалить затеняемое правило;

– удалить затеняющее правило.

б) Если найдено обобщение между двумя правилами, то система предлагает:

– ничего не делать, так как обобщение могло быть допущено администратором специально;

– переписать обобщаемое правило;

– переписать обобщающее правило;

– удалить обобщаемое правило;

– удалить обобщающее правило.

в) Если найдено пересечение между двумя правилами, то система предлагает:

– разбить пересекающиеся правила на несколько и задать правильную реакцию для каждого;

– удалить пересекающиеся правила.

г) Если найдена избыточность между двумя правилами, то система предлагает:

– удалить избыточное правило;

– исправить избыточное правило.

Также, для каждого предлагаемого метода устранения противоречий отображается результат, который получится в случае выбора данного метода, и его пояснение.

Таким образом, в результате анализа конфигураций МЭ с помощью нашей системы, системный администратор, руководствуясь рекомендациями по устранению противоречий, может значительно повысить безопасность и производительность сети.

Следует отметить, что предложенные нами подходы и алгоритмы применимы как для одного МЭ, так и для множества МЭ в сети. Прототип системы находится в разработке. Его реализация и экспериментальное исследование будут описаны отдельно.

АЛГОРИТМ ПРИМЕНЕНИЯ КОНТРМЕР К УЯЗВИМОСТЯМ В СЕТИ НА ОСНОВЕ ГРАФА АТАК

Штанько А.С., Мордвин Д.В.

Южный федеральный университет, Таганрог, e-mail: neverminden@gmail.com

В настоящее время компьютерные сети имеют повсеместное распространение. Сетевые технологии применяются во всех сферах деятельности людей. Эксплуатация уязвимостей, присутствующих в сетях, может привести к большим убыткам различного характера. Кроме того постоянный рост размера сетей и усложнение их структуры замедляют процесс обнаружения угроз безопасности и снижают скорость применения контрмер. По этой причине остается актуальной проблема разработки удобного способа выработки контрмер направленных на минимизацию количество уязвимостей вычислительной сети.

Статья представляет собой описание предлагаемого нами способа выявления наиболее критичных узлов защищаемой сети и алгоритма системы автоматизированного расчета контрмер для устранения уязвимостей в защищаемой сети. В данной работе под контрмерами понимаются действия различного характера, направленные снижения количественной оценки уровня опасности каждого узла. Статья представляет собой продолжение исследований проблем, представленных в других публикациях [1].

Статья состоит из трех частей. В первой части описана структура необходимой модели сети и способ поиска ее наиболее уязвимых узлов. Вторая часть описывает характер мер, направленных на минимизацию уязвимостей вычислительной сети. Третий раздел дает вербальное описание разработанного нами алгоритма функционирования разрабатываемой автоматизированной системы расчета контрмер для устранения уязвимостей в защищаемой сети. В разделе вы-