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-

водов приведен краткий анализ проделанной работы и задан вектор для дальнейших исследований.

1. Использование модели сети для поиска наиболее уязвимых узлов

Для разработки контрмер, направленных на устранение уязвимостей в защищаемой сети необходимо иметь максимально подробную информацию об этой сети. Наиболее удобной формой представления такой информации является компьютерное моделирование защищаемой сети. Мы считаем, что использование модели сети предпочтительна, поскольку это позволяет облегчить анализ защищенности сети. Модель целевой сети может быть получена в результате ее сканирования или построена администратором вручную.

В данном исследовании мы предлагаем использовать модель сети, разработанной ранее в другом проекте [1]. Данная модель карты сети включает в себя информацию о связности, маршрутизации, фильтрации и возможность расчета доступа между узлами. Помимо информации о структуре сети модель содержит информацию о каждом узле, которая включает в себя сведения об используемых на хосте сервисах (имя CPE) и об их уязвимостях (CVE-код).

Используя полученную модель карты сети автоматизированная система расчета контрмер должна определять уязвимые точки сети. Для этого система должна определять количественную оценку опасности каждой. В данной работе для расчета оценки опасности узла сети мы предлагаем использовать базовые метрики CVSS [2].

После получения количественной оценки опасности для каждого узла сети, необходимо выделить наиболее критичные из них.

На основе полученного списка критичных точек сети можно генерировать контрмеры. Данная часть системы будет рассмотрена далее.

2. Контрмеры направленные на устранение уязвимостей в защищаемой сети

Контрмеры для минимизации уязвимостей, имеющихся в целевой сети, должны быть не только эффективны, но и целесообразны. Они должны вносить минимальные изменения в архитектуру защищаемой сети.

На наш взгляд, контрмеры для обеспечения безопасности уязвимых узлов сети можно разделить на несколько направлений:

• контрмеры направленные на уязвимый сервис;

• контрмеры направленные на ограничение доступа к уязвимому сервису;

• контрмеры направленные на повышения уровня защищенности вычислительной сети в целом.

Данные направления расположены в порядке убывания их приоритета и подробно будут рассмотрены далее.

2.1. Контрмеры, направленные на уязвимый сервис. Мы считаем что, данное направление контрмер является наиболее приоритетным, поскольку действия, направленные на устранение уязвимости в самом сервисе не несут никаких изменений в архитектуру защищаемой сети. Ниже приведены способы устранения уязвимостей в сервисе:

Замена сервиса версией, в которой отсутствует уязвимость. Такой способ достаточно эффективный, может быть применим на используемом сервисе, только если альтернативная версия обладает всем необходимым функционалом. Сервис можно заменить как более новой (обновление), так и более старой версией (“откат”), что может повлечь за собой ликвидацию/изменения некоторых возможностей используемого ПО.

Замена сервиса на альтернативный, в котором отсутствует уязвимость. Этот метод имеет высокий уровень эффективности, но уже не так удобен, как способ описанный выше. Дело в том, что замена сервиса на альтернативный может быть осуществлена только в том случае, если он дублирует функционал заменяемого сервиса. Кроме того замена ПО может повлечь за собой реконфигурацию каждого узла сети, на котором данный сервис заменен.

Перемещения сервиса на отдельный узел сети. Применение данного метода повышения уровня защищенности целесообразно лишь в том случае, когда уязвимость в текущем сервисе возможно эксплуатировать, только после эксплуатации уязвимости в другом сервисе на данном узле сети. В этом случае необходимо переместить текущий сервис на отдельный узел сети, что позволит предотвратить эксплуатацию его уязвимости.

Удаление уязвимого сервиса. Данный способ избавления от уязвимости является очень эффективным, но может быть применен только тогда, когда удаляемый сервис абсолютно неиспользуемый.

2.2. Контрмеры, направленные на ограничение доступа к уязвимому сервису. Такие контрмеры несут за собой реконфигурацию целевой сети, в связи с этим их приоритет ниже, чем у контрмер, описанных в пункте 2.1. Применение контрмер данного типа напрямую зависит от необходимости удаленного доступа к сервисам из внешней или локальной сети. Исходя из данной характеристики можно выделить несколько типовых ситуаций:

К уязвимому сервису узла сети необходим удаленный доступ из внешней сети. В данном случае оптимальным решением задачи обеспечения безопасности целевой сети является перемещение данного сервиса в отдельный сегмент сети, т.е. в демилитаризованную зону (DMZ). Так же было бы целесообразным запретить доступ из DMZ в другие сегменты сети, если в этом нет необходимости.

К уязвимому сервису узла сети необходим удаленный доступ внутри локальной сети. В данной ситуации наилучшим решением проблемы безопасности сети является запрет доступа к уязвимому сервису хоста, если в этом нет явной необходимости.

К уязвимому сервису узла сети отсутствует необходимость удаленного доступа. В такой ситуации необходимо просто ограничить доступ к уязвимому сервису.

2.3. Контрмеры, направленные на повышение уровня защищенности вычислительной сети в целом. Мы предлагаем реализацию контрмеры данного направления в виде системы предотвращения атак (IPS). Данный метод является очень эффективным и позволяет тщательно настроить систему на противодействие конкретным типам атак.

Произведя анализ расположения уязвимостей в сети можно определить места размещения сенсоров IPS, которые будут максимально эффективно детектировать атаки. Это обеспечит оптимальную работу системы, что позволит решить проблему безопасности, когда иные способы оказываются бессильны.

В тоже время данный метод сложен в реализации и настройке влечет за собой изменения в структуре вычислительной сети. Этим обусловлен самый низкий приоритет данного типа контрмер.

3. Алгоритм расчета контрмер для устранения уязвимостей в защищаемой сети

На основе предложенного нами метода определения потенциально опасных узлов сети, описанного в разделе 1, и рассмотренных нами направлений контрмер в разделе 2, мы предлагаем использовать следующий алгоритм выработки контрмер, направленных на устранение конкретной уязвимости в защищаемой сети:

1. Выполнить поиск альтернативной версии сервиса, в которой отсутствует уязвимость, и если такая версия найдена предложить использовать ее;

2. Иначе выполнить поиск альтернативного сервиса, в котором отсутствует уязвимость, и если такая альтернатива найдена предложить использовать ее;

3. Иначе, если эксплуатация данного сервиса возможна только после эксплуатации другой уязвимости на данном узле сети, то предложить переместить данный сервис на отдельный сетевой узел;

4. Иначе проверить используется ли данный сервис, если это неиспользуемый сервис, то предложить удалить его;

5. Иначе проверить необходимость доступа к данному сервису из внешних сетей, если необходимость есть, то предложить переместить данный сервис в DMZ, и оптимально ограничить доступ из DMZ в другие сегменты ЛВС.

6. Иначе проверить необходимость доступа к данному сервису из локальной сети, если необходимость есть предложить блокировать весь доступ который не является необходимым.

7. Иначе предложить полностью блокировать доступ к данному сервису.

8. Иначе предложить использовать в ЛВС систему предотвращения атак.

Мы предлагаем реализовать автомтизированную систему выработки контрмер для устранения уязвимостей в защищаемой сети, которая будет составлять список рекомендаций для повышения уровня безопасности целевой сети, применяя предложенный нами алгоритм к каждой уязвимоти.

Заключение

В данной статье были описаны идеи для поиска наиболее уязвимых узлов целевой сети. Помимо этого нами были проанализированны основные направления разработки контрмер для обеспечения безопасности защищаемой сети. Ключевой частью данной статьи является предложенный нами алгоритм выработки контрмер. Развитием исследований данного направления станет практическая реализация автоматизированной системы расчета контрмер для устранения уязвимостей в защищаемой сети.

Список литературы

1. Абрамов Е.С., Кобилев М.А., Крамаров Л.С., Мордвин Д.В. Использование графа атак для автоматизированного расчета мер противодействия угрозам информационной безопасности сети // Известия Южного федерального университета. Технические науки. – 2014. – № 2 (151). – С. 92-100.

2. Mell P., Scarfone K., Romanosky S. Common Vulnerability Scoring System (CVSS). <http://www.first.org/cvss/cvss-guide.html>.

Секция «Инженерные инновации в текстильной и легкой промышленности», научный руководитель – Черунова И.В., д-р техн. наук, профессор

ИССЛЕДОВАНИЕ ФОРМОУСТОЙЧИВОСТИ ОБЪЁМНЫХ ДЕТАЛЕЙ ШВЕЙНЫХ ИЗДЕЛИЙ

[1]Ташпулатов С.Ш., [1]Бабаева Г.И., [2]Нутфуллаева Л.Н., [3]Черунова И.В.

[1]Ташкентский институт текстильной и легкой промышленности, Ташкент, e-mail: el_s@list.ru; [2]Бухарский инженерно-технологический институт, Бухара; [3]Институт сферы обслуживания и предпринимательства, филиал Донского государственного технического университета, Шахты

Оценка формоустойчивости деталей швейных изделий остается сложной задачей. В работе [1] предложен принцип косвенной оценки формоустойчивости по показателям жёсткости, упругости, усадки, несминаемости и др.

Показатели деформации изгиба (жесткость и упругость) являются одними из основных свойств материалов, определяющих формоустойчивость швейных изделий. Стандартные методы определения жёсткости и упругости позволяют получить статические показатели свойств материалов, не раскрывая динамики процессов их деформирования.

Оценка формуемости и формоустойчивости по методике, включающей плоскостное и пространственное деформирование материалов под воздействием термовлажностных факторов и давления исследовались в работе [2]. Исследование устойчивости формы деталей швейных изделий осуществляют, как правило, экспериментально, путем моделирования процесса формообразования и формования, в зависимости от формы деталей, на плоских или объёмных образцах. Вопрос, как себя поведет под сосредоточенной нагрузкой пакет «материал+полимерный композит», состоящий из нетканого полотна с нанесенным на него полимерным покрытием в виде сетки армирования, расположенной под различными углами ориентации, является основным в данной работе.

Верхнюю часть цельноформованного головного убора приближенно можно представить в виде заданного сферического сегмента. Следовательно, оценка формоустойчивости также может быть выполнена посредством расчета величины деформирующей нагрузки, под воздействием которой изменяется предварительная заданная форма.

Рассмотрим гипотезу о том, что значение деформирующей сосредоточенной нагрузки P зависит от физико-механических свойств образцов пакетов детали и может служить одним из косвенных показателей, характеризующих его формоустойчивость.

Для определения величины деформирующей нагрузки для объемной формы сегмента $P_{деф}$ был использован энергетический метод [1].

Учитывая [1], значение деформирующей нагрузки $P_{деф}$ можно определить из равенства $U = A$

$$P = \frac{\frac{\pi E \delta^3}{12 R_1^2 (1-\nu^2)}(9H_1^2 - 72H_1H_2 + 272H_2^2) - 2\pi z_1 (\frac{3}{20}R_1^2 H_1 - \frac{2}{5}R_1^2 H_2)}{H_1}, \quad (1)$$