Содержание

ВВЕДЕНИЕ

Внутренняя сетевая инфраструктура практически любой организации представляет собой сложную структуру, состоящую из множества различных сервисов, направленных на поддержание функционирования компании. Эта структура очень динамична: добавляются новые сервисы, меняются конфигурации существующих, создаются новые связи между сервисами. В процессе роста (жизни) системы процесс обеспечения её информационной безопасности и защита критически важных объектов становятся нетривиальной задачей (все более сложной задачей).

Причиной нарушения информационной безопасности чаще всего становятся:

- уязвимости в операционных системах;
- уязвимости приложений, осуществляющих сетевое взаимодействие с пользователем или друг с другом;
- неправильная конфигурация программного обеспечения;
- ошибки контроля доступа

Используя имеющиеся уязвимости и недостатки системы внешние и внутренние нарушители проводят сетевые атаки, приводящие к компрометации различных узлов и реализации различных угроз информационной безопасности сети.

Для выявления недостатков компонентов системы, а также поиска уязвимостей и потенциальных векторов атак на информационные ресурсы, проводится анализ защищенности сети. Одним из наиболее эффективных методов анализа является тестирование на проникновение, в ходе которого осуществляется моделирование атак реальных злоумышленников. Такой подход позволяет в полной мере провести оценку защищенности сетевой инфраструктуры, оценить существующие и предложить новые способы защиты.

При этом все возрастающая сложность компьютерных систем: количество узлов в сети, множество различных версий сервисов и потенциальных уязвимостей и наличие средств защиты обуславливает необходимость в разработке автоматизированных систем анализа защищенности сети. Такие системы должны не только проводить анализ сети на наличие уязвимостей, построение трасс атак и их реализацию (моделирование), но также предлагать оптимальные пути их исправления.

В больших компаниях существует практика закрытия только тех уязвимостей, что приводят к проникновению в сеть и компрометации наиболее критических узлов. Из-за сложности сетевой инфраструктуры часто закрывают глаза на остальные цепочки уязвимостей, которые все ещё остаются в локальной сети организации после проведения анализа защищенности, что приводит к повторной компрометации сети при нахождении новой точки входа.

Необходимо не только защитить сеть от проникновения извне, но и обеспечить должный уровень защищенности внутренней сети. Так, по данным Positive Technologies за 2019 год [https://www.ptsecurity.com/ru-ru/about/news/pentestery-preodoleli-setevoj-perimetr-92-procenta-kompanij/] при проведении внешнего тестирования на проникновение экспертам удалось преодолеть сетевой периметр 92% организаций, тогда как от лица внутреннего нарушителя был получен полный контроль над инфраструктурой во всех исследуемых системах.

Для оценки уровня защищенности системы в данной работе предлагается использовать подход, основанный на анализе графа потенциальных атак с составлением метрик защищенности узлов и сети в целом для определения наиболее эффективных защитных мер, которые приведут к максимальному увеличению уровня защищенности сети с наименьшими усилиями. Для построения графа атак предлагается использовать различные базы уязвимостей и средства автоматизации этапов сканирования узлов и эксплуатации уязвимостей.

Добавлено примечание ([z1]): Может впихнуть сюда итерированность метода. То есть получение максимального профита за 1 итерацию – закрытие 1 уязвимости.

Цель данной работы – разработать автоматизированное средство оценки уровня защищенности сетевой инфраструктуры.

Цель данной работы – разработать автоматизированное средство выработки рекомендаций по повышению уровня защищенности сетевой инфраструктуры.

Для достижения поставленной цели определены следующие задачи:

- Описать основные классы уязвимостей, приводящие к компрометации узлов и методы их поиска; / Описать концепцию тестирования на проникновение и основные методы его автоматизации;
- Проанализировать применимость графов атак в задачах оценки защищенности сети;
- Разработать метод поиска оптимальных защитных мер на основе анализа графа атак;
- Разработать автоматизированную систему построения графа атак и анализа защищенности сети;
- Провести экспериментальную оценку эффективности разработанного метода.

1. Проблема сетевых атак
    1.1. Уязвимости
    1.2. Виды уязвимостей, приводящих к проникновению на узел
    1.3. Эксплоиты
    1.4. Методы сканирования узлов
    1.5. Основные способы эксплуатации уязвимостей
2. Графы атак в задачах оценки защищенности сети
    2.1. Описание графа атак
    2.2. Способы построения графов атак
        2.2.1. Ручное построение графа
        2.2.2. Автоматизация построения графа (metasploit + nmap, Ш.1)
    2.3. Анализ графа атак (Ш. 2-3)
        2.3.1. Методики оценки защищенности сети
            2.3.1.1. Метрики защищенности узлов (CVSS, Научные работы(есть ссылки)
        2.3.2. Методики выбора защитных мер
            2.3.2.1. –
            2.3.2.2. –
            2.3.2.3. Недостаточность существующих методов
        2.3.3. Метод выбора контрмер на основе удаления узла, максимизирующего риски (СВОЙ МЕТОД)
        2.3.4. Оптимизация процесса выбора наиболее критического узла (Ш.4)
3. Разработка системы автоматизированного тестирования на проникновение и итерированного выбора контрмер
    3.1. Архитектура системы
    3.2. Оценка сложности системы до оптимизации
    3.3. Оценка сложности системы после оптимизации
4. Тестирование разработанной системы

Добавлено примечание ([ПW2]): В работах Котенко и из практики закрытия дыр после пентестов

Добавлено примечание ([ПW3]): Здесь как раз из актуальности, что часто закрывают вход и всё.

Добавлено примечание ([ПW4]): Может оставить просто оценку сложности системы, либо перенести оценку сложности в 2.3.3 и 2.3.4

4.1. Оценка влияния введенного метода выбора контрмер на безопасность системы

Ш.1.

В процессе автоматизированного тестирования на проникновение логируем все достижимые узлы, способ проникновения и подсчитанный уровень уязвимости узла, для чего используем различные метрики (cvss, есть научная работа по метрикам уязвимости узлов)

Ш.2.

После выполнения Ш.1. строим граф, состоящий из узлов-устройств (D), узлов-уязвимостей (V) и ребер E, где E – подмножество V x D (данном графе не будет существовать ребер типа D x D). (Возможно стоит рассмотреть ребра V x V -> переход от уязвимости к уязвимости).

Считаем угрозу, создаваемую каждым узлом:

$a_i = \sum d_j$, где d_j – уязвимость устройств j, достижимых из i.

Ш.3.

Считаем уязвимость всей исследуемой системы:

$$M = \sum a_i$$

Ш.4.

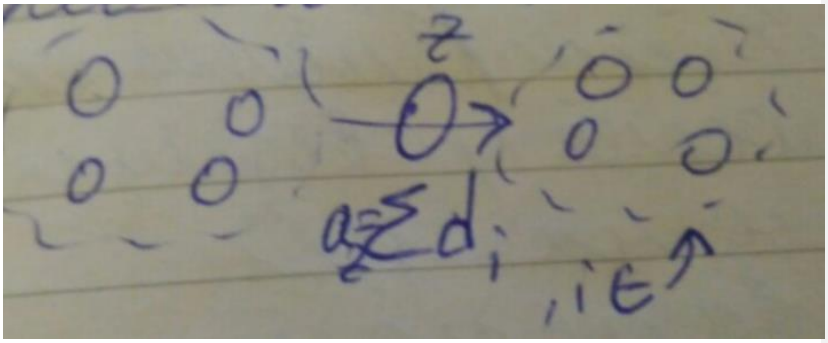
Анализируем существующий граф с целью нахождения такого узла-уязвимости (V), при удалении которого значение M максимально уменьшится.

Решение задачи без оптимизаций:

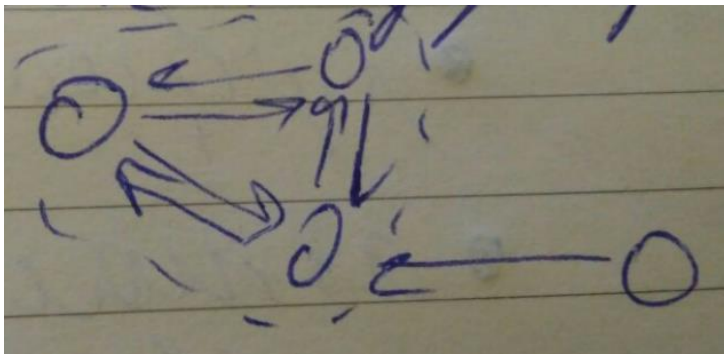
По очереди удаляем каждый из узлов-уязвимостей и пересчитываем угрозу попадания на каждый из узлов a и уязвимость всей системы M. Ищем такой узел, при котором изменение уязвимости системы максимально.

Оптимизации:

- Для мостов (единственный путь между двумя подграфами) нет необходимости пересчитывать пути.



- Значения уязвимости всех «листов» графа можно просуммировать с родителями до тех пор, пока не натыкаемся на узел с несколькими дочерними объектами (считаем рекурсивно, проблема возникает только при подходе к полным графам (там куча циклов, по идее не сможем просчитать однозначно угрозу маршрута через них);
- Находим полные подграфы и считаем для них $\sum d_i$, так как при попадании в любой из узлов такого подграфа мы гарантированно попадаем на все остальные (или решаем задачу нахождения максимально длинного цикла!, попав на экземпляр цикла гарантированно достигаем всех элементов)



1. ПРОБЛЕМА СЕТЕВЫХ АТАК

Под сетевой атакой можно понимать любой процесс, нацеленный на компрометацию безопасности сети. Результатом выполнения сетевой атаки может стать: захват контроля над удаленной/локальной вычислительной системой, либо её дестабилизация, повышение прав в системе, получение данных пользователей, пользующихся данной вычислительной системой.

Видов сетевых атак огромное множество и часть из них в состоянии реализовать человек, не владеющий большим запасом знаний в области информационной безопасности (далее ИБ).

(ТУТ КАКАЯ-НИБУДЬ СТАТИСТИКА ПО СЕТЕВЫМ АТАКАМ)

1.1. Уязвимости

В области информационной безопасности уязвимостью считается любой недостаток системы, используя который злоумышленник может провести атаку на систему. Уязвимость может быть результатом ошибок, допущенных на этапе программирования или проектирования отдельной программы, протокола или запроса, слабых паролей или ненадежных политик доступа.

Наиболее частой причиной возникновения уязвимостей становятся:

- ошибки проектирования, разработки программного продукта, протокола или запроса;
- слабые пароли;
- намеренно оставленные лазейки;
- неправильные настройки оборудования;
- отсутствие надежных политик доступа;
- несанкционированные неумышленные действия пользователей;

В наиболее широком смысле уязвимости можно разделить на две группы: известные и 0-day эксплойты. Известные уязвимости хорошо задокументированы исследователями, а соответствующие программные продукты имеют патчи, устраняющие возможность эксплуатации данных уязвимостей.

Под угрозой [https://bdu.fstec.ru/ubi/terms] в ИБ понимается любая потенциальная возможность тем или иным образом нарушить информационную безопасность. Попытка реализации такой угрозы посредством уязвимости и называется атакой.

В начале 90-х годов стало очевидно, что для хранения всего объема записей о найденных уязвимостях необходимо провести их классификацию и систематизацию. Актуальность данной задачи обуславливалась появлением большого числа нового программного обеспечения: операционных систем, программ, платформ разработки; увеличением количества версий программных продуктов, а также частотой нахождения уязвимостей.

Каждой обнаруженной уязвимости требовалось присвоить некий идентификатор и дать ей краткое описание. Также важно было определить критичность данной уязвимости, например, по таким критериям как простота эксплуатации и последствия эксплуатации. Данная информации впоследствии могла быть дополнена рекомендациями по устранению уязвимости, а также информацией об уязвимых версиях продукта.

В наше время данные об уязвимости различного программного обеспечения вовсю используются злоумышленниками для совершения атак.

1.1.1 *Базы уязвимостей*

На текущий момент существует несколько широко распространенных баз уязвимостей[https://safe-surf.ru/specialists/article/5228/607311/]:

1. Реестр уязвимостей БДУ ФСТЭК России;
2. MITRE CVE и база данных NVD;
3. OSVDB;
4. Secunia Advisory and Vulnerability Database;
5. VND от CERT/CC;
6. Exploit Database;

Добавлено примечание ([z5]): Расширить описанием каждой из баз? Например потом можно будет сделать ссылку на CVE, как на основу в metasploit и т.п.

1.1.2 *Классификация уязвимостей по ГОСТ Р 56546-2015*

https://files.stroyinf.ru/Data2/1/4293759/4293759791.pdf

1.1.3 *Уязвимости приводящие к проникновению на узел*

Для проникновения во внутреннюю сеть организации, злоумышленнику необходимо найти узел данной сети, доступный из внешней сети, после чего он должен получить доступ к этому узлу и иметь возможность выполнить на нем произвольные команды. Уязвимости, способствующие проникновению на узел можно поделить на следующие типы [https://xakep.ru/2017/04/10/hacking-attack-types/]:

1. недостатки управления учетными записями и паролями;
2. уязвимости веб-приложений;
3. недостатки фильтрации трафика;
4. недостатки управления уязвимостями и обновлениями;
5. плохая осведомленность пользователей в вопросах информационной безопасности;
6. недостатки конфигурации и разграничения доступа;

Любая уязвимость каждого из этих типов способствует проникновению на удаленное устройство. Иногда для проникновения используются комбинации данных методов, однако это лишь повышает сложность атаки, но не увеличивает вероятность проникновения.

Добавлено примечание ([z6]): Сюда же входят существующие эксплоиты

1.2. Эксплойты

Эксплойт [https://www.gizmosphere.org/network-security-vulnerabilities-vs-exploits/] – скрипт или программа, разработанная исключительно для эксплуатации определенной уязвимости. В процессе проведения сетевой атаки могут задействоваться одновременно несколько эксплойтов для достижения поставленных целей. При этом необходимо успешное выполнение каждого из эксплойтов. Закрытие любой из уязвимостей подобной цепочки приведет к несостоятельности исходной атаки.

Эксплойтом может также являться обычный текст или словестное описание того, как проэксплуатировать уязвимость. Чаще всего для написания эксплойтов используются следующие языки программирования: Python, Ruby, PHP, Perl, HTML, Javascript, C/C++.

В общем случае определяют 2 вида эксплойтов:

- удаленный – эксплуатирует уязвимость удаленной системы без предварительного доступа к ней;
- локальный – эксплуатирует уязвимость локальной системы. Требует наличие предварительного доступа к целевой системе. Чаще всего используется для повышения привилегий.

Так как эксплойты разрабатываются для выполнения различных действий на атакуемой системе, то они могут классифицироваться по объекту назначения [https://www.anti-malware.ru/threats/exploits]:

- для браузеров и дополнений к ним;
- для операционных систем;
- для офисных программ, проигрывателей и другого прикладного программного обеспечения;
- для серверного программного обеспечения;
- для веб-сервисов, например, WordPress, Joomla, Drupal и др.
- для аппаратных компонентов.

Добавлено примечание ([u7]): Объект назначения зависит от уязвимости => имеет ли смысл классифицировать эксплойты таким образом, если принадлежность классу целиком зависит от уязвимости?

Большинство существующих эксплойтов входят в ту или иную базу эксплойтов, которые в большинстве случаев создаются для научных целей.

1.2.1 *Базы эксплойтов*

Наиболее обширные базы эксплойтов на данный момент [https://null-byte.wonderhowto.com/how-to/top-10-exploit-databases-for-finding-vulnerabilities-0189314/]:

1.3. Сценарий сетевой атаки

Реализация угрозы, как было описано выше, может быть направлена на нарушение целостности, доступности и конфиденциальности информации. Процесс реализации угрозы в общем случае состоит из 4 этапов [https://cyberleninka.ru/article/v/klassifikatsiya-ugroz-i-uyazvimostey-informatsionnoy-bezopasnosti-v-korporativnyh-sistemah]:

- Сбор информации о цели;
- Проникновение в целевую систему;
- Осуществление несанкционированных действий;
- Сокрытие следов несанкционированного доступа.

На каждом из этих этапов может быть задействовано неограниченное число дополнительных операций, представленных на рисунке X.

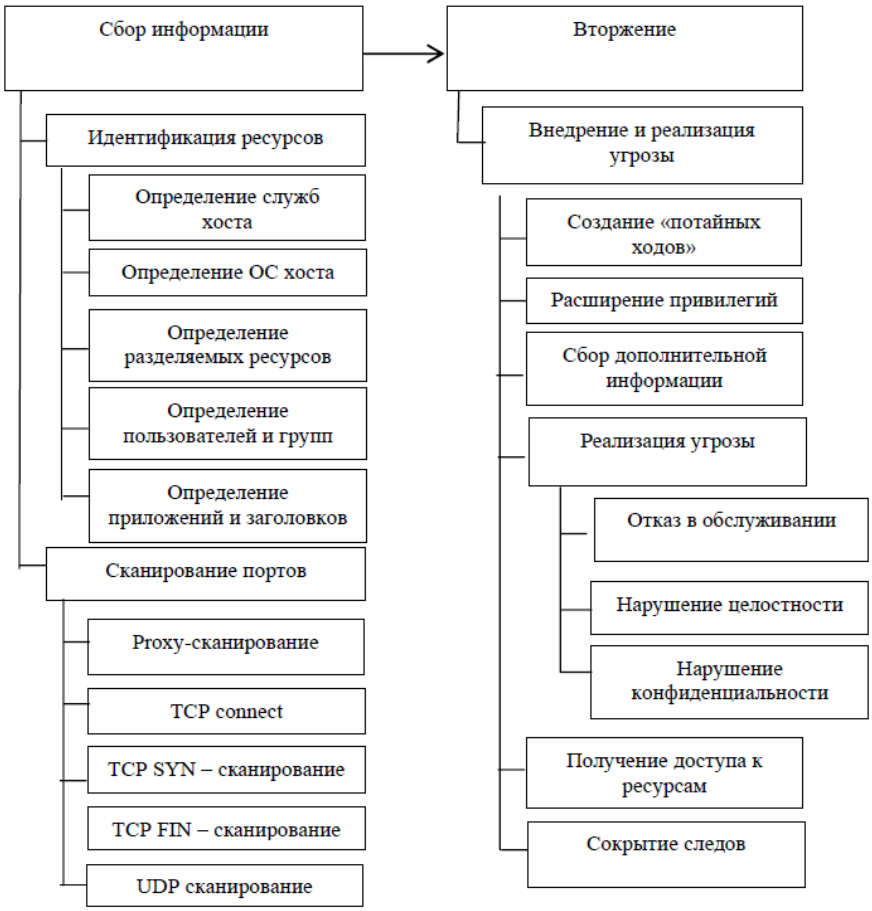


Рисунок X – Процесс реализации угрозы

1.3.1*Сбор информации*

2. ОЦЕНКА ЗАЩИЩЕННОСТИ И ВЫБОР МЕР ЗАЩИТЫ В СОВРЕМЕННЫХ КОМПЬЮТЕРНЫХ СЕТЯХ

Задача оценки защищенности и выбора защитных мер в компьютерных сетях является одной из основных задач информационной безопасности. Согласно

2.1. Нормативная база задачи оценки защищенности сетевой инфраструктуры

2.1.1 *Стандарты в области оценки безопасности компонентов сетевой инфраструктуры*

2.1. Методики оценки защищенности компьютерной сети

Оценка защищенности компьютерной сети – это процесс выявления уязвимостей, угроз и рисков, связанных с активами организации и мер защиты, которые могут смягчить эти риски [https://www.tcdi.com/criteria-for-selecting-an-information-security-risk-assessment-methodology-qualitative-quantitative-or-mixed/]. Существует два базовых подхода к оценке защищенности: качественный и количественный.

Методики оценки защищенности позволяют определить понятие показателя защищенности – параметра, определяющего качественную или количественную оценку защищенности анализируемой сети.

2.1.1 *Качественные методики оценки защищенности*

Качественные методики позволяют идентифицировать уязвимости и угрозы, описать причины их возникновения, возможные последствия и применяемые защитные меры и на их основе ранжировать риски, однако такие методики не позволяют определить численную величину риска.

Методика оценки риска называется качественной, если в процессе её выполнения формируется качественная оценка уровня риска. И.А. Педерсен и Н.Е. Брюковецкая определили несколько этапов качественной оценки риска (рисунок X) [https://cyberleninka.ru/article/n/metodologiya-otsenki-riskov-predpriyatiya/viewer]

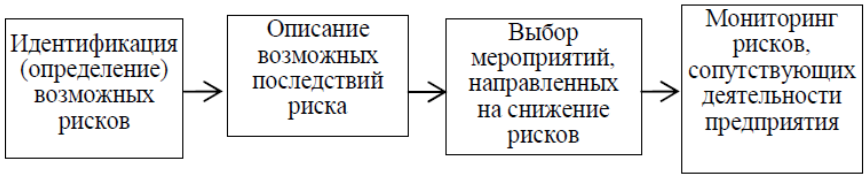


Рисунок X – Последовательность проведения качественного анализа рисков

Качественные методики используют опыт и суждения экспертов, а не математические формулы. Они также могут использовать опросы для определения уровня угрозы и ожидаемых рисков. Данные методики особенно эффективны, когда невозможно определить денежный эквивалент конкретного риска.

Качественные методики не требуют больших математических вычислений, но результаты, как правило, менее точны, чем полученные при количественной оценке.

Примеры качественных методик:

- COBRA [Visintine, V. Global information assurance certification paper];
- OCTAVE [Caralli, R. A. Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process];
- FRAP [Peltier, T. R. Information security risk analysis, Third Edition].

2.1.1.1 Методика COBRA

Методика COBRA разработана компанией Risk Associates [https://www.securityauditor.net/]. COBRA представляет собой процесс анализа рисков на основе экспертных систем, использующих обширную базу знаний по угрозам, уязвимостям и множеству опросников. Данная методика позволяет

оценить соответствие оцениваемой системы стандарту ISO 17799 [ISO/IEC 17799:2005. Information technology. – Security techniques].

COBRA включает в себя 2 модуля: COBRA Policy Compliance Analyst и COBRA Data Protection Consultant. Первый позволяет определить, удовлетворяет ли оцениваемая система стандарту ISO 17799 и получить рекомендации. Второй содержит вопросы, позволяющие идентифицировать активы, угрозы, уязвимости и средства защиты.

Результат работы данной методики – отчёты, содержащие оценки рисков и рекомендации по их уменьшению, базирующиеся на общепринятых практиках.

2.1.1.2 Методика OCTAVE

Данная методика была создана для решения проблем информационной безопасности США. OCTAVE направлена на выявление, оценку и управление рисками информационной безопасности.

Эта методика помогает организации:

- определить наиболее важные для организации активы;
- выявить уязвимости и угрозы для этих активов (автоматизированное или ручное сканирование инфраструктуры);
- определить и оценить потенциальные последствия для организации в случае реализации угроз;
- инициировать действия по постоянному улучшению средств защиты для снижения рисков.

Данная методика основана на применении экспертных знаний.

2.1.1.3 Методика FRAP

Методика оценки рисков FRAP разработана Томасом Пелтиером. В методике обеспечение информационной безопасности предлагается рассматривать в рамках процесса управления рисками. Управление рисками в

сфере информационной безопасности — процесс, позволяющий компаниям найти баланс между затратами средств и сил на средства защиты и получаемым эффектом.

Основные этапы оценки рисков по FRAP:

1. Определение защищаемых активов производится с использованием опросных листов, изучения документации, использования инструментов автоматизированного анализа сетей;
2. Идентификация угроз. При составлении списка угроз могут использоваться разные подходы:
   - заранее подготовленные экспертами перечни угроз, из которых выбираются актуальные для данной сети;
   - анализ статистики происшествий в данной сети и в подобных ей — оценивается частота их возникновения;
   - «мозговой штурм», проводимый сотрудниками компании.
3. Когда список угроз закончен, каждой из них сопоставляют вероятность возникновения. После чего оценивают ущерб, который может быть нанесен данной угрозой. Исходя из полученных значений, оценивается уровень угрозы.

При проведении анализа, как правило, принимают, что на начальном этапе отсутствуют средства и механизмы защиты. Таким образом оценивается уровень риска для незащищенной сети, что в последствии позволяет показать эффект от внедрения средств защиты информации.

Оценка производится для вероятности возникновения угрозы и ущерба от нее по следующим шкалам:

- Высокая – высокая вероятность того, что угроза реализуется в течение года;
- Средняя – угроза может быть реализована в течение следующего года;

- Низкая – низкая вероятность того, что угроза будет реализована в течение следующего года.

Ущерб активу определяется схожим образом (высокий/средний/низкий).

4. После определения вероятности возникновения ущерба и величины ущерба эксперты определяют средства, позволяющие снизить риски, ориентируясь на наиболее рентабельные. Данные о рисках и возможных защитных мерах документируются и передаются управляющему лицу.

5. Документирование. Результат работы по FRAP – полный набор документации по угрозам и рискам рассматриваемой системы, а также возможные средства минимизации уровней риска угроз.

2.1.2*Количественные*

Количественные методики описывают возможные риски в денежном или частотном эквиваленте. На основе полученных значений и стоимости реализации мер защиты риски сравниваются для принятия оптимальных мер защиты. При количественном анализе рисков выделяют несколько последовательных этапов (рисунок X) [https://cyberleninka.ru/article/n/metodologiya-otsenki-riskov-predpriyatiya/viewer].

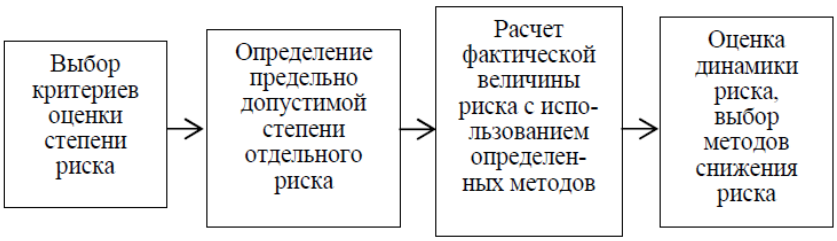


Рисунок X – Последовательность проведения количественного анализа рисков

К количественным методикам относят:

- RiskWatch;
-

2.1.3*Смешанные*

На практике обособленное применение количественных или качественных методик. Смешанные методики сопоставляют качественному уровню определенный количественный диапазон [Астахов, А. Искусство управления информационными рисками].

К таким методикам относятся:

- CRAMM;
- Методика оценки рисков на основе графов атак.

2.1.3.1 Смешанная методика CRAMM

Методика CRAMM [https://managementmania.com/en/cramm-ccta-risk-analysis-and-management-method] объединяет в себе количественные и качественные методы для проведения комплексной оценки рисков. Разработана Central Computer and Telecommunications Agency (CCTA) в Великобритании. Данная методика включает в себя следующие этапы:

- Выявление и оценка активов. В качестве активов могут выступать: программное обеспечение, аппаратные ресурсы, либо данные;
- Выявление угроз и уязвимостей;
- Оценка рисков;
- Выбор защитных мер и их приоритезация.

Для оценки потенциального ущерба используется шкала от 1 до 10. Выявление угроз и уязвимостей осуществляется на основе экспертных знаний с помощью опросов. Значение уровня уязвимостей определяется как: высокий, средний и низкий, уровень угроз: очень высокий, высокий, средний, низкий и очень низкий. Риск оценивается в зависимости от годовых потерь по шкале от 1 до 7. Потери зависят от стоимости активов, уровня угрозы и уязвимости. Полученные уровни рисков используются для генерации вариантов защитных мер.

Таким образом, CRAMM комплексно охватывает все этапы управления рисками, начиная от фактического анализа рисков и заканчивая предложением контрмер, включая генерацию выходных данных для документации по безопасности (планирование действий в чрезвычайных ситуациях и обеспечение непрерывности). CRAMM одновременно поддерживается одноименным приложением, которое помогает в сборе данных, а также в расчете и обработке отчета по управлению рисками.

CRAMM также помогает доказать эффективность затрат на управление рисками, безопасность и планирование действий в чрезвычайных ситуациях. Он содержит обширную библиотеку контрмер безопасности. Также применение методики CRAMM позволяет организациям подготовиться к сертификации в соответствии с ISO 27001.

Графы атак являются ценнейших инструментом, позволяющим проиллюстрировать, каким образом злоумышленник может получить доступ к целевой системе. Проведя анализ графа, специалисты по информационной безопасности могут сосредоточить свою усилия на устранении уязвимостей, предоставляющих злоумышленникам максимальный доступ.

Графы отражают все возможные пути атак, кроме того они могут иллюстрировать все состояния системы с переходами из состояния в состояние в соответствии с используемыми уязвимостями.

Существует несколько типов графов атак:

1.

Рассматриваемый граф, содержащий множество вершин-узлов и множество вершин-уязвимостей является двудольным графом, так как множество вершин V разбито на два непересекающихся подмножества V1 и V2, причём всякое ребро E инцидентно вершине из V1 и вершине из V2 (то есть соединяет вершину из V1 с вершиной из V2.

5.1. Анализ существующих аналогов

В [Ingols, K. Practical attack graph generation for network defense] предлагается подход к генерации графа атак с множеством предусловий, позволяющий выявить наиболее критичные уязвимости системы (дающие

Добавлено примечание ([u8]): Вариант похож на мой. Также используется поиск узких мест. Однако не учитывается целиком защищенность сети, лишь количество узлов, достижимых от точки входа. Если сравнить такой метод в рамках моей метрики – защищенность сети, то он проиграет.

нарушителю наибольший доступ в системе). Достижимость хоста для нарушителя определяется наличием администраторского или гостевого доступа. Авторы формируют рекомендации по реализации защитных мер на основе показателей, определяющих как много путей атаки будет заблокировано введением защитной меры. Недостаток: в работе не описаны конкретные показатели и не предложено методик их вычисления и методик определения уровня защищенности.

В итоге они анализируют, каким образом можно попасть на выбранный узел и, соответственно, какие нужно закрыть уязвимости, чтобы этого избежать. Причём уззлы они группируют в мой аналог компонент сильной связнности и называют его prerequisite node. Для выбора лучшего решения они "взвешивают" все рекомендации по числу закрытых для доступа хакера хостов (почему они тогда просто не обрывают начальные ребра от точки входа?).

6. РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПОСТРОЕНИЯ И АНАЛИЗА ГРАФА ПОТЕНЦИАЛЬНЫХ АТАК

Для применения описанного выше метода анализа в первую очередь необходимо определить способ хранения графа. В данной работе предлагается использовать …

Каждая точка принадлежит определенному подграфу графа потенциальных атак. Имея полный набор подграфов, достижимых из текущей точки легко посчитать уровень угрозы этой точки. Более того, уровень угроз данной точки должен совпасть с уровнем угрозы любого узла, входящего в тот же подграф, если точки находятся в одном цикле. Таким образом, анализ графа должен проходить в несколько этапов:

- Определение подграфов графа потенциальных атак, уровень подсчет уровня угрозы в которых может быть оптимизирован (должен быть критерий, согласно которому найденные подграфы нужно относить к «обособленным». Сюда точно будут входить компоненты связности (точно компоненты сильной связности), любые циклы)
- Подсчет угрозы каждого узла
- Подсчет текущего уровня угрозы сети
- Удаление одного из узлов-уязвимостей -> пересчет угроз узлов и сети

7. ИСТОЧНИКИ ИНФОРМАЦИИ

https://bdu.fstec.ru/ubi/terms

8. КРАТКОЕ ОПИСАНИЕ РАЗРАБАТЫВАЕМОЙ СИСТЕМЫ

Краткое изложение сути работы

В процессе проведения тестирования на проникновение производится построение графа атак.

G = (V, A) – ориентированный граф, где V – непустое множество узлов, A – множество триплетов, называемых ребрами, вида (r_i, r_j, c_j), где $r_i, r_j \in V, c_j \in CVE$ – идентификатор уязвимости, присутствующей на узле r_j . Направление дуги задаётся последовательностью следования узлов в записи.

$$t_{r_i} = \sum_{j \in V_i \subset V} Criticality(r_j) \qquad (*)$$

Где t_i – угроза попадания злоумышленником на узел $r_i \in V$, r_j – узел графа, достижимый из r_i, $Criticality(r_j)$ – критичность захвата узла r_j, V_i – подмножество достижимых узлов из r_i.

Узел r_j достижим из $r_i <=> \exists$ хотя бы 1 путь из r_i в r_j, $r_i \neq r_j$.

$$Criticality(r_j) = device\_type\_coef(r_j) \sum service\_cost(service\_name_{i_j})$$

Где $device\_type\_coef(r_j)$ – коэффициент зависящий от типа устройства. Можно использовать базу nmap классификации устройств (28 штук) и сопоставить каждому коэффициент по личным соображениям. Например, от 1 до 2. $service\_cost()$ – критичность одного сервиса. Каждому сервису или типу сервисов можно сопоставить конкретную критичность компроментации, $service\_name_{i_j} - j_{\text{ый}}$ сервис узла r_i.

Угроза компрометации системы на k-ом шаге поиска контрмер рассчитывается следующим образом:

$$T_k = \sum_{i=0}^{|V|} t_{r_i}$$

Далее необходимо найти такую уязвимость c_i, при удалении которой максимально снизится угроза компрометации системы, то есть:

$$(T_{k-1} - T_k/\{c_i\}) \to max$$

Для подсчета $\mathrm{t_{r_i}}$ необходимо в первую очередь определить множество V_i. Для этого обойти граф (можно использовать любой алгоритм обхода, пока что используется обход в ширину (DFS)), начиная от узла r_i, и пометить все достижимые узлы. Сложность алгоритма обхода $O(|V| + |A|)$. Таким образом, сложность подсчета T_k в худшем случае равняется

$$O\big((|V| + |A|) * (|V|)\big) = O(V^2 + A)$$

Данный алгоритм можно оптимизировать следующим образом. Представим, что каждый узел исходного графа принадлежит подграфу одного из двух типов – компоненте сильной связности или N-арному дереву.

Ориентированный граф называется сильно связным, если любые два его узла сильно связны. Два узла s и t любого графа сильно связны, если существует ориентированный путь из s в t и ориентированный путь из t в s. Компонентами сильной связности орграфа называются его максимальные по включению сильно связные подграфы. Областью сильной связности называется множество узлов компоненты сильной связности.

Осуществим поиск областей сильной связности (обозначим их S_i, например, с помощью алгоритма Косарайю, который использует двойной обход в глубину, следовательно его сложность O(2|V| + 2|A|).

Ориентированное N-арное дерево — ацикличный орграф (ориентированный граф, не содержащий циклов), в котором только один узел имеет нулевую степень захода (в него не ведут дуги), а все остальные узлы имеют степень захода не больше N. Узел с нулевой степенью захода называется корнем дерева, узлы с нулевой степенью исхода (из которых не исходит ни одна дуга) называются концевыми узлами или листьями. Обозначим их NT_i. Значение N может отличаться от дерева к дереву.

При этом в N-арное дерево включаются только те узлы, входящие дуги которых направлены от узлов того же дерева. Исключение может составлять только корневой узел. (доп условие)

В первую очередь производится поиск всех компонент сильной связности и соответствующих им областей сильной связности. Проиллюстрируем данный шаг на примере. Считаем, что все узлы имеют по 1 уязвимости, вследствие чего дуги не подписаны (рисунок X).

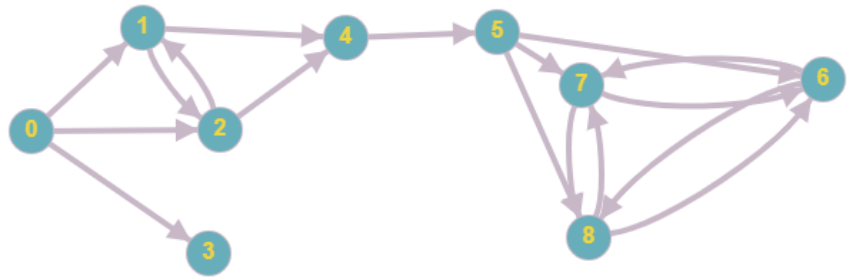


Рисунок X – Пример ориентированного графа

Обозначим на графе компоненты сильной связности и N-арные деревья (рисунок X). Компоненты сильной связности обозначены зеленым цветом, N-арные деревья – оранжевым.

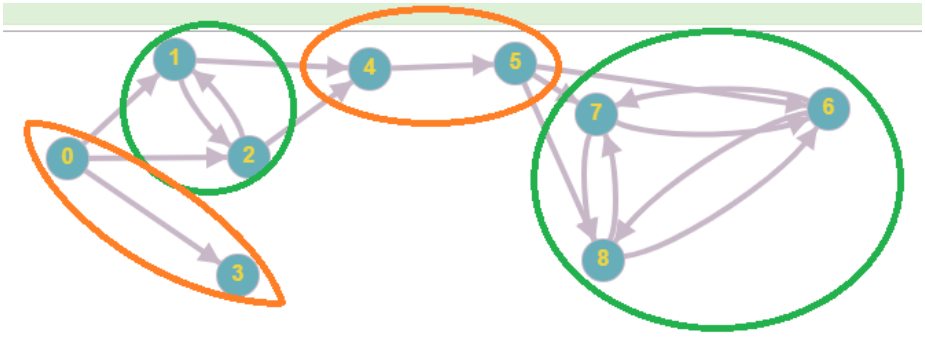


Рисунок X – Иллюстрация поиска подграфов двух типов

Рассмотрим некоторые свойства полученных подграфов:

- Из определения компоненты сильной связности следует, что в ней существуют маршруты из любого узла компоненты в любой другой. Отсюда:

$$\forall \mathrm{r}_i, \mathrm{r}_j \in S_k \colon t_{r_i} = t_{r_j}, r_i \neq r_j$$

S_k – k-ая область сильной связности

Следовательно, для каждой области сильной связности достаточно посчитать t_r единственный раз.

- Из определения компоненты сильной связности и N-арного дерева следует, что попав в любую из данных компонент гарантируется прохождение по всем узлам подграфа. При этом в N-арном дереве существует только один корневой узел, следовательно вход в данный подграф возможен исключительно через него, что позволит пройти по всем узлам данного дерева. Следовательно при достижении корневого узла такого дерева нет необходимости в пересчете критичности его узлов. Достаточно сделать это один раз:

$$S_{NT_i} = \sum_{r_j \in NT_i} Criticality(r_j)$$

S_{NT_i} – критичность захвата подграфа NT_i

- Ребра исходного графа распределяются на 3 категории:
  - Ребра, принадлежащие компонентам сильной связности;
  - Ребра, принадлежащие N-арным деревьям;
  - Ребра, связывающие различные подграфы.

В связи с этим при удалении ребер возможны 3 ситуации:

1. Если ребро находилось в компоненте сильной связности или N-арном дереве, необходимо осуществить перераспределение узлов данной компоненты по новым подграфам сильной связности и N-арным деревьям. Также необходимо осуществить новые расчёты параметров t_r и S_{NT}

2. Если ребро соединяло две компоненты, нет необходимости в пересчете параметров подграфов.

Исходя из вышеописанных свойств можно провести следующее преобразование: представим все компоненты сильной связности и N-арные деревья как узлы нового графа, которые включают в себя все входящие и все исходящие дуги исходного подграфа. При этом новое значение критичности узла высчитывается по формуле:

$$Criticality(r_i) = \sum_{r_j \in S_k} Criticality(r_j)$$

А новое значение угрозы попадания на узел r_i

$$t_{r_i} = |S_k| * t_{r_j}, \quad t_{r_j} \in S_k$$

Где S_k – k-ая область сильной связности

Полученный в результате граф, будет иметь следующий вид (рисунок X):

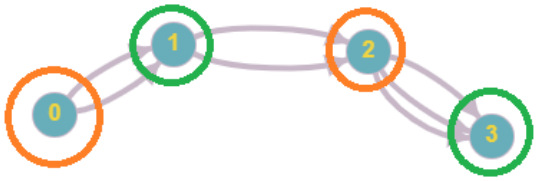


Рисунок X – Упрощенный вид графа

Таким образом, при определении значения угрозы компрометации системы T_k можно будет воспользоваться результатами расчетов, полученных в процессе вычисления угрозы компрометации системы T_{k-1}, что весьма критично, так рассматриваемая структура сетевой организации очень часто

выглядит так, что $|A| \gg |V|$. Следовательно перебор всех возможных решений займет продолжительное время.

(ОЦЕНИТЬ СЛОЖНОСТЬ)