ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
Высшая школа кибербезопасности и защиты информации
(наименование учебного подразделения)

Отчет о прохождении производственной (преддипломной) практики
(вид и тип практики)

Орел Евгений Михайлович
(Ф.И.О. обучающегося)

6 курс, 3651001/40201
(номер курса обучения и учебной группы)

10.05.01 Компьютерная безопасность
(Направление подготовки (код и наименование)

Место прохождения практики: СПбПУ, ИПММ, Высшая школа кибербезопасности и защиты информации
(указывается наименование профильной организации или наименование структурного подразделения

г. Санкт-Петербург, Политехническая ул., д. 29
ФГАОУ ВО «СПбПУ», фактический адрес)

Сроки практики: 02.09.2019 – 23.12.2019

Руководитель практики от ФГАОУ ВО «СПбПУ»:

Иванов Д.В., к.т.н., доцент
(Ф.И.О., уч.степень, должность)

Оценка:

Руководитель практики
от ФГАОУ ВО «СПбПУ»: Д.В. Иванов

Обучающийся: Е.М. Орел

Дата: 23.12.19

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Сетевая инфраструктура практически любой организации представляет собой сложную структуру, состоящую из множества различных сервисов, направленных на поддержание функционирования компании. Данная структура очень динамична: добавляются новые сервисы, меняются конфигурации существующих, создаются новые связи между сервисами. В процессе роста системы задачи обеспечения её информационной безопасности и защиты критически важных объектов становятся нетривиальными.

Причиной нарушения информационной безопасности чаще всего становятся:

- уязвимости в операционных системах;
- уязвимости приложений, осуществляющих сетевое взаимодействие с пользователем или друг с другом;
- неправильные конфигурации 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 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: большое 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 дать оценку защищенности сети, определить наиболее критические уязвимости и недостатки сети, но и предложить оптимальные пути их исправления.

В результате проведения анализа защищенности посредством тестирования на проникновение проводится описание основных обнаруженных уязвимостей, а также способов их устранения. Однако в крупных компаниях, как и в большинстве рассмотренных методик, существует практика устранения только тех уязвимостей, эксплуатация которых приводит к проникновению во внутреннюю сеть организации или компрометации наиболее критически важных узлов системы. Из-за сложности сетевой инфраструктуры устранение всех прочих уязвимостей затягивается на неопределенный срок, что создаёт опасную ситуацию, в которой злоумышленник, обнаружив новую точку входа, может воспользоваться существующими цепочками уязвимостей для компрометации сети.

Необходимо не только защитить сеть от проникновения извне, но и обеспечить должный уровень защищенности внутренней сети. Так, по данным Positive Technologies за 2019 год [1] 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 мер защиты. Рассмотренный подход позволяет оценить уровень риска системы при проникновении нарушителя на любой из узлов сетевой инфраструктуры, включенных в граф

атак, и, следовательно, снизить риск компрометации системы с любого из узлов графа.

Цель данной работы – разработать автоматизированную систему оценки уровня защищенности сетевой инфраструктуры и выбора защитных мер на основе графов атак.

Для достижения поставленной цели определены следующие задачи:

- Определить основные способы идентификации узлов сети и их уязвимостей в процессе тестирования на проникновение;
- Проанализировать применимость графов атак в задачах оценки защищенности сети и существующие методики оценки защищенности сети и выбора защитных мер;
- Разработать методики оценки защищенности сетевой инфраструктуры и выбора защитных мер на основе анализа графа атак;
- Реализовать автоматизированную систему анализа защищенности сетевой инфраструктуры на основе анализа графа атак;

1. АНАЛИЗ ОСНОВНЫХ УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ СЕТЕВОЙ ИНФРАСТРУКТУРЫ ОРГАНИЗАЦИИ

Добавлено примечани

Сетевая инфраструктура представляет собой совокупность различного оборудования, а также программного обеспечения, которая формирует особую среду для эффективного процесса обмена данными, а также для работы бизнес-приложений. Любая составляющая сетевой инфраструктуры может стать причиной компрометации безопасности сети посредством проведения сетевой атаки или цепочки атак.

Под сетевой атакой принято 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). На рис.1.1 представлена статистика по различным сетевым атакам за декабрь 2019 [27].

Top - Network attacks IN THE LAST MONTH		
1	Bruteforce.Generic.Rdp.d	13.28%
2	Intrusion.Win.MS17-010.o	12.8%
3	Intrusion.Win.MS17-010.p	3.52%
4	Bruteforce.Generic.Rdp.a	2.98%
5	Bruteforce.Generic.RDP	0.33%
6	Intrusion.Win.NETAPI.buffer-overflow.exploit	0.14%
7	Bruteforce.Generic.Rdp.c	0.11%
8	Intrusion.Win.CVE-2017-0147.sa.leak	0.06%
9	DoS.Generic.SYNFlood	0.04%
10	Intrusion.Win.EternalRomance.s	0.02%

Рис.1.1. Статистика по проведенным сетевым атакам за последний месяц, собранная в Лаборатории Касперского

Наиболее часто встречаются атаки типа подбора пароля к сервису удаленного рабочего стола, а также атаки на SMB-сервер, использующие эксплойты, которые применялись при проведении атак типа WannaCry и ExPetr.

Для осуществления подобных атак необходимо наличие уязвимостей, эксплуатация которых в свою очередь позволит атакующему получить полный доступ к устройству жертвы от ее имени.

1.1. Уязвимости сетевой инфраструктуры

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 уязвимости, так называемые «уязвимости нулевого дня». Известные уязвимости хорошо задокументированы исследователями, а соответствующие программные продукты имеют патчи, устраняющие возможность эксплуатации данных уязвимостей. Термин 0-day обозначает не устраненные уязвимости, против которых ещё не разработаны защитные механизмы.

Под угрозой [28] в ИБ понимается любая потенциальная возможность тем или иным образом нарушить информационную безопасность. Попытка реализации такой угрозы посредством уязвимости 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 настоящее время для обозначения критичности уязвимости чаще всего используется стандарт Common Vulnerability Scoring System (CVSS).

Разработанные системы накопления знаний об уязвимостях широко используются специалистами по информационной безопасности для оценки защищенности информационных систем, а также нарушителями, осуществляющими попытки их компрометации.

1.1.1 Стандарт CVSS

Общая система оценки уязвимостей CVSS впервые была опубликована 23 февраля 2005 года исследовательской группой NIAC. Данная система должна была обеспечить универсальность и открытость оценки критичности уязвимостей программного обеспечения, однако, первая версия не была подвержена анализу со стороны организаций, в следствие чего имела ряд недостатков. Основным недостатком CVSS было слишком малое число критериев, по которым проводилась оценка из-за чего существовало

множество уязвимостей с одинаковой оценкой. Чтобы избежать подобных недостатков в будущем, была собрана группа CVSS-SIG, задача которой заключалась в выявлении и исправлении всех недостатков CVSS.

Так 1 июня 2007 года было опубликовано описание системы CVSS 2.0, которая в дальнейшем получила широкое распространение. До 2011 года она вошла в следующие стандарты:

- PCI DSS – стандарт безопасности данных платежных карт;
- NIST – национальный институт стандартов и технологий США;
- ITU-T X.1521 – международный стандарт оценки уязвимостей.

CVSS 2.0 оценивает уязвимости по степени серьёзности угрозы и состоит из трех основных групп метрик [29]: базовой, временной и контекстной, каждая из которых состоит из набора метрик, как показано на рис.1.2. При этом каждая метрика представляет собой оценку от 0 до 10 и описание, содержащее информацию для вывода данной оценки.

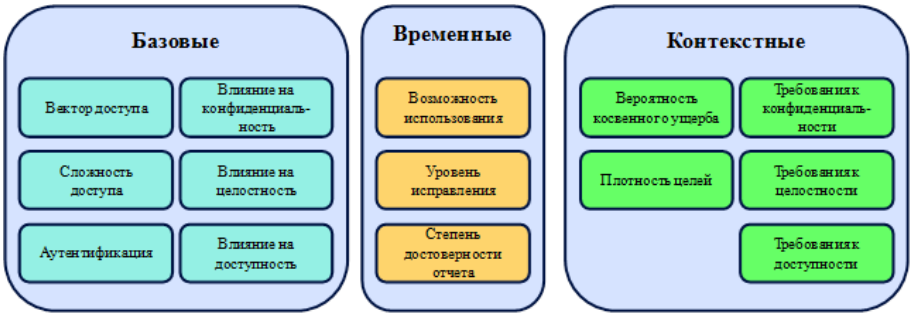

Рис.1.2. Группы метрик CVSS

Группы метрик в стандарте определены следующим образом:

- Базовые метрики представляют внутренние характеристики уязвимости, которые не меняются со временем и средой пользователя.
- Временные метрики представляют характеристики уязвимости, которые меняются со временем, но не от пользовательской среды.

- Контекстные метрики представляют характеристики, которые являются релевантными и уникальными для определенной среды пользователя.

Таким образом, базовые метрики передают основные характеристики уязвимости, что позволяет пользователю получить четкое представление об уязвимости, чтобы затем посчитать временные и экологические метрики, которые более точно отражают критичность уязвимости для их уникальной среды. Данный подход позволяет принимать более обоснованные решения при попытке снизить риски, связанные с уязвимостями.

Для оценки уязвимостей с точки зрения внешнего нарушителя достаточно рассматривать исключительно базовую группу метрик, создающую общее представление об уязвимостях, так как при оценке защищенности системы зачастую отсутствует возможность получения информации о стоимости активов, потенциальном ущербе в результате эксплуатации уязвимости и т.п.

Базовая группа состоит из 6 метрик:

- Способ получения доступа: локальный, удаленный через локальную сеть, удаленный через глобальную сеть. При этом чем дальше нарушитель, тем выше базовая оценка;
- Сложность атаки: низкая, средняя, высокая;
- Показатель аутентификации: не требуется, единственная, множественная. Показатель аутентификации отражает сложность проведения атаки с точки зрения предоставляемых злоумышленником валидных данных;
- Влияние на конфиденциальность: не оказывает, частичное, полное;
- Влияние на целостность: не оказывает, частичное, полное;
- Влияние на доступность: не оказывает, частичное, полное;

Частичным считается влияние на актив, которое может быть ограничено. При наличии физического доступа к системе или прав root, система считается полностью компрометированной.

Данные показатели подставляются в базовое уравнение:

$$BaseScore = round\_to\_1\_decimal(((0.6 * Impact) + (0.4 * Exploitability) - 1.5) * f(Impact))$$

где $Imapct$, $Exploitability$ и $f(impact)$ вычисляются следующим образом:

$$Impact = 10.41 * (1 - (1 - ConfImpact) * (1 - IntegImpact) * * (1 - AvailImpact))$$

$$Exploitability = 20 * AccessVector * AccessComplexity * Authentication$$

$$f(Impact) = \begin{cases} 0, \text{если } Impact = 0 \\ 1{,}176, \text{если } Impact \neq 0 \end{cases}$$

$round\_to\_1\_decimal$ – функция округления до одного десятичного знака после запятой в большую сторону.

В табл.1.1 проводится сопоставление качественных и количественных характеристик базовых метрик.

Таблица 1.1

Шкала оценок базовых метрик

Метрика	**Качественная характеристика**	**Количественная характеристика**
Способ получения доступа	Локальный	0.395
	Удаленный через локальную сеть	0.646
	Удаленный через глобальную сеть	1.0
Сложность атаки	Высокая	0.35
	Средняя	0.61
	Низкая	0.71

Продолжение табл.1.1

Метрика	**Качественная характеристика**	**Количественная характеристика**

Показатель аутентификации	Множественная	0.45
	Единственная	0.56
	Не требуется	0.704
Влияние на конфиденциальность, целостность и доступность	Нет	0
	Частичная	0.275
	Полная	0.660

Однако и в CVSS 2.0 был обнаружен ряд проблем

- Неоднозначная трактовка различных показателей CVSS 2.0 приводила к тому, что в различных базах уязвимостей одни и те же уязвимости имели разную оценку критичности, в виду выбора разных значений метрики сложности эксплуатации;
- Известные компании, такие как Oracle, IBM, Cisco и другие помимо CVSS 2.0 начали использовать оценки собственной разработки, которые являлись более информативными;
- Отсутствие явно определенной области действия ущерба.

В 2015 году была предложена третья версия стандарта, в которой были учтены проблемы второй версии. В неё были внесены следующие изменения:

1. Введен дополнительный уровень доступа: физический;
2. Удален средний уровень сложности эксплуатации уязвимости. В данном стандарте любое дополнительное условие, неподконтрольное нарушителю, приводит к повышению уровня сложности до высокого;
3. Добавлена метрика взаимодействия с пользователем: требуется, не требуется;
4. Метрика аутентификации заменена показатель уровня привилегий, который может принимать значения: отсутствует, низкий, высокий. Первый уровень эквивалентно отсутствию аутентификации, второе – аутентификации от имени обычного пользователя, а третий – аутентификации от имени администратора;

5. Изменены коэффициенты в уравнениях для уменьшения зависимости оценки от сложности эксплуатации и повышения её зависимости от причиняемого ущерба;

6. Определено понятие цепочки уязвимостей. Эксплуатация одной отдельной уязвимости может нести низкую угрозу, однако в цепочке с другими уязвимостями опасность уязвимостей значительно повышается;

7. Метрики воздействия изменены с частичной и полной на низкую и высокую;

8. Добавлена метрика $Scope$, определяющая область действия ущерба, наносимого уязвимостью. Иногда уязвимости, обнаруженные в одной системе, могут влиять на другие, не связанные с ней, в таком случае предлагается повышать оценку опасности уязвимости. Данный показатель может принимать значения: без изменения, с изменением.

Нововведенная версия CVSS 3.0 позволяет более точно определить опасность уязвимости для системы, однако, и в ней могут быть найдены неоднозначные трактовки, например, при определении влияния уязвимости на целостность, конфиденциальность и доступность информации.

Общая схема оценки уязвимостей по стандарту CVSS представлена на рис.1.3.

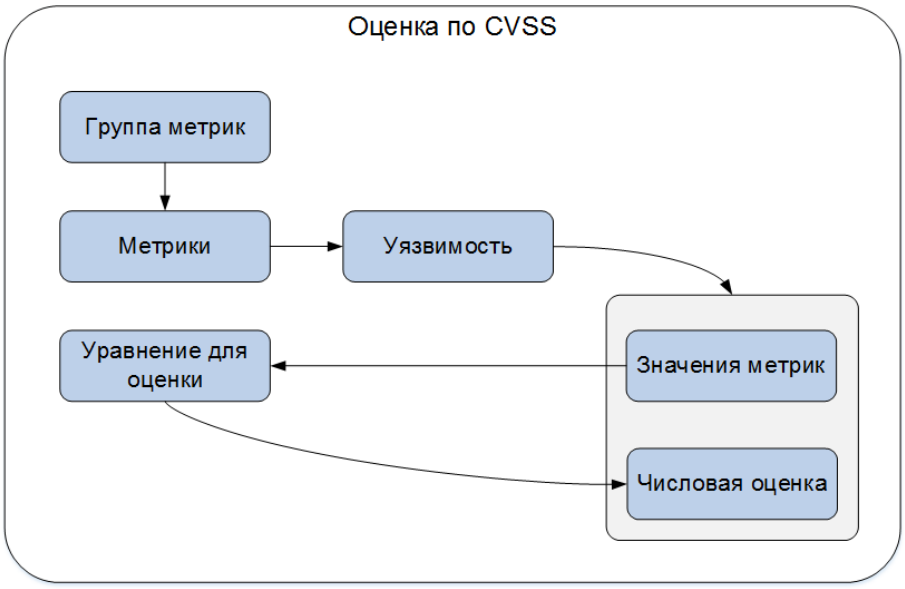

Рис.1.3. Оценка уязвимости по стандарту CVSS

1.1.2Базы уязвимостей

Для обнаружения уязвимостей и определения их опасности для исследуемой системы используются базы уязвимостей. Они хранят такую информацию как: идентификатор уязвимости, её краткое описание, подверженные уязвимости версии программного продукта.

На текущий момент существует несколько наиболее распространенных баз уязвимостей[30]:

1. Банк данных угроз безопасности информации ФСТЭК России [31];
2. MITRE CVE [32] и NVD [33];
3. OSVDB [34];
4. Secunia Advisories and Vulnerability Database [35];

Добавлено примечани
каждой из баз? Например по
CVE, как на основу в metasp

1.1.2.1 Банк данных угроз безопасности информации ФСТЭК России

На территории Российской Федерации одной из основных организаций, отвечающих за обеспечение информационной безопасности в ключевых системах информационной инфраструктуры, включая компьютерные сети органов государственной власти и компьютерные сети критичных объектов инфраструктуры и предприятий, является Федеральная служба по техническому и экспортному контролю – ФСТЭК России.

Для обеспечения деятельности по сертификации средств защиты информации и обнаружения уязвимостей программного обеспечения, ФСТЭК России с 2014 года поддерживает собственный реестр известных угроз информационной безопасности и уязвимостей программного обеспечения – Банк данных угроз безопасности информации (БДУ ФСТЭК России).

Данный реестр уязвимостей в первую очередь ориентирован на сбор и хранение информации об угрозах и уязвимостях ПО, используемого в государственных организациях Российской Федерации, включая информационные системы и системы управления критичными производственными процессами. Пополняется реестр ФСТЭК России путем мониторинга общедоступных источников информации – информационных бюллетеней российских и иностранных компаний, производящих ПО, а также реестров и информационных бюллетеней исследовательских организаций и компаний, предоставляющих услуги в области информационной безопасности.

На конец декабря 2019 года данное хранилище содержало порядка 24 тысяч записей об уязвимостях. Все хранящиеся в БДУ ФСТЭК России записи имеют единообразный формат и включают: текстовое описание уязвимости, дату обнаружения уязвимости, названия, версии и производителей уязвимого ПО, информацию о типе ошибки, классе уязвимости и текущем ее статусе (потенциально возможная либо подтвержденная производителями ПО или независимыми исследователями уязвимость, устранена ли уязвимость в

новых версиях ПО). Также записи содержат оценку критичности уязвимости и сопутствующий вектор CVSS, пометку о наличии известных готовых сценариев эксплуатации уязвимости и возможного результата эксплуатации уязвимости, указание уязвимых аппаратных платформ или операционных систем, список возможных методов противодействия уязвимости и ссылки на источники дополнительной информации по уязвимости (включая идентификаторы данной уязвимости в иных реестрах и базах данных).

Следует отметить, что записи в базе данных БДУ ФСТЭК России предоставляют более подробную информацию о различных аспектах, связанных с уязвимостью, чем иностранные реестры уязвимостей CVE List и NVD, предоставляемые американскими некоммерческими и государственными организациями.

К возможным недостаткам БДУ ФСТЭК России можно отнести меньшее общее количество покрытых реестром уязвимостей (в сравнении как с базами CVE List и NVD, так и с базами данных уязвимостей, созданных коммерческими компаниями), а также отсутствие какой-либо агрегации отдельных записей (которая характерна для такой базы данных, как Vulnerability Notes Database).

1.1.2.2 MITRE CVE и NVD

Стандарт Common Vulnerabilities and Exposures (CVE), разработанный американской некоммерческой исследовательской корпорацией MITRE Corporation в 1999 году, является на сегодняшний день основным стандартом в области унифицированного именования и регистрации обнаруженных уязвимостей программного обеспечения. Данный стандарт непосредственно определяет, как формат идентификаторов и содержимого записей об отдельных обнаруженных уязвимостях, так и процесс резервирования идентификаторов для новых обнаруженных уязвимостей и пополнения соответствующих баз данных.

В настоящее время (по данным на март 2018 года) поддержкой и администрированием реестра уязвимостей CVE занимается группа из 84 организаций по всему миру, в число которых входят ведущие производители программного обеспечения, телекоммуникационного оборудования и интернет-сервисов, такие как Apple, Cisco, Facebook, Google, IBM, Intel, Microsoft, Oracle и ряд компаний, специализирующихся в области информационной безопасности, например, F5 Networks, McAfee, Symantec, «Лаборатория Касперского» и прочие.

В рамках поддержки проекта MITRE CVE основными задачами этих организаций, называемых CVE Numbering Authorities (CNAs), являются:

- поиск и сбор информации об уязвимостях программного обеспечения (в случае разработчиков и распространителей ПО, их область ответственности ограничена непосредственно их собственными продуктами и сервисами);
- классификация найденных уязвимостей;
- резервирование CVE-идентификаторов для найденных уязвимостей;
- актуализация соответствующей информации в двух официальных каталогах – реестре уязвимостей CVE List самой MITRE Corporation и базе данных уязвимостей NVD (National Vulnerability Database, https://nvd.nist.gov/), поддерживаемой национальным Институтом Технологий и Стандартов США.

На конец 2019 года базы данных Mitre CVE и NVD содержат порядка 125 тысяч записей об отдельных уязвимостях, обнаруженных за период 1999 – 2019 г.

При этом, хотя сами базы данных различаются на уровне функциональных возможностей, предоставляемых пользователям, сами списки записей об уязвимостях фактически идентичны друг другу. Формально CVE List выступает изначальным источником записей для базы данных NVD, а специалисты, отвечающие за поддержку базы NVD,

производят уточненный анализ и сбор доступной информации по уязвимостям, зарегистрированным в CVE List (например, собирают ссылки на сторонние источники информации об уязвимости и мерах по ее устранению или предотвращению эксплуатации).

Формат идентификаторов CVE представлен на рис.1.4. Он состоит из года регистрации уязвимости – первых четырех цифр и последующих 4-6 цифр, описывающих уникальный номер уязвимости в рамках года уязвимости.

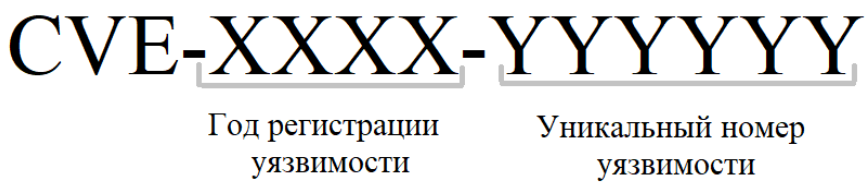

Рис.1.4. Формат идентификатора уязвимостей CVE

Для каждой из обнаруженных уязвимостей запись в базе содержит краткое описание типа и причин уязвимости, уязвимые версии ПО, оценку критичности уязвимости в соответствии со стандартом CVSS и ссылки на внешние источники с информацией об уязвимости – чаще всего, таковыми выступают информационные бюллетени на сайтах производителей программного обеспечения или исследовательских организаций.

В CVE List поддерживаются возможности простейшего поиска среди записей (по ключевым словам и CVE-идентификаторам) и скачивания архивов записей за любой выбранный год в различных форматах (HTML, XML, CVRF, CSV или Plain Text). Также возможно автоматическое получение обновлений в машиночитаемом виде через CVE Data Feed – список изменений, позволяющий отслеживать появление новых

идентификаторов CVE, а также изменения в записях уже существующих уязвимостей.

Для базы NVD в свою очередь доступны продвинутые функции поиска уязвимостей по ключевым словам, временным диапазонам создания\модификации записи, компонентам CVSS-метрики и т.п. Кроме того, доступны скачивание всех записей базы данных в XML, а также получение информации об обновлениях базы в виде RSS-подписки и JSON data feed.

Преимуществом баз данных MITRE CVE List и NVD являются ежедневное обновление реестров известных уязвимостей и продолжительность сбора данных (20 лет). При обнаружении новой уязвимости производителем ПО или исследовательской организацией под нее оперативно регистрируется новый идентификатор CVE и создается запись в базе, после чего происходит периодическое обновление информации.

Следует отметить, что в настоящее время среди участников CVE Numbering Authorities лишь две организации имеют статус корневых CNAs, находящихся под непосредственным администрированием Primary CNA (самой MITRE Corporation). На рис.1.5 представлена иерархическая структура CNAs.

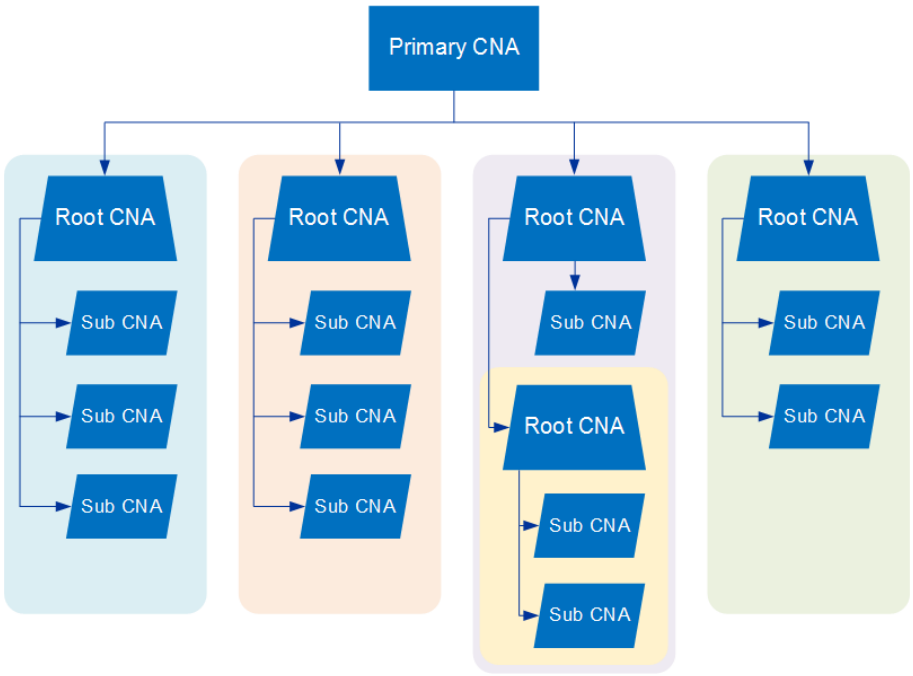

Рис.1.5. Иерархическая структура CNAs

Некоторым естественным ограничением баз данных CVE List и NVD является отсутствие в записях об уязвимостях какой-либо информации о точном месте локализации уязвимости в коде уязвимого ПО и возможных векторах атак, посредством которых возможна эксплуатация данной уязвимости. В некоторых случаях данная информация может быть найдена по ссылкам на внешние ресурсы, однако в большинстве случаев производители и поставщики ПО избегают публикации данной информации, причем не только на период разработки и внедрения патчей, закрывающих обнаруженную уязвимость, но и в последующем.

1.1.2.3 OSVDB

База уязвимостей Open Source Vulnerability Database (OSVDB) была создана в 2004 году по окончанию конференций DEF CON и Blackhat. Для

Добавлено примечание

поддержания данной базы знаний в 2005 году была сформирована организация Open Security Foundation, специалисты которой проводили агрегацию данных об уязвимостях из открытых источников, а также собственные исследования программных продуктов для обнаружения новых уязвимостей и сценариев их эксплуатации. Обнаруженные уязвимости проходили классификацию и валидацию, после чего заносились в OSVDB с ссылками на конкретные исследования, списки уязвимого ПО и способы устранения уязвимости.

Однако в 2016 году произошло закрытие базы данных уязвимостей с последующим его перерождением в коммерческий продукт VulnDB. На конец декабря 2019 года данная база содержит порядка 140 тысяч записей об уязвимостях, часть из которых отсутствует в базах CVE List и NVD. Данная база пополняется в среднем 50 записями об уязвимостях каждый день.

1.1.2.4 Secunia Advisories and Vulnerability Database

Secunia Advisories and Vulnerability Database - это база данных с информационными бюллетенями (Secunia Advisories), содержащими сведения об обнаруженных угрозах и уязвимостях ПО. Бюллетени формируются на основе собственных исследований специалистов Secunia Research и агрегации информации об уязвимостях, полученных из иных публичных источников.

Бюллетени в базе данных Secunia зачастую публикуются еще до того, как соответствующие записи появляются в базе CVE List, и уже впоследствии размечаются ссылками на соответствующие CVE-идентификаторы. При этом нередки ситуации, когда никто из CVE Numbering Authorities так и не берется за регистрацию обнаруженной уязвимости, в результате чего соответствующая запись из базы Secunia Research так и остается без выделенного CVE-идентификатора и, соответственно, не попадает в базы CVE List и NVD.

По своей структуре записи в базе Secunia сходны с содержимым баз CVE List и NVD и содержат дату регистрации уязвимости, тип и краткую классификацию уязвимости, критичность уязвимости (описывается с помощью перечислимого типа Secunia Research Criticality Rating вместо скалярной оценки по стандарту CVSS), списки уязвимого ПО и его версий, ссылки на внешние источники информации и рекомендации по устранению угрозы (как правило, установку патчей от производителя ПО или апгрейд уязвимого ПО до безопасной версии – в этом случае бюллетень содержит упоминание минимального номера безопасной версии).

Характерной особенностью Secunia Advisories является агрегирование в одной записи информации о множестве отдельных уязвимостей, одновременно обнаруженных в одном и том же программном обеспечении. Это означает, что одной записи из базы данных Secunia может соответствовать множество различных CVE-идентификаторов. База данных Scunia Research пополняется с 2003 года.

При бесплатном доступе к данным Secunia информация об уязвимостях предоставляется только в формате html. Для коммерческого использования доступ к базе данных от Secunia Research предоставляется посредством специализированного средства Software Vulnerability Manager и соответствующей подписки на сервис компании Flexera, которой и принадлежит с 2015 года Secunia Research.

1.1.3 Основные уязвимости, приводящие к компрометации узла

Компрометация узла возможна в том случае, если злоумышленнику удаётся удаленно выполнить произвольный код на целевом узле. Уязвимости, способствующие проникновению на узел можно поделить на следующие типы [36]:

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;

Добавлено примечание
существующие эксплоиты

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

1.2. Эксплойты

Эксплойт [37] – последовательность действий, 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 [38]:

- 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.

Большинство существующих эксплойтов входят в ту или иную базу эксплойтов. Часть из них находится в открытом доступе, другая – продается в «DarkNet».

Наиболее распространенные базы эксплойтов [39]:

- 0day.today – база данных эксплойтов, работающая на коммерческой основе. Содержит множество эксплойтов, недоступных в открытых источниках. Доступна только через TOR;
- Rapid7 – база данных, содержащая рекомендации по исправлению уязвимостей, а также ссылки на модули Metasploit Framework, которые автоматизируют эксплуатацию эксплойта;
- Exploit-db – архив общедоступных эксплойтов и уязвимого программного обеспечения, разработанный для использования в процессе тестирования на проникновение.

1.3. Тестирование на проникновение

Тестирование на проникновение – это услуга в сфере информационной безопасности, суть которой заключается в санкционированной попытке

проникнуть в информационную систему и обойти существующий комплекс средств ее защиты.

Процесс тестирования на проникновение предусматривает моделирование реальных действий злоумышленника, поиск уязвимостей системы защиты и их дальнейшую эксплуатацию. Тест на проникновение позволяет получить независимую оценку и экспертное заключение о состоянии защищенности сетевой инфраструктуры организации.

Существует несколько подходов к проведению тестирования на проникновение (см. рис.1.6):

1. Метод белого ящика – подход, при котором тестировщик имеет полный доступ к глубоким знаниям о функционировании и основных атрибутах системы. Это тестирование очень эффективно, так как понимание каждого аспекта системы очень полезно при проведении обширных испытаний на проникновение;
2. Метод серого ящика – подход, при котором тестировщик получает ограниченную информацию о системе (например, знания алгоритма, архитектуры, внутренних состояний) для имитации внешней атаки на систему;
3. Метод черного ящика – подход, при котором тестировщику предоставляется только высокоуровневая информация (например, URL или IP-адрес организации) для проведения тестирования на проникновение. Это весьма трудоемкий подход, так как тестировщику требуется значительное количество времени для изучения свойств и деталей системы; кроме того, высока вероятность пропустить часть областей из-за недостатка времени и информации. Также значительно усложняется оценка рисков, ввиду отсутствия знаний о бизнес-логике организации.

Рис.1.6. Подходы к проведению тестирования на проникновение

Тестирование методом черного ящика позволяет максимально приблизить процесс тестирования на проникновение к хакерской атаке, от которой организация стремится защитить свои активы.

Разрабатываемые в данной работе методики оценки защищенности и выбора защитных мер способны обеспечить корректную выработку рекомендаций независимо от применяемого подхода.

Процесс тестирования на проникновение в общем случае состоит из 4 этапов [40]:

1. Планирование;
2. Сбор информации о системе;
3. Проникновение на целевой узел (атака);
4. Составление отчёта.

Этап планирования включает в себя сбор требований, определение сферы применения, стратегий и целей тестирования проникновения в соответствии с нормами безопасности. Кроме того, он может содержать оценку и перечисление проверяемых областей, типы планируемых испытаний и другие связанные с этим проверки.

На втором этапе осуществляется сбор и анализ максимально подробной информации о системных и связанных с ними атрибутах безопасности,

которая напрямую влияет на эффективность тестирования системы. Также на данном этапе тестировщики выявляют и обнаруживают уязвимые области системы, которые в дальнейшем будут использоваться для увеличения контроля над системой.

Этап проникновения включает в себя фактическое испытание обнаруженных уязвимостей. Проникновению могут подвергнуться как внешние узлы сетевой инфраструктуры, так и внутренние, расположенные в локальной сети организации.

Этап составления отчётности включает в себя документационную работу по мероприятиям, проводимым на всех упомянутых этапах. Кроме того, она может описывать различные риски, выявленные проблемы, уязвимые области (использованные или нет) и предлагаемые для устранения недостатков решения.

В процессе тестирования возможно возвращение ко второму этапу при обнаружении новых целей. Более того, на каждом этапе может быть задействовано неограниченное число дополнительных операций.

Тестирование на проникновение системы может осуществляться с использованием любого из следующих подходов:

- ручное тестирование;
- автоматическое тестирование;
- сочетание ручного и автоматического тестирования.

Общая схема проведения тестирования выглядит следующим образом (см. рис.1.7).

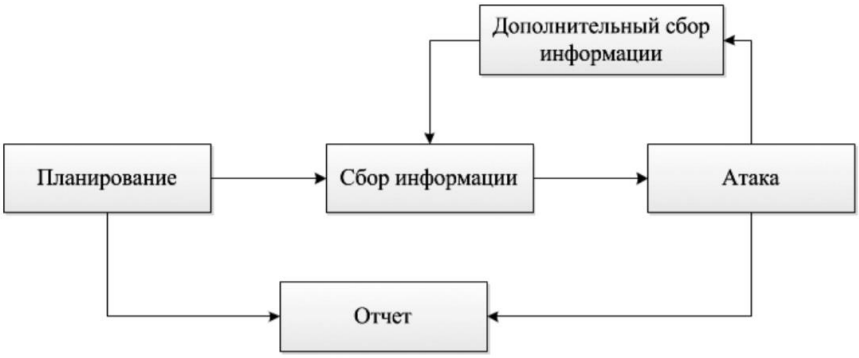

Рис.1.7. Процесс тестирования на проникновение

Оценка защищенности сетевой инфраструктуры проводится на основании информации, полученной на этапе сбора информации о системе и подтвержденной на этапе проникновения. Данные этапы рассматриваются более подробно в п.1.3.1 и п.1.3.2.

1.3.1Сбор информации

Разнообразие приложений, протоколов, операционных систем и прошивок оборудования ставит перед тестировщиком задачу по точной идентификации как самого сетевого устройства, так и установленного на нем программного обеспечения и других важных для этапа проникновения параметров.

На этапе сбора информации о системе осуществляется:

- Определение всех доступных узлов системы;
- Определение имен хостов;
- Определение контактной информации сотрудников организации;
- Определение типа узлов;
- Определение типа и версии операционных систем;
- Получение баннеров с обнаруженных портов;
- Определение типа и версий определенных сервисов;

- Определение разделяемых ресурсов;
- Определение пользователей и групп сервисов;
- Идентификация средств защиты;
- Получение списка актуальных уязвимостей;
- Получение списка доступных эксплойтов.

Для осуществления вышеописанных действий могут быть использованы различные автоматизированные сканеры, например, Devsploit, RedHawk, Nmap, Nessus или OpenVAS и снифферы, например, WireShark. Для определения уязвимых версий программного обеспечения, а также доступных эксплойтов могут использоваться базы данных, описанные в п.1.1.2 и п.1.2 данной работы. Определение имен хостов и контактной информации сотрудников может быть осуществлено с помощью сервисов WHOIS или запросов к DNS-серверу.

Помимо автоматического поиска уязвимостей обязательно проводится ручной поиск уязвимостей на основе опыта, накопленного тестировщиком.

На основании полученных данных проводится выявление потенциальных векторов атак и их апробирование.

1.3.2Проникновение на целевой узел

На данном этапе осуществляется проверка ранее обнаруженных уязвимостей и недостатков системы путем их эксплуатации. Таким образом, подтверждается или опровергается факт наличия той или иной уязвимости и определяется её влияние на информационную безопасность сетевой инфраструктуры организации.

На этапе проникновения используются как автоматизированные утилиты (Metasploit, Autopwn, Armitage и т.п.), содержащие набор готовых модулей для эксплуатации уязвимостей, так и написанные вручную эксплойты.

В результате проникновения на очередной узел тестировщик может получить: информацию об учетных записях пользователей системы и

сервисов (Системы Управления Базами Данных (СУБД), почты, Системы контроля версий, и т.п.); исходные коды различных проектов; доступ к ранее недоступным узлам или новой сети и т.д.

Полученный в результате проникновения доступ к узлам системы и новым сетям используется на этапе сбора информации для расширения покрытия сети тестировщиком.

В данной главе была рассмотрена концепция тестирования на проникновение: понятие уязвимости и эксплойта, крупнейшие базы уязвимостей и эксплойтов, общая система оценки уязвимостей CVSS, основные подходы к тестированию и этапы его проведения.

В процессе тестирования на проникновение рейтинг CVSS может сыграть ключевую роль при определении векторов атак, однако, он не может быть в полной мере использован для оценки защищенности отдельного узла или сетевой инфраструктуры, так как расчет вероятности эксплуатации конкретной уязвимости, эксплойты к которой отсутствуют в открытом доступе, является трудно решаемой задачей. Анализ уязвимости и написание собственного эксплойта может занять намного больше времени, чем выделено на тестирование на проникновение.

Следовательно, в методиках оценки защищенности и выбора защитных мер следует учитывать только те узлы сети и уязвимости, эксплуатация которых привела к компрометации данных узлов.

Информация, полученная на этапе сбора сведений об узлах сети, будет отфильтрована и использована в качестве метрики риска компрометации для каждого узла в главе 3 данной работы. Такой способ не требует наличия дополнительной информации о системе, предоставить которую может только заказчик, и, как следствие, отлично подходит для использования в автоматизированной системе оценки защищенности.

2. ОЦЕНКА ЗАЩИЩЕННОСТИ И ВЫБОР МЕР ЗАЩИТЫ СЕТЕВОЙ ИНФРАСТРУКТУРЫ ОРГАНИЗАЦИИ

Оценка защищенности сетевой инфраструктуры – это устоявшийся метод выявления слабых мест организации. Обеспечить полноценную оценку безопасности информационных систем, содержащих сотни или тысячи узлов, и выбор мер защиты задача нетривиальная. В данной главе рассматриваются основные методики оценки защищенности компьютерных сетей, характеристики систем и узлов, определяющие показатели защищенности, на основании которых проводится поиск оптимальных защитных мер.

2.1. Методики оценки защищенности сетевой инфраструктуры

Оценка защищенности сетевой инфраструктуры – это процесс выявления уязвимостей, угроз и рисков, связанных с активами организации и мер защиты, которые могут смягчить эти риски [2]. Существует два базовых подхода к оценке защищенности: качественный и количественный, а также смешанный подход, представляющий собой совокупность обоих подходов. Методики оценки защищенности позволяют выразить защищенность сетевой инфраструктуры организации в виде показателей защищенности.

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 (см. рис.2.1) [3]

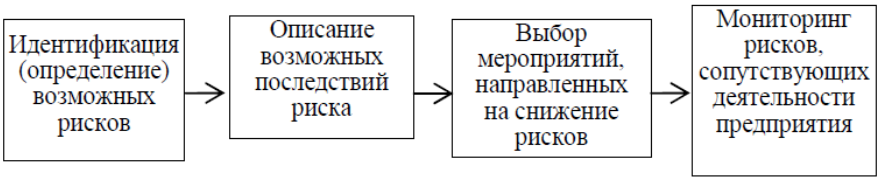

Рис.2.1. Последовательность проведения качественного анализа рисков

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 [4];
- OCTAVE [5];
- FRAP [6].

2.1.1.1 Методика COBRA

Методика COBRA разработана компанией Risk Associates [7]. 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 [8].

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 методики оценки защищенности

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 (см. рис.2.2) [3].

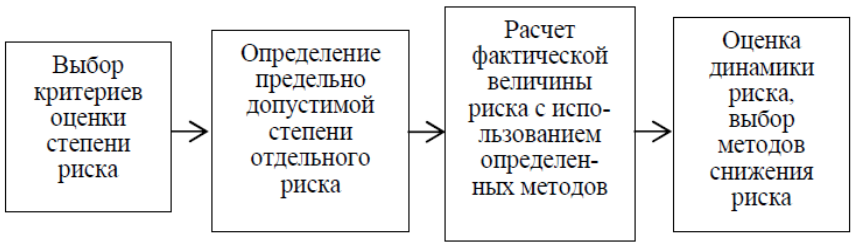

Рис.2.2. Последовательность проведения количественного анализа рисков

К количественным методикам относятся:

- RiskWatch;
- ГРИФ.

2.1.2.1 Методика RiskWatch

Методика RiskWatch [9], разработанная при участии NIST, использует значение ожидаемых годовых потерь для оперирования значениями риска (ALE) и значение возврата инвестиций от реализации методов защиты (ROI).

На основе данных параметров методика позволяет определить точное соотношение потерь от реализации угроз безопасности и стоимости реализации средств защиты.

В основе методики лежит обширная база знаний, содержащая информацию по активам, угрозам, уязвимостям, видам ущерба, защитным мерам а также опросным листам для оценки факторов риска.

Процесс анализа рисков по методике RiskWatch осуществляется в несколько этапов с использованием одноименного программного обеспечения:

1. Определение параметров исследования. Данный этап включает в себя выбор основных угроз безопасности, уязвимостей системы, применяемых мер защиты, видов активов и категории ущерба;
2. Ввод данных. На данном этапе в систему оценки вносится информация о важности активов, вероятности реализации угроз, частоте угроз, степени уязвимости системы и стоимости мер защиты. Важность актива определяется в денежном эквиваленте и соответствует величине ущерба, причиненного организации в результате нарушения конфиденциальности, целостности и доступности конкретного актива;

В базе знаний RiskWatch каждой угрозе сопоставляется стандартная оценочная частота реализации угрозы (SAFE). В процессе вычисления рисков используется локальная оценочная частота угрозы (LAFE), которую вводит пользователь на данном этапе, используя в качестве базового значения SAFE.

Для каждой защитной меры определяется стоимость, которая зависит от стоимости внедрения и сопровождения данной меры. Дополнительно учитываются процент реализации контрмеры, срок её эксплуатации и насколько она уменьшит значение LAFE.

Также оценщик проходит опрос приблизительно из 600 вопросов для более точной оценки рисков.

3. Оценка рисков. На данном этапе осуществляется расчет показателя ALE для каждой комбинации актив-угроза-уязвимость на основе информации о ценности активов, частоте угроз и степени уязвимости. Далее производится расчет ROI для мер защиты и поиск такой комбинации, при которой произойдёт максимальное уменьшение рисков при оптимальной комбинации контрмер. Формула расчёта ROI:

$$ROI = \sum_i NVP(Benefits_i) - \sum_j NVP(Costs_j)$$

где $Costs_j$ – затраты на реализацию мер защиты; $Benefits_i$ – ожидаемое снижение риска, полученное в результате реализации защитных мер; NVP – чистая стоимость.

4. Формирование отчётов. На данном этапе генерируются отчёты по проведенной оценке рисков. Данные отчёты содержат информацию о:

- стоимости защищаемых ресурсов и ущербе от их компрометации;
- ROI для каждой контрмеры;
- угрозах и мерах противодействия;
- результате аудита безопасности.

Рекомендуемые контрмеры могут быть отсортированы в порядке убывания значения ROI.

Таким образом, данная методика позволяет оценить не только текущие риски организации, но и выгоду, которую получит организация от внедрения тех или иных мер защиты.

2.1.2.2 Методика ГРИФ

Система ГРИФ была разработана компанией Digital Security[10] в 2005 году, как система анализа и управления информационными рисками. Система ГРИФ может проводить несколько видов анализа: на основе модели информационных потоков, требующей от пользователя информации обо всех критически важных ресурсах, о пользователях, которые имеют к ним доступ и их правах доступа, и на основе модели угроз и уязвимостей. Для целей

данной работы наиболее интересной представляется модель анализа угроз и уязвимостей, которая позволяет определить уязвимость каждого ресурса, содержащего информацию, подлежащую защите, и угрозы, которые могут быть реализованы, используя данные уязвимости. В результате работы данной системы создаётся полная картина того, какие слабые места есть в системе и какой ущерб может быть ей нанесен.

Работа системы на основе модели уязвимостей и угроз выполняется в несколько этапов:

1. Пользователь определяет отделы, активы, угрозы и уязвимости информационной системы. Система ГРИФ содержит большую базу знаний об угрозах и уязвимостях, полнота которой достигается благодаря специально разработанной классификации угроз DSECCT, воплотившей в себе многолетний опыт в области ИБ (около 200 уязвимостей и 100 угроз);
2. Для вышеописанных ресурсов системы пользователь определяет связи (актив-отдел, уязвимость-угроза, актив-угроза);
3. Система ГРИФ анализирует полученные данные и генерирует отчёт, содержащий значение риска для каждого актива;
4. На основе полученных отчётов производится поиск оптимальных контрмер для снижения уровня остаточного риска;
5. На выходе система генерирует отчёт об уровне риска каждого актива, результат анализа причин наличия рисков и оценку эффективности различных мер защиты.

2.1.3 Смешанные методики оценки защищенности

Смешанные методики сопоставляют качественному уровню определенный количественный диапазон [11].

К таким методикам относятся:

- CRAMM;
- Методики оценки рисков на основе графов атак [13].

2.1.3.1 Методика CRAMM

Методика CRAMM [12] 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 приоритизация.

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.

2.1.3.2 Методика оценки рисков на основе графов атак

В работе [13] была предложена методика оценки рисков на основе построения и анализа графа атак для оценки защищенности сети. Данная методика использует модифицированную формулу расчета CVSS для определения уровня риска. Вместо исходного параметра CVSS SecurityRequirements вводится показатель Criticality, определяющий ценность актива для организации, который вычисляется с учетом финансовой стоимости актива и зависимостей свойств безопасности активов. Значения варьируются от 0 до 100. При этом выделяют следующие диапазоны критичности:

- [0:0,01) – ничтожно малая;
- [0,01:0,1) – малая;
- [0,1:1) – значительная;
- [1:10) – повреждающая;
- [10:100) – серьезная;
- 100 – смертельная.

В табл.1.1 приведено преобразование шкалы для использования в уравнении CVSS.

Таблица 1.1

Преобразование шкалы оценок критичности

Критичность	**Значение**
[0:0,01)	0
[0,01:0,1)	0,5
[0,1:1)	1
[1:10)	1,2
[10:100)	1,4
100	1,51

Уравнение CVSS:

$$Risk = round\_to\_1\_decimal(AdjustedBase)$$

В раскрытом виде:

$$Risk = round\_to\_1\_decimal(((0{,}6 * AdjustedImpact) + (0{,}4 * Exploitability) - 1{,}5) * f(AdjustedImpact))$$

где $Exploitablity$ – возможность использования уязвимости,

$$f(AdjustedImpact) = \begin{cases} 0, \text{если } AdjustedImpact = 0 \\ 1{,}176, \text{если } AdjustedImpact \neq 0 \end{cases}$$

$$AdjustedImpact = \min(10{,}10{,}41 * (1 - (1 - ConfImpact * ConfReq) * * (1 - IntegImpact * IntegReq) * (1 - AvailImpact * AvailReq))),$$

где $ConfImpact,\ IntegImpact,\ AvailImpact$ – влияние на конфиденциальность, целостность и доступность в результате эксплуатации уязвимости; $ConfReq, IntegReq, AvailReq$ – требования безопасности, а в данном контексте - критичность актива.

Тогда уравнение принимает вид:

$$AdjustedImpact = \min(10{,}10{,}41 * (1 - (1 - ConfImpact * Criticality(c)) * * (1 - IntegImpact * Criticality(i)) * (1 - AvailImpact * Criticality(a))))$$

где функция Criticality – критичность актива по нарушению конфиденциальности, целостности и доступности.

Исходя из вышеописанных формул риск может принимать значение от 0 до 10. После определения риска каждой уязвимости актива проводится определение риска программного обеспечения, как максимального риска его уязвимостей. Затем проводится оценка риска актива схожим образом. Уровень риска системы определяется как максимальной оценкой риска всех активов системы, как высокий/средний/низкий в соответствии с CVSS. Данная методика позволяет выделить наиболее незащищенные участки системы.

Данная методика наиболее близка к применению в реальных условиях тестирования защищенности сетевой инфраструктуры, так как оценку уязвимостей по CVSS может провести и сторонний наблюдатель в отличие от методик, использующих опросы и требующих указать критичность активов с точки зрения бизнес-процессов организации. Однако данная методика не учитывает важность самих узлов, но лишь возможность их компрометации путем эксплуатации наиболее критичных уязвимостей. Так, например, два

узла, имеющие одинаковые максимальные по критичности уязвимости получат одинаковое значение риска, однако на одном из них может находиться несколько жизненно важных для работы инфраструктуры сервисов, важность которых никак не будет учтена. Также данная методика не описывает меры, принимаемые для снижения рисков.

Таким образом, необходимо разработать методику оценки защищенности сетевой инфраструктуры, которая соответствует следующим требованиям:

- Активы организации представляют собой сетевые узлы;
- Стороннее лицо, проводящее оценку защищенности сетевой инфраструктуры, должно быть способно провести самостоятельную оценку ценности активов для организации;
- Ценность узлов должна определяться на основании общедоступной информации и определяться целочисленным значением;
- Ценность узла должна зависеть от работающих на нем сервисов;
- Выбор защитных мер должен осуществляться с целью минимизации риска всей инфраструктуры.

2.2 Концепция графов атак

Существует множество работ, в которых графы атак применяются в задаче оценки защищенности сетевой инфраструктуры. Они применяются для анализа того, каким образом может развиваться атака внутри сети организации.

Граф атак можно представить как последовательность всех возможных действий злоумышленника для реализации угроз, так называемых трасс атак. Существует несколько общепринятых типов графов атак [14]:

- Полный граф атак – узлы такого графа представляют собой состояния, а ребра – уязвимости. Такие графы иллюстрируют каждую возможную трассу атак, которую может реализовать нарушитель. Они имеют

сложность $O(n!)$, что негативно сказывается на их размере и, следовательно, на скорости вычислений. Часть полного графа атак представлена на рис.2.3.

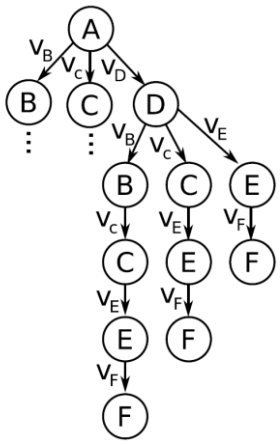

Рис.2.3. Часть полного графа

- Граф предсказаний – узлы и ребра представляют собой те же сущности, что и в полном графе. Каждый узел попадает в граф предсказаний, если ни один его предок не использует ту же уязвимость для попадания в то же состояние. Данные графы не имеют недостатка полного графа по скорости построения и могут правильно прогнозировать влияние удаления любой из уязвимостей в сети. Как следствие такие графы строятся намного быстрее, чем полные, но тем не менее всё ещё содержат лишние структуры. Пример полного графа приведен на рис.2.4.

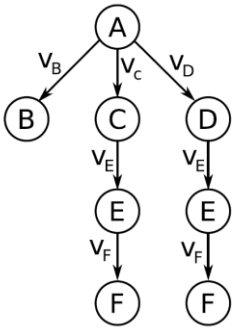

Рис.2.4. Граф предсказаний

- MP-граф (граф со множеством предусловий) – содержит три типа узлов: уязвимости, состояния и предусловия. Для отображения связей с уже существующими узлами добавляются дополнительные циклические дуги. Данный граф строится быстро и может быть преобразован в полный граф или граф предсказаний. Пример такого графа приведен на рис.2.5.

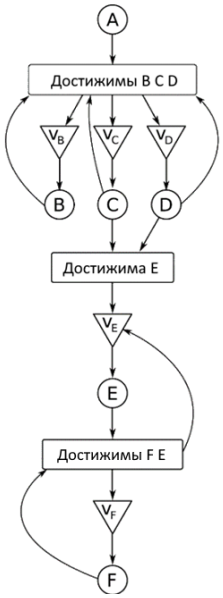

Рис.2.5. MP-граф атак

При работе с графами атак возникает две основные проблемы:

- Обработка циклов;
- Значительное время обработки.

Для решения данных проблем применяются различные предположения, такие как: свойство монотонности графа и отказ от повторного посещения узлов; выделяются различные типы циклов графа и производятся расчеты вероятностей применения атакующих действий в зависимости от вида цикла.

2.3 Показатели защищенности

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

Показатели защищенности делятся на 2 типа:

- Базовые показатели защищенности;
- Интегральные показатели защищенности.

2.3.1Базовые показатели

Базовыми называются показатели защищенности, непосредственно характеризующие элементы конфигурации и безопасности анализируемой системы, такие как: запущенные на узлах сервисы; уязвимости; источники угроз; атакующие действия; защитные меры. Определение их значений не требует проведения дополнительных вычислений.

В частности, в различных работах встречаются следующие виды показателей защищенности:

- В работе [15] показатель защищенности определяется исходя из ущерба сетевой инфраструктуры, связанного с реализацией угроз;
- В работе [16] рассматривается показатель на основе уровня навыков атакующего. Уровень навыков варьируется в зависимости от уровня,

атакующего и его знаний о системе, а уровень атакующего определяется на основе максимальной сложности атакующих действий;

- В [17] используется целый ряд показателей защищенности:
  - Суммарное количество обнаруженных уязвимостей;
  - Актив с наибольшим текущим уровнем риска организации;
  - Процент персонала, прошедшего обучение соблюдению политик и процедур информационной безопасности.
- В работе [18] выделяют такие показатели, как стоимость инцидентов, процент активов без уязвимостей, характеризующие количественные и стоимостные характеристики уязвимостей. В качестве показателей конфигурации системы используются метрики количества приложений и сервисов, процент критичных приложений и сервисов и другие.

Рассмотренные показатели защищенности позволяют охватить лишь часть критериев, требуемых для проведения полноценной проверки защищенности. Также большинство рассмотренных показателей требует вмешательства эксперта и дополнительных знаний о системе, что недоступно лицу, проводящему независимую оценку защищенности.

2.3.2Базовые показатели на основе графов атак

Добавлено примечание
схемой или таблицей?

В работе [19] рассматриваются релевантные исследования в области показателей защищенности, которые делятся на следующие группы:

- Топологические характеристики;
- Стоимостные характеристики;
- Характеристики атаки и реакции на атаку;
- Характеристики системы;
- Характеристики нарушителя;
- Показатели, применяемые при анализе уязвимостей нулевого дня.

Так, в работе [20] выделяются следующие топологические показатели:

- Критичность хоста – зависит от ущерба для бизнеса при потере данного хоста;
- Незащищенность – определяется простотой использования уязвимостей хоста и его достижимостью;
- Нисходящий риск – риск, полученный в результате прохождения через все хосты, атакуемые с данного хоста.

К стоимостным характеристикам относятся такие показатели, как:

- Ожидаемые годовые потери;
- Общий выигрыш – рассчитывается как разница между выигрышем от снижения годовых потерь и затратами на реализацию средств защиты;
- Возврат инвестиций от реагирования на атаку.

В качестве характеристики атаки рассматриваются:

- Потенциал атаки – определяет насколько близко нарушитель находится относительно своей цели;
- Уровень уверенности – определяет меру уверенности в том, что атака осуществляется в текущий момент;
- Показатель уверенности в компрометации – вероятностная величина, определяющая вероятность компрометации определенного узла;
- Ущерб от атаки – показатель, определяющийся на основе ценности активов;
- Эффективность реагирования;
- Выигрыш при реагировании;
- Побочные потери при реагировании.

В качестве характеристики нарушителя рассматривается уровень навыков нарушителя – помогает определить возможность проведения атаки нарушителем.

К характеристикам, используемым при анализе уязвимостей нулевого дня относятся:

- Вероятностная мера уязвимости – определяет насколько вероятно появление уязвимости нулевого дня определенной критичности за определенный срок;
- К-безопасность нулевого дня – определяет устойчивость сети к уязвимостям нулевого дня.

На основании вышеописанных уязвимостей авторы [20] проводят классификацию показателей защищенности, приведенную на рис.2.6.

Рис.2.6. Классификация показателей защищенности на основе графа атак

В данной работе предлагается использовать риск компрометации каждого узла в качестве первичного базового показателя и топологический показатель нисходящего риска, рассчитанного как сумма показателей риска компрометации каждого узла, достижимого из текущего.

2.3.3Интегральные показатели

Интегральными называются показатели, непосредственно характеризующие безопасность всей сетевой инфраструктуры. Использование интегральных показателей – один из наиболее перспективных путей решения задачи оценки защищенности сети и выбора контрмер [24].

Многие рассмотренные методики на финальной стадии выбора защитных мер используют именно интегральные показатели, потому что они позволяют определить такие характеристики системы как: уровень риска или поверхность атаки, которая определяется на основе ресурсов, которые могут использоваться при проведении атаки. Однако, использование интегральных показателей имеет свои недостатки: перерасчет таких показателей с целью выбора наиболее оптимальных защитных мер занимает значительное время. Уменьшению времени обработки интегральных показателей способствует оптимизация алгоритмов вычисления показателей, либо внедрение ограничений на используемые графы атак.

В данной работе рассматривается методика вычисления интегрального показателя уровня риска с использованием оптимизаций, способствующих уменьшению числа пересчитываемых базовых показателей, от которых зависит уровень риска системы.

2.4Методики выбора защитных мер на основе графов атак

В процессе оценки защищенности сетевой инфраструктуры, на этапе выбора защитных мер, необходимо провести приоритизацию и применение соответствующих защитных мер, способствующих понижению уровня риска системы.

Задача поиска оптимальных защитных мер на графе атак зачастую является нетривиальной ввиду наличия проблем, описанных в п.2.2 данной работы. В работе [21] описано несколько вариантов решения данной задачи.

2.4.1 Методика поддержки принятия решений, интегрированная в IDS

Одним из вариантов решения данной задачи является система поддержки принятия решений в выборе контрмер, которая реализуется как часть системы обнаружения вторжений (IDS) [22]. Система содержит хранилище ранее проведенных атак и контрмер, представленных в LAMBDA – универсальном языке, используемом для описания атак в терминах условий, предшествующих эксплуатации (предусловий) и пост-эксплуатации (постусловий).

Контрмеры представлены в LAMBDA следующим образом: предварительное условие представляет собой состояние системы, которое требует применения контрмеры, а постусловие является состоянием системы после применения контрмеры (обычно отрицание предварительного условия). Авторы определили понятия корреляции и антикорреляции следующим образом: два состояния коррелируют, если постусловие одного поддерживает предварительные условия другого (т.е. эти экземпляры представляют собой этапы атаки в сценарии многоступенчатой атаки). Два состояния не коррелируют, если постусловие одного противоречит предварительным условиям второго (т.е. одно состояние представляет собой защитную меру, которая предотвращает появление второго состояния, представляющего атаку). Процесс рекомендаций по противодействию инициируется оповещениями IDS. При получении предупреждения будущие шаги атаки определяются путем поиска корреляций в хранилище, а контрмеры - путем поиска антикорреляций.

Таким образом, для каждого узла графа-состояния строится набор антикорреляций, который не позволит совершить атаку. Метод хорош для противостояния известным атакам в режиме реального времени, но совершенно не подходит при статическом анализе графа с целью минимизации рисков.

2.4.2Методики на основе теории игр

Данные методики характеризуются последовательностью состояний и действий атакующего и защитника (стратегии). Изменение состояния происходит при осуществлении действий в рамках определенной стратегии. Защищающая сторона стремится к выбору минимальной по стоимости стратегии, характеризующейся минимизацией затрат и времени, необходимых для реализации всех действий стратегии. На выходе данной методики – оптимальная стратегия, которую администратору необходимо применить для обеспечения защищенности системы.

Преимущество таких методик – учет временного аспекта.

2.4.3Методики на основе количества достижимых узлов

В работе [14] рассматривается методика выбора защитных мер на основе показателей, определяющих как много путей атаки будет заблокировано, введением конкретной защитной меры. В качестве защитной меры в данном случае выступает удаление узла-уязвимости MP-графа. Для упрощения поиска оптимального решения используется дополнительный тип узлов графа, все исходящие ребра которого направлены на узлы-уязвимости, которые требуют одинаковых условий для эксплуатации. Таким образом, остаётся определить, устранение путей к какому из таких узлов принесет наибольшую эффективность. Однако в работе не определяются конкретные показатели защищенности и методы их вычисления.

2.4.4Методика на основе показателя процента компрометации сети (NCP)

Данная методика [23] опирается на показатель NCP, определяющий процент узлов, на которых атакующий может получить пользовательский или администраторский доступ. Показатель принимает значения в диапазоне от 0 до 100%. Выбор защитных мер осуществляется следующим образом:

- Производится обход графа в ширину, для каждого узла создаётся список входящих дуг. Дуги соответствуют уязвимостям, которые могут быть применены для компрометации узла;
- Для полученного набора дуг подбираются защитные меры;
- Узлы с одинаковым набором защитных мер объединяются в группы;
- Для каждой защитной меры рассчитывается NCP (каждая защитная мера удаляет одну дугу графа);
- Все защитные меры сортируются по увеличению значения NCP;
- Выбирается мера с наименьшим NCP, то есть такая мера, разница предыдущего и текущего NCP которой максимальна.

Данная методика схожа с методикой, представленной в п.2.4.3 данной работы, однако, работает куда медленнее ввиду того, что достижимость пересчитывается для каждой отдельной контрмеры.

Большинство рассмотренных методик [13][23][14] используют показатели, которые могут быть применены при оценке защищенности системы сторонним лицом, не имеющим доступа к описанию бизнес-процессов организации, реальной стоимости активов и их важности для организации, однако, они не решают проблему сложности построения графа и устранения циклов. Более в методиках [14][23] не предлагаются алгоритмы вычисления показателей защищенности и выбора защитных мер.

В данной главе было рассмотрено несколько различных методик оценки защищенности сети, выбора контрмер и показателей защищенности. Наиболее актуальными для данной работы являются методики, основанные на анализе графов потенциальных атак. Такие методики позволяют определить слабые места сетевой инфраструктуры организации и осуществить выбор средств защиты для минимизации рисков в процессе проведения тестирования на проникновение.

Методики, основанные на графах атак, допускают ограниченную осведомленность атакующего о топологии сети, её бизнес-процессах и ценности узлов для организации, однако ни одна из рассмотренных методик не может в полной мере обеспечить автоматизацию процесса анализа с полным набором входных данных, доступных для атакующего. При этом не во всех рассмотренных методиках решаются ключевые проблемы использования графов атак в процессе оценки защищенности: наличие циклов и значительное время построения.

Возникает необходимость в создании методики, которая будет соответствовать следующим требованиям:

1. Активы организации представляют собой сетевые узлы;
2. Проверяющий должен быть способен провести оценку защищенности сетевой инфраструктуры, опираясь исключительно на те сведения о системе, которые может получить самостоятельно.
3. Ценность узлов должна вычисляться на основании общедоступной информации, полученной на этапе сбора информации в процессе проведения тестирования на проникновение и определяться целочисленным значением;
4. Ценность узла должна напрямую зависеть от запущенных на нем сервисов;
5. Ценность узла должна зависеть от его типа;
6. В качестве базового показателя защищенности должен быть использован параметр нисходящего риска, вычисление которого основано на ценности достижимых из текущего узла хостов.
7. В качестве интегрального показателя должен выступать уровень риска системы, вычисляющийся как сумма параметров нисходящего риска каждого из узлов графа;
8. Алгоритм выбора защитных мер должен обеспечивать отсутствие циклов и оптимизацию этапа перестройки графа атак;

9. Выбор защитных мер должен осуществляться с целью минимизации уровня риска всей сетевой инфраструктуры.

3 РАЗРАБОТКА МЕТОДИК ОЦЕНКИ ЗАЩИЩЕННОСТИ И ВЫБОРА ЗАЩИТНЫХ

В данной главе описываются основные алгоритмы вычисления показателей защищенности, оценки защищенности сети с использованием данных показателей, а также выбора защитных мер.

В качестве структуры для хранения и анализа данных используется граф атак, построение которого может происходить в процессе проведения тестирования на проникновение.

В процессе оценки защищенности сетевой инфраструктуры на основе графа атак используются следующие сущности:

- узел сети – актив организации, имеющий определенную ценность, представленный на графе атак вершиной графа;
- уязвимость узла – использованная для компрометации узла сети уязвимость, представленная на графе атак группой дуг, направленных к одной вершине, олицетворяющей уязвимый узел сети;
- защитная мера – действие по устранению уязвимости конкретного актива, приводящее к удалению группы дуг на графе атак, олицетворяющих уязвимость узла сети;
- риск компрометации – базовый показатель защищенности узла сети;
- нисходящий риск – топологическая характеристика риска узла, полученная в результате прохождения через все узлы сети, компрометируемые с данного узла;
- уровень риска системы – интегральный показатель риска, основанные на нисходящем риске всех узлов.

Оценка защищенности производится по ориентированному графу с собственной идентификацией дуг (орграфу с идентификацией) следующего вида:

$$G =< V, A, C, vuln >$$

Где V – непустое множество вершин графа; A – множество дуг графа; C – множество успешно проэксплуатированных уязвимостей; $vuln$ – отображение вида:

$$vuln: C \to A$$

При этом возможна ситуация:

$$\exists c_1, c_2 \in C, x, y \in V: (x, y) \in A, vuln(c_1) = vuln(c_2) = (x, y)$$

Множество A отображает возможность компрометации узла y из узла x, а отображение $vuln$ определяет набор уязвимостей, позволяющих осуществить компрометацию. Таким образом, $vuln$ – сюръективное отображение.

Направление дуги задаётся последовательностью следования узлов.

3.1 Показатели защищенности узлов сетевой инфраструктуры

В главе 1 были рассмотрены основные сведения об узле сети, которые может получить любое лицо, имеющее сетевой доступ к целевому узлу. Информация о доменном имени узла, контактной информации сотрудников организации, типе и версии операционной системы, версии запущенных сервисов может быть использована в процессе построения векторов атак, однако её недостаточно для определения риска компрометации конкретного узла сети. Для этой цели в данной работе используется информация о типе узла и запущенных на нем сервисов.

Обладая данной информацией, оценщик, как и злоумышленник, может сделать вывод о важности данного узла для системы даже в отсутствие информации о системе, предоставленной заказчиком.

Используя сведения о типе узла и запущенных на нем сервисов, вычисляется значение базового показателя защищенности узла – риска компрометации.

В Nmap Security Scanner выделено несколько типов узлов [25], представленных в табл.3.1. Каждому из данных типов узлов был сопоставлен коэффициент критичности в зависимости от важности данного узла, принимающий значения от 0 до 1, также содержащийся в табл.3.1.

Таблица 3.1

Описание типов узлов, предоставляемых nmap, и их коэффициента критичности для типовой сетевой инфраструктуры

Тип узла	**Описание**	**CC**
Общего назначения	Эта категория содержит операционные системы общего назначения, такие как Linux и Windows.	0,5
Мост	Мост объединяет две или более подсетей в одну. Работает на более низком уровне, чем роутер.	0,75
Широкополосный маршрутизатор	Устройства этой категории подключают сеть к Интернету по кабелю, ADSL, оптоволоконному кабелю и т.д. Некоторые из этих устройств осуществляют трансляцию сетевых адресов, переадресацию портов и другие услуги.	0,9
Межсетевой Экран	Узлы данной категории осуществляют фильтрацию траффика, идущего в сеть и из неё.	0,8
Игровая консоль	Консоль для видеоигр, такая как Xbox или PlayStation	0,2
Сетевой концентратор	Объединяет сегменты сети, повторно транслируя весь трафик. Концентраторы отличаются от коммутаторов, которые выборочно передают пакеты только в соответствующие пункты назначения.	0,5
Балансировщик нагрузки	Осуществляет распределение входящего траффика по узлам сети, выполняющим одинаковые функции, для снижения нагрузки на отдельные устройства.	0,4

Мультимедийное устройство	Под эту категорию попадают все виды мультимедийного оборудования, в том числе музыкальные проигрыватели, телевизоры, проекторы и аудиосистемы.	0,3
PBX	Осуществляет перенаправление звонков на VoIP или телефоны сети общего пользования.	0,6
PDA	Категория устройств, к которым относятся карманные портативные компьютеры (КПК)	0,5
Телефон	Мобильные телефоны	0,7
Устройство питания	Различные источники питания, такие как источники бесперебойного питания и сетевые фильтры.	0,2
Принтер	Сетевые принтеры, включая принтеры со встроенным сервером печати	0,4
Сервер печати	Сервер, подключающий принтер к сети.	0,4
Прокси-сервер	Прокси любого типа, включая веб-прокси и другие серверы, которые кэшируют данные или понимают протоколы высокого уровня.	0,6
Сервер удаленного управления	Устройства, которые позволяют удаленно контролировать или управлять другим оборудованием.	1
Роутер	Устройства, соединяющие несколько сетей. Отличаются от концентраторов и коммутаторов, поскольку маршрутизируют пакеты между различными сетями, а не расширяют одну сеть.	0,9
Прочие устройства безопасности	Любое сетевое устройство, не попадающее в категорию «Межсетевые экраны». К ним относятся IDS, IPS и т.д.	0,8
Специализированное устройство	Если устройство не попадает ни в одну из других категорий, оно является специализированным.	0,5
Устройства хранения	Устройства хранения данных, такие как магнитные ленты и сетевые устройства хранения данных.	1
Коммутатор	Устройство, осуществляющее ретрансляцию пакетов канального уровня.	0,4
Телекоммуникационное	Устройства, используемые телефонными системами,	0,3

устройство	которые не являются PBX, как голосовая почта и ISDN.	
Терминал	Устройство, главной целью которого является связь с терминальным сервером.	0,8
Терминальный сервер	Устройство, предоставляющее терминальные услуги клиентам сети.	0,95
VoIP адаптер	Устройства, преобразующие протоколы передачи голоса по IP в обычный телефонный трафик.	0,4
VoIP телефон	Телефон с поддержкой VoIP	0,4
WAP	Точки доступа, обеспечивающие беспроводное подключение к сети.	0,75
Веб-камера	Любое устройство, транслирующее изображение или видео.	0,35

Значение коэффициента критичности варьируется в зависимости от важности для атакующего того или иного типа узлов.

Каждому узлу графа сопоставляется список открытых портов и запущенных на них сервисов. Тип сервиса, запущенный на открытом порту, и его версия могут быть также определены с помощью Nmap. Для этого используется команда:

```
nmap -sV -T2 -p- target
```

Добавлено примечани

Полученная в результате выполнения данной команды информация о сервисах узла представлена на рис. 3.1.

```
Host is up (0.0000030s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 131061 closed ports
PORT     STATE         SERVICE
22/tcp   open          ssh
80/tcp   open          http
443/tcp  open          https
3306/tcp open          mysql
8118/tcp open          privoxy
9040/tcp open          tor-trans
9050/tcp open          tor-socks
9475/tcp open          unknown
5300/udp open|filtered hacl-hb
```

Рис.3.1. Информация о сервисах узла

Риск компрометации узла складывается из риска компрометации каждого из сервисов, запущенных на данном узле и умноженных на коэффициент критичности.

Для определения риска узла необходимо задать уровень риска каждого из сервисов. Разделим сервисы на группы:

- Web-сервисы;
- Системы управления базами данных;
- Сервисы удаленного управления рабочей станцией;
- Сервисы цифровой видео- и аудиосвязи;
- Файловые сервисы;
- Почтовые сервисы;
- Прочие сервисы.

В табл.3.2 приведен риск каждой группы сервисов в терминах компрометации узла сети. Параметр риска компрометации может принимать значение от 0 до 100.

Таблица 3.2

Риск компрометации групп сервисов

Группа	**Значение риска**
Системы управления базами данных	90
Сервисы удаленного управления рабочей станцией	85
Файловые сервисы	75
Web-сервисы;	60
Почтовые сервисы	40
Сервисы цифровой видео- и аудиосвязи	40
Прочие сервисы	20

3.2 Методика оценки защищенности сетевой инфраструктуры

Для оценки защищенности сетевой инфраструктуры необходимо в первую очередь провести оценку риска компрометации каждого узла, а также

значения нисходящего риска, который вычисляется как сумма рисков компрометации всех узлов, достижимых из текущего узла. Для оценки риска компрометации узла используется формула:

$$Criticality(x) = device\_type\_coef(x) \sum_{s \in S(x)} service\_cost(s)$$

Где $device\_type\_coef(x)$ – коэффициент критичности узла, зависящий от типа узла x, представленный в таблице 3.1, а $service\_cost$ – риск компрометации сервиса, зависящий от принадлежности сервиса к одной из групп, определенных в таблице 3.2, $S(x)$ – множество сервисов узла x.

После вычисления риска компрометации всех узлов графа атак, необходимо осуществить оценку нисходящего риска каждого узла. Для оценки нисходящего риска узла используется формула:

$$t(x) = \sum_{y \in W(x) \subseteq V} Criticality(y)$$

где $W(x)$ – множество узлов графа, достижимых из x. Фактически нисходящий риск показывает, насколько критично попадание злоумышленника на конкретный узел для всей системы.

Узел y достижим из x тогда и только тогда, когда существует хотя бы 1 путь из x в y. При этом если x является листом графа, то есть не существует ни одной дуги вида (x, y, c), где $x \neq y$, то нисходящий риск данного узла равен риску компрометации.

Для подсчета $t(x)$ необходимо определить множество $W(x)$. Для этого выполняется обход графа в глубину (алгоритм DFS), начиная от узла r_i, и помечаются все достижимые узлы.

Уровень риска системы определяется, как сумма нисходящего риска каждого из узлов, что фактически иллюстрирует насколько уязвима система, при осуществлении атаки на сеть с любого из узлов системы.

Для вычисления уровня риска системы используется следующая формула:

$$T(G) = \sum_{x \in G(V)} t(x)$$

где G – граф атак; $G(V)$ – множество всех точек графа.

Определив риск системы таким образом, полученные на этапе выбора защитных мер рекомендации будут максимально эффективно снижать риски системы вне зависимости от того, где находится точка входа.

3.3 Методика выбора защитных мер

Процесс выбора защитных мер опирается на показатель уровня риска системы. Методика выбора защитных мер заключается в поиске такой уязвимости $c \in C$, при устранении которой произойдет максимально возможное снижение уровня риска системы:

$$\exists\, c \in C : A' = A \backslash \{a \in A |\, vuln(c) = a\}$$

$$C' = C \backslash \{c\}$$

$$vuln' : C' \to A'$$

Добавлено примечани

$$G' = < V, A', C', vuln' >$$

$$|T(G) - T(G')| \to max$$

В результате устранения уязвимости c будут удалены все дуги графа атак, соединяющие любой узел с узлом y посредством уязвимости c, если только не останется другой уязвимости, соединяющей узел x с узлом y. Все дуги графа, эксплуатирующие одинаковые уязвимости можно заранее объединить в группы, тем самым осуществляя удаление уязвимости за одну итерацию. Использование данного подхода позволит максимально снизить риск компрометации сети с любого из узлов системы, присутствующего в графе атак.

Так как в наивном исполнении построение графа атак и пересчет нисходящего риска всех узлов занимает значительное время, были разработаны методы оптимизации и предварительных вычислений

(предвычислений), призванные сократить время повторного вычисления нисходящего риска каждого узла.

Представим, что каждый узел исходного графа принадлежит подграфу одного из двух типов – компоненте сильной связности или модифицированному N-арному дереву.

3.3.1Компонента сильной связности

Орграф 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. На рис.3.2 изображен орграф, в котором найдены все три компоненты сильной связности.

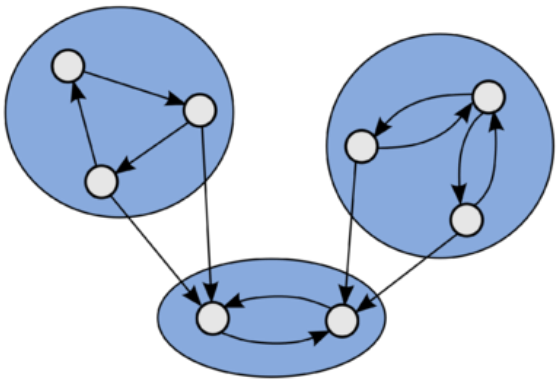

Рис.3.2. Компоненты сильной связности

Поиск компонент сильной связности может проводиться, например, с помощью алгоритма Косарайю [26], который выполняется в несколько шагов:

1. Инвертируются дуги исходного ориентированного графа;
2. Выполняется DFS на обращенном графе с запоминанием порядка выхода из вершин;
3. Выполняется DFS на исходном графе, в очередной раз выбирая не посещенную вершину с максимальным номером в векторе, полученном в ш.2;
4. Полученные в результате выполнения ш.3 деревья и являются сильно связанными компонентами.

3.3.2Модифицированное N-арное дерево

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. Значение N может отличаться от дерева к дереву.

Пример N-арного дерева приведен на рис.3.3

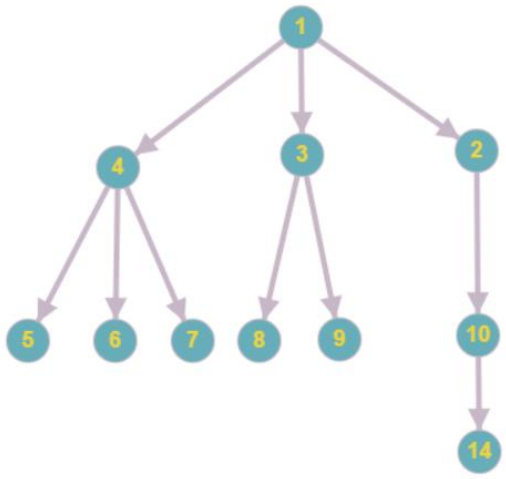

Рис.3.3. N-арное дерево

Для использования N-арных деревьев в качестве подграфов графа атак необходимо ввести несколько дополнительных условий:

1. В данный подграф включаются только те узлы, входящие дуги которых направлены от узлов того же дерева;
2. Только корень может иметь входные дуги, направленные из других подграфов графа атак;
3. При удалении всех исходящих дуг данного подграфа, направленных на узлы, расположенные вне данного подграфа, корень является шарниром, то есть узлом, при удалении которого возрастет число компонент связности исходного графа.

Введение данных свойств позволит обеспечить данному типу подграфов единственную точку входа – через корень дерева.

Поиск модифицированных N-арных деревьев графа атак происходит в несколько шагов:

1. Инвертируются дуги исходного ориентированного графа;

2. Все листья инвертированного графа добавляются в список узлов для посещения;

3. Для всех узлов из списка поочередно осуществляется обход в ширину вплоть до узлов, имеющих более одной исходящей дуги в инвертированном графе. При окончании обхода узел удаляется из списка для посещения;

4. Для каждого узла из списка строится дерево посещенных узлов;

5. деревья, имеющие одинаковые элементы объединяются в одно дерево;

6. Если все входящие дуги построенного дерева исходят из одного дерева, такие деревья объединяются;

7. Все узлы, к которым направлены дуги из построенных графов заносятся в список для посещения;

8. Пока в списке для посещений присутствуют узлы, алгоритм повторяется с ш.2.

3.3.3Свойства полученных подграфов

По определению рассматриваемых структур, а также благодаря ограничениям, введенным над N-арным деревом, данные подграфы имеют следующие свойства:

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

$$t(x) = t(y), \forall\, sc \in SC, \forall x, y \in sc$$

где SC – множество областей сильной связности

Следовательно, для каждой области сильной связности достаточно посчитать значение ниспадающего риска один раз.

- Из определения компоненты сильной связности и модифицированного N-арного дерева следует, что, попав в любую из данных

компонент, гарантируется прохождение по всем узлам подграфа. При этом в модифицированном 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:

Пусть NC – множество наборов вершин модифицированных N-арных деревьев графа атак; SC – множество областей сильной связности графа атак.

$$Criticality(s) = \sum_{x \in s} Criticality(x), \forall\, s \in NC \cup SC$$

где s – множество узлов компоненты сильной связности, либо модифицированного N-арного графа; x – узел подграфа.

- Дуги исходного графа делятся на 3 группы:
  - Дуги, принадлежащие компонентам сильной связности;
  - Дуги, принадлежащие N-арным деревьям;
  - Дуги, 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 модифицированным N-арным деревьям, если удаление ребра повлекло изменение структуры подграфа. Также необходимо осуществить новые расчёты параметров риска компрометации подграфов и, в случае компоненты сильной связности, нисходящего риска подграфа.
2. Если ребро соединяло две компоненты, нет необходимости в пересчете параметра риска компрометации подграфов.

Исходя из вышеописанных свойств можно провести следующее преобразование: представим все обнаруженные подграфы как узлы нового графа, которые включают в себя все входящие и все исходящие дуги

исходного подграфа. При этом значение риска компрометации узла соответствует значению риска компрометации подграфа.

В случае компоненты сильной связности данному узлу также присваивается значение нисходяшего риска:

$$t(y) = |sc| * t(x), x \in sc, sc \in SC$$

где y – новый узел графа.

Добавлено примечание
математическое описание не
дугами.

3.3.4Алгоритм выбора защитных мер

Поиск наиболее эффективной меры защиты – трудоёмкий процесс. Поэтому необходимо выполнить некоторые предвычисления на графе атак, такие как поиск компонент сильной связности и модифицированных N-арных графов

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 (см. рис.3.4).

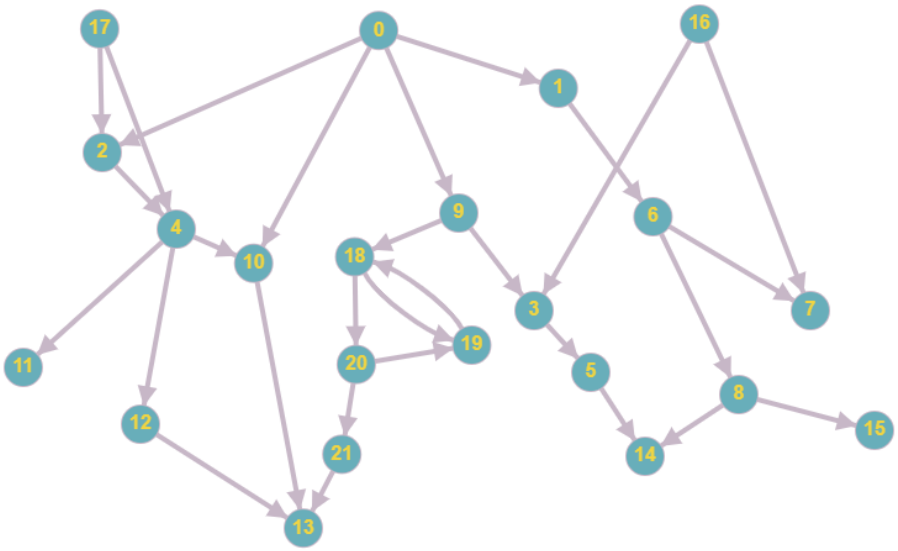

Рис.3.4. Пример ориентированного графа

В данном графе присутствует одна компонента сильной связности – (18,19,20). Данная компонента заменяется единственным узлом с идентификатором 22. Далее осуществляется поиск модифицированных N-арных деревьев по алгоритму, описанному в п.3.3.2.

В результате работы первой итерации алгоритма (до заполнения списка ожидающих новыми узлами) будет создано несколько подграфов, представленных на рис.3.5

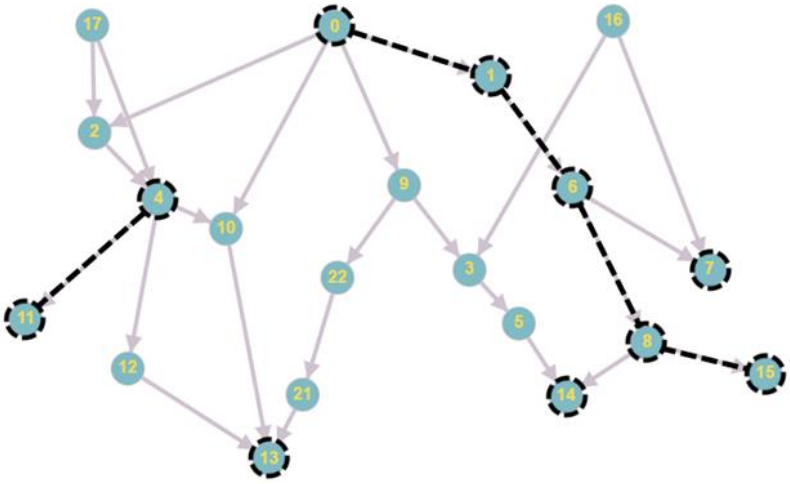

Рис.3.5. Созданные за 1 итерацию подграфы

На следующем этапе будет выполнен обход узлов с номерами: 2, 17, 12, 10, 21, 5, 16. Результат работы второй итерации представлен на рис.3.6.

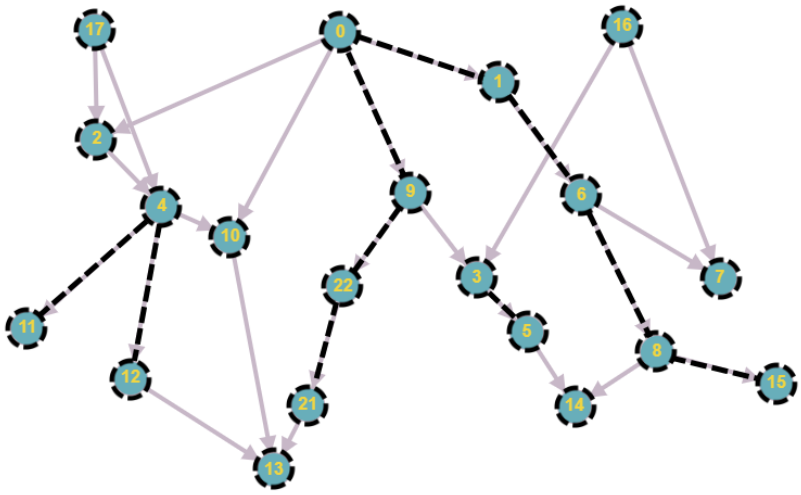

Рис.3.6. Результат работы второй итерации алгоритма

На каждой итерации производилось объединение деревьев в соответствии со свойствами. При этом узел 22 является компонентой сильной связности.

Далее производится создание новых узлов по правилам, описанным в п.3.3.3. Подграф (4,11,12) преобразовывается в вершину 23, подграф (21, 22, 9, 0, 1, 6, 8, 15) – в 24, подграф (3,5) – в 25. Преобразованный граф представлен на рис.3.7.

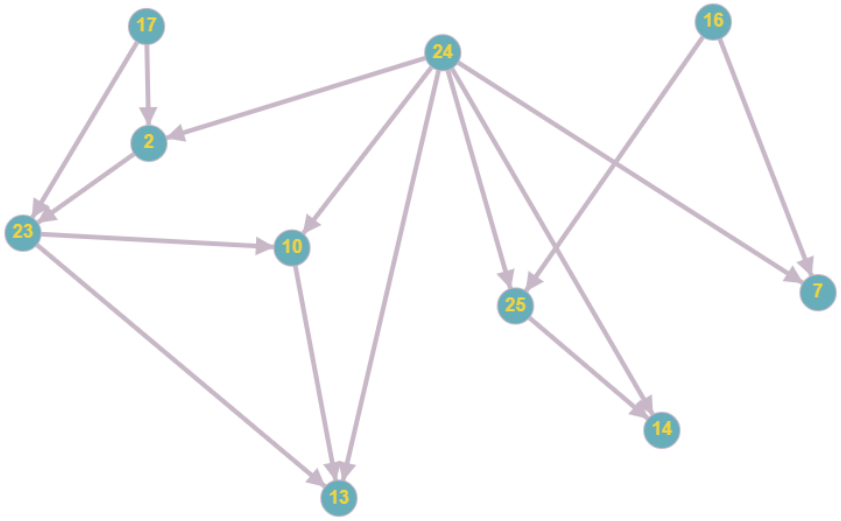

Рис.3.7. Вид преобразованного графа атак

В результате такого преобразования граф избавляется от циклов, затрудняющих оценку защищенности системы, а также обеспечивается набором оптимизаций по вычислению показателей защищенности.

Далее проводится поиск наиболее эффективной защитной меры с использованием результатов предварительных вычислений, что позволяет значительно ускорить процесс оценки рисков, так рассматриваемая структура сетевой организации очень часто выглядит так, что $|A| \gg |V|$. Следовательно, перерасчету критичности узлов и нисходящего риска на каждом этапе весьма затруднителен.

Применение оптимизаций позволит не только точно определить, какие подграфы затронуло введение защитной меры, но и осуществить оперативный перерасчет всех показателей защищенности.

4. РЕАЛИЗАЦИЯ СИСТЕМЫ ОЦЕНКИ ЗАЩИЩЕННОСТИ И ВЫБОРА ЗАЩИТНЫХ МЕР

Для реализации автоматизированного системы, осуществляющей оценку защищенности сетевой инфраструктуры и выбор защитных мер на основе графа атак был выбран язык Python ввиду его кроссплатформенности, что позволит использовать систему в любой современной ОС.

Функциональность системы поддерживается 4 основными компонентами:

- Компонентом обработки данных;
- Компонентом оценки защищенности;
- Компонентом выбора контрмер;
- Компонентом визуализации.

4.1 Архитектура системы оценки защищенности и выбора защитных мер

Архитектура реализованной системы оценки защищенности сетевой инфраструктуры и выбора защитных мер выглядит следующим образом (см. рис.4.1).

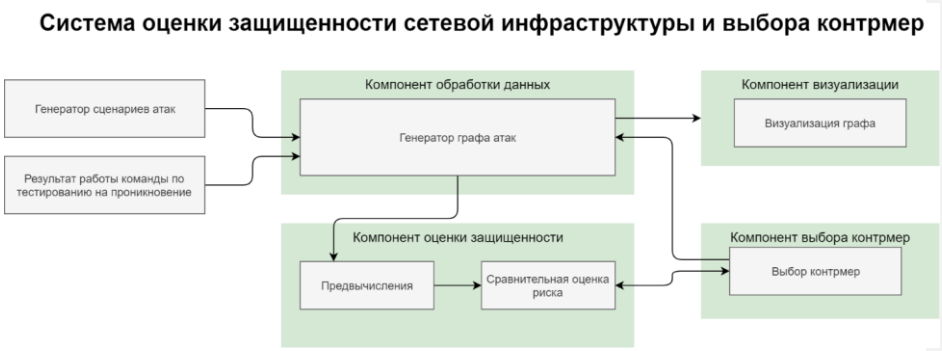

Рис.4.1. Архитектура разработанной системы

Компонент обработки данных отвечает за преобразование входных данных и данных, поступающих с компонента выбора контрмер, в набор классов, реализующих операции над графом атак.

Компонент оценки защищенности производит вычисление основных показателей риска и производит сравнительную оценку текущего состояния и состояния, которому предшествовало удаление одной из существующих уязвимостей.

В компоненте выбора контрмер реализуются алгоритмы поиска уязвимости, при удалении которой максимально снизится уровень риска системы.

Компонент визуализации осуществляет отправку данных о графе атак на сервис 3vis, разработанный компанией NeoBIT для полноценной визуализации графа атак.

Разработанный программный комплекс осуществляет итеративный поиск оптимальных контрмер. Это значит, что на вход подаётся количество уязвимостей, которые необходимо удалить, данные, имитирующие граф атак, описанный в п.3.3, а на выходе имеется набор пар (r, c), где r – целевой узел, а c – одна из его уязвимостей, упорядоченные по приоритету устранения.

4.1.1 Компонент обработки данных

Как уже было отмечено выше, компонент обработки данных оперирует набором параметров, описывающих граф атак. Данные в систему поступают в формате языка описания графов (DOT).

Структура графа на языке DOT описывается в виде списка субграфов, каждый из которых представляет собой конструкцию:

```
graph %имя_графа% {
}
```

Внутри такой конструкции находятся инструкции и комментарии, описывающие субграф. Инструкции могут описывать вершины и ребра целевого графа и разделяются точкой с запятой.

Язык DOT определяет ориентированные и неориентированные типы графов. Для описания ориентированного графа используется конструкция:

```
digraph %имя_графа% {
```

```
    a -> b -> c;
    b -> d;
}
```

Сочетание символов –> используется для обозначения ориентированности, в неориентированном графе используется – –.

При описании графа на языке DOT можно также задать использовать атрибуты, позволяющие задать параметры вершин и ребер. Атрибуты описываются парами ключ=значение, заключенными в квадратные скобки. Каждый элемент графа может иметь несколько атрибутов, разделенных пробелом.

С учетом всего вышеописанного граф на входе в компонент обработки приобретает вид:

```
digraph attack_graph {
[узел 1] [compromitation_risk=значение риска компрометации];
[узел 2] [compromitation_risk=значение риска компрометации];
……
[узел 1] -> [узел 2]  [key=[идентификатор уязвимости]];
}
```

Значение `[узел 1], [узел 2], [идентификатор уязвимости]` могут принимать любые уникальные значения.

Данный компонент проводит парсинг входных данных и строит граф атак, который затем используется остальными компонентами для оценки защищенности, выработки контрмер и построения визуальной модели.

4.1.2 Компонент оценки защищенности

Компонент оценки защищенности состоит из двух основных функциональных блоков: блока предвычислений и блока сравнительной оценки защищенности.

Блок предвычислений выполняет основные вычисления, необходимые для оптимизации процесса поиска наиболее эффективных защитных мер. На этапе предвычислений осуществляется:

1. Поиск компонент сильной связности;
2. Определение основных параметров компонент сильной связности – риск компрометации компоненты, нисходящий риск компоненты;
3. Создание псевдо-узлов графа, содержащих все входящие и исходящие дуги исходных компонент сильной связности; данным узлам присваиваются значения риска компрометации и нисходящего риска компонент;
4. Поиск модифицированных N-арных графов;
5. Определение основных параметров модифицированных N-арных графов – риск компрометации компоненты, нисходящий риск компоненты;
6. Создание псевдо-узлов графа, содержащих входящие и исходящие дуги исходных подграфов с присвоением им значения риска компрометации и нисходящего риска подграфа;
7. Каждому узлу присваивается ссылка на его подграф.

Операции определения принадлежности узла подграфу, поиска дуг подграфа и т.п. выполняются за 1 операцию, что достигается использованием хэш-таблиц.

Блок сравнительной оценки защищенности используется компонентом оценки защищенности для определения уровня риска системы, а также компонентом выбора контрмер для оценки эффективности введенной меры защиты.

Блок сравнительной оценки осуществляет расчет уровня риска переданного ему графа атак. Для этого осуществляется обход в глубину всех вершин графа, нисходящий риск которых не определен. Фактически это все вершины, которые не лежат внутри компонент сильной связности или те, которые затронуло внедрение защитной меры. В процессе обхода используются элементы оптимизации в виде предвычислений и псевдо-узлов, чтобы ускорить вычисление нисходящего риска.

4.1.3 Компонент выбора контрмер

Компонент выбора контрмер осуществляет перебор всех возможных защитных мер, которые могут быть введены на графе атак. Для этого создаётся копия текущего графа, в котором поочередно устраняются все уязвимости. Устранение одной конкретной уязвимости приводит к тому, что удаляются все дуги, направленные в сторону уязвимого узла, которые эксплуатируют только данную уязвимость.

Следует уточнить, что, если по одной дуге возможно использование нескольких уязвимостей, такая дуга не удаляется с графа атак до тех пор, пока не будут удалены все уязвимости данной дуги.

Удаление дуг графа приводит к необходимости перерасчета нисходящего риска всех узлов, которые затронуло её удаление. Получив на вход новый граф атак, компонент оценки защищенности осуществляет локализацию изменений, перестроение подграфов, которые затронуло удаление дуг, а также перерасчет нисходящего риска затронутых узлов.

Новый уровень риска системы вычисляется как сумма по неизмененным нисходящим рискам незатронутых узлов и новым значениям нисходящих рисков затронутых узлов.

В худшем случае перерасчету будут подлежать все узлы графа, однако, благодаря ранее проведенным предвычислениям полный расчет нисходящих рисков будет выполняться намного быстрее.

После обнаружения наиболее эффективной контрмеры текущая копия графа отправляется в компонент обработки данных для принятия в качестве основной. Найденная уязвимость и затронутый узел сохраняются для составления статистики. Компонент обработки данных принимает решение о продолжении процесса анализа, либо отправляет граф в компонент визуализации.

4.1.4 Компонент визуализации

Компонент визуализации осуществляет построение графической версии графа атак в сервисе «3VIS» компании NeoBIT.

Для взаимодействия с сервисом используется протокол HTTP.

Для работы с сервисом необходимо получить так называемый access token, который подтверждает права пользователя на взаимодействие с сервисом. Данный токен добавляется в заголовок каждого запроса, отправляемого на сервер.

Заголовок имеет вид:

```
Authorization: Bearer [access token]
```

В табл.4.1 приведены все используемые запросы.

Таблица 4.1

Запросы к сервису 3vis

Тип запроса	**Путь**	**Параметры**	**Описание**
POST	3vis.neobit.ru/api/projects	{“name”:имя графа}	Создание графа с заданными именем. В ответ приходит идентификатор графа.
DELETE	3vis.neobit.ru/api/projects	{“pid”:id графа}	Удаление графа с заданным pid
POST	3vis.neobit.ru/api/graph/node	{"node": { "name": имя узла, "type": тип узла }, "pid": идентификатор графа, "needCompare": False}	Создаёт узел графа определенного типа. В ответ приходит идентификатор узла.
POST	3vis.neobit.ru/api/graph/link	{"sourceId": id первого узла, "targetId": id второго узла, "pid": pid графа}	Создаёт дугу от первого узла ко второму в конкретном графе

Каждому узлу присваивается определенный тип. Типы создаются пользователями и могут иметь собственный идентификатор типа и название. Типы используются для поиска скрытых связей в графах.

Пример созданного с помощью сервиса графа представлен на рис.4.2

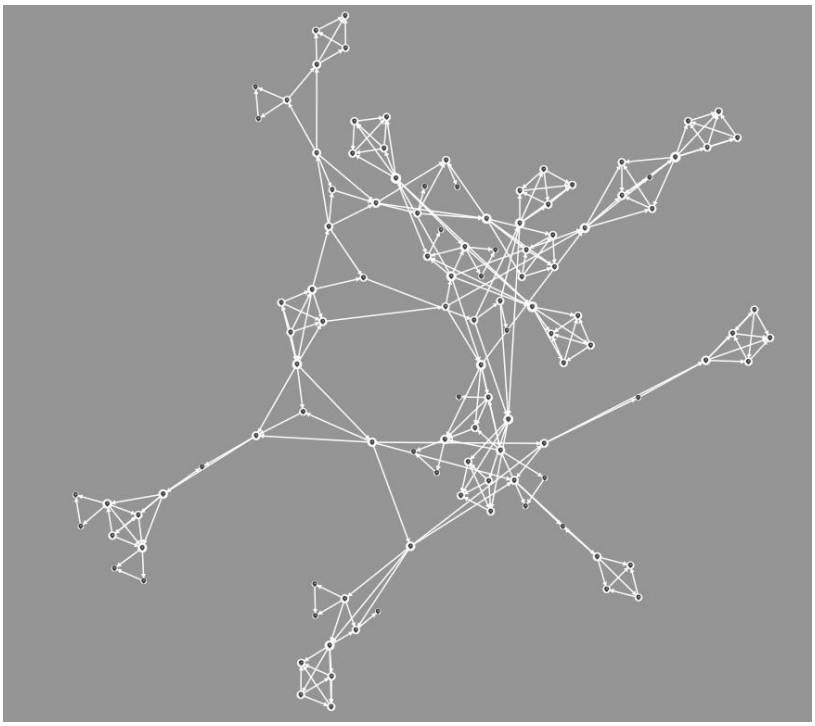

Рис.4.2. Иллюстрация возможностей сервиса 3VIS

Возможности данного сервиса также позволяют продемонстрировать конкретное влияние каждого узла на граф атак, путем выделения его дуг (см. рис.4.3), что может быть продемонстрировано заказчику при сдаче проекта по тестированию на проникновение.

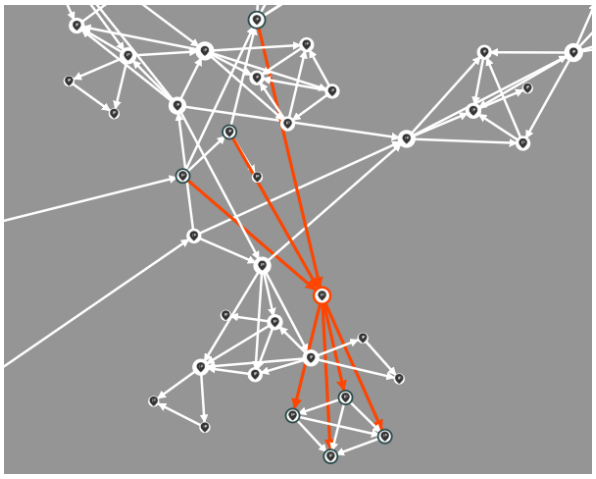

Рис.4.3. Подсветка дуг конкретного узла

4.2 Пример выработки рекомендаций

Рассмотрим процесс работы системы на примере графа атак, приведенного на рис.4.4.

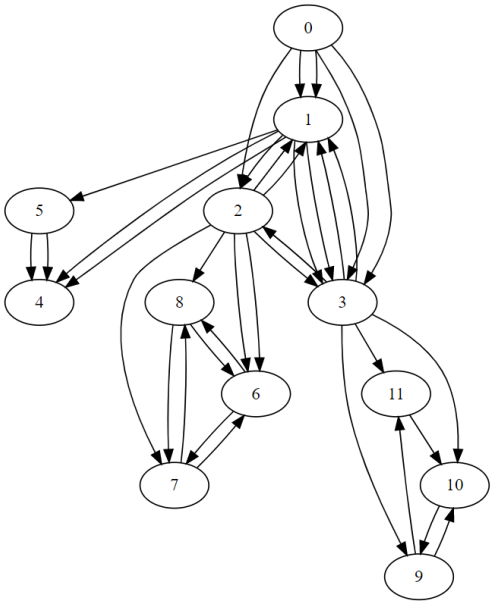

Рис.4.4. Пример графа потенциальных атак

Многореберность используется для обозначения количества уязвимостей, которые может проэксплуатировать один узел, чтобы попасть на другой.

Вершины данного графа имеют порядковые номера от 0 до 11. В соответствии с номером вершины в табл.4.2 определены риски компрометации каждого из узлов графа и идентификаторы уязвимостей узлов графа.

Таблица 4.2

Описание узлов графа и их уязвимостей

Идентификатор узла	**Риск компрометации**	**Идентификаторы уязвимостей**
0	20	[‘CVE_0_0’]
1	10	['CVE_1_0', 'CVE_1_1']
2	10	['CVE_2_0']
3	80	['CVE_3_0', 'CVE_3_1']
4	70	['CVE_4_0', 'CVE_4_1']
5	80	['CVE_5_0']
6	40	['CVE_6_0', 'CVE_6_1']
7	10	['CVE_7_0']
8	30	['CVE_8_0']
9	80	['CVE_9_0']
10	60	['CVE_10_0']
11	70	['CVE_11_0']

Узел 0 представляет собой точку входа, уязвимость которой может проэксплуатировать только атакующий. Идентификаторы уязвимостей имеют следующий формат:

`CVE_[идентификатор узла, на котором найдена уязвимость]_[порядковый номер обнаруженной уязвимости].`

Рассмотрим, как меняется уровень риска на протяжении поиска 5 наиболее критичных уязвимостей:

0. T = 3270;
1. T = 2720, c = CVE_9_0;
2. T = 2240, c = CVE_1_1;
3. T = 1880, c = CVE_10_0;
4. T = 1260, c = CVE_11_0;
5. T = 1020, c = CVE_3_1.

После устранения данных уязвимостей граф приобрел следующий вид (см. рис.4.5).

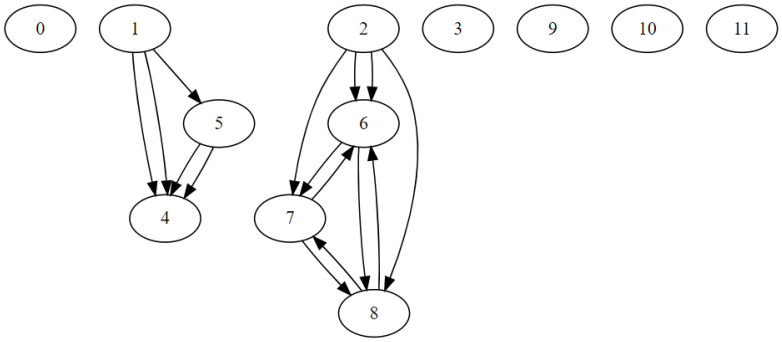

Рис.4.5. Граф атак после устранения 5 уязвимостей

Как видно из рис.4.5, удаление наиболее критических уязвимостей повлекло разложение графа на небольшие подграфы, благодаря чему значительно уменьшился уровень риска системы.

4.3 Оценка эффективности разработанных методик

Эксперименты проводились на сгенерированных графах атак различного размера для двух реализаций разработанных методик: наивной, то есть не учитывающей оптимизаций, основанных на разбиении исходного графа атак на 2 типа подграфов и полной, включающей оптимизации.

Эксперименты показали, что введенные меры оптимизации вычислении действительно эффективны, что отображено на рис.4.6.

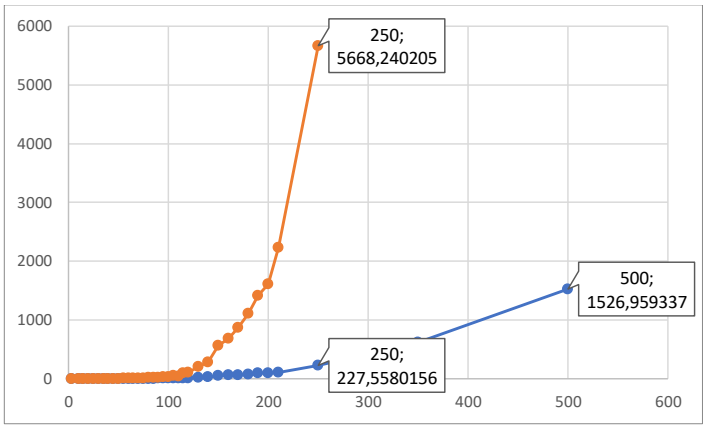

Рис.4.6. Сравнение времени работы двух реализаций

Наблюдаемы на рис.4.6 графики линейны, однако угол наклона у графика, использующего оптимизации заметно ниже.

В процессе проведения экспериментов было установлено, что время вычисления сильно зависит от стркутуры графа. Чем больше в графе компонент сильной связности, тем, следовательно, больше разница во времени выполнения между методами.

Также невозможно точно оценить выигрыш от введения оптимизаций, так как у двух графов с одинаковым числом узлов может быть различное число уязвимостей, дуг и связи между узлами тоже могут отличаться.

По полученным данным, приведенным в Приложении Б, был осуществлен анализ коэффициента увеличения времени выполнения, который рассчитывался как:

$$K(n) = \frac{t_n}{t_{n-1}}$$

В результате была найдена средняя величина увеличения времени выполнения обеих реализаций при шаге в 10 узлов. Она составила 1,39 для

4.4 Сравнение с существующими методиками

В табл.4.3 представлен сравнительный анализ наиболее популярных на сегодняшний день методик, а также методики, предложенной в данной работе.

Таблица 4.3

Сравнительный анализ методиц оценки защищенности и выбора контрмер

Критерии сравнения	**Разработанная методика**	**CRAMM**	**ГРИФ**	**RiskWatch**	**FRAP**	**Дойникова** [13]
Способы измерения величин рисков						
Качественная оценка риска	-	+	+	+	+	+
Количественная оценка риска	+	-	+	+	-	+
Способы управления						
Качественное ранжирование рисков	-	+	+	+	+	+
Количественное ранжирование рисков	+	-	+	+	-	+
Ограничения						
Не требует прохождения опросников	+	-	+	-	-	+
Не требуется информация о бизнес-процессах организации	+	+	-	-	-	- используется критичность бизнес-сервисов
Не требует взаимодействия	+	-	-		-	+-

с человеком						
Не требуется информация об инцидентах ИБ	+	+	+	-	+	+
Подлежит автоматизации	+	-	-	-	-	+-

Разработанная методика может быть использована в качестве дополнительного модуля в любой системе анализа защищенности, на неё не накладываются никакие ограничения об информированности оценщика и она полностью подлежит автоматизации.

ЗАКЛЮЧЕНИЕ

Оценка защищенности сетевой инфраструктуры и выбор защитных мер на основе анализа графа атак без наличия дополнительной информации о системе является важной задачей информационной безопасности. Реализованная в данной работе система эффективно решает поставленную задачу и может быть использована в процессе проведения анализа защищенности сетевой инфраструктуры. Основные результаты работы состоят в следующем:

1) Исследованы основные способы идентификации уязвимостей и узлов сетевой инфраструктуры с целью получения максимально возможной информации о системе, которая пригодна для использования в качестве показателей защищенности в процессе оценки защищенности системы.

2) Проанализированы наиболее распространенные методики оценки защищенности и выбора защитных мер, в том числе использующие графы атак. Определены основные преимущества и недостатки данных методик, которые воплощены в требованиях к разработанной системе.

3) Разработанные методики оценки защищенности и выбора защитных мер, на основе предложенных показателей, предлагают принципиально новый способ оценки защищенности системы от проникновений. При внесении незначительных изменений, данные методики могут быть использованы для оценки защищенности системы от других угроз подобного рода. Предложенные алгоритмы оптимизации вычислений на графе атак на порядок уменьшают время нахождения наиболее эффективной контрмеры по сравнению с наивным методом поиска.

4) Реализована автоматизированная система оценки защищенности сетевой инфраструктуры и выбора защитных мер. Основным отличием является применение оригинальных методик оценки защищенности сетевой инфраструктуры и выбора защитных мер.

Реализованная система может успешно решать задачу оценки защищенности и поиска защитных мер и применяться как дополнение к

системам автоматизированного тестирования на проникновение с последующей выработкой рекомендаций.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Positive Technologies pentesters breach network perimeter at 92 percent of companies // PTSECURITY.COM: основной сайт Positive Technologies. – URL: https://www.ptsecurity.com/ww-en/analytics/corp-vulnerabilities-2019/ – (дата обращения 11.10.2019).

2. Criteria for Selecting an Information Security Risk Assessment Methodology: Qualitative, Quantitative, or Mixed // TCDI.COM: information security forum. – URL: https://www.tcdi.com/criteria-for-selecting-an-information-security-risk-assessment-methodology-qualitative-quantitative-or-mixed/. – (дата обращения 21.10.2019)

3. Брюховецкая Н.Е., Педерсен И.А. Методология оценки рисков предприятия//Стратегия и механизмы регулирования промышленного развития. – 2011. – С. 58.

4. Visintine V. Global information assurance certification paper// SANS Institute – 2003. – P. 13.

5. Caralli R.A., Stevens J.F., Young L.R., Wilson W.R. Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process. – 2007. – P.9.

6. Peltier T. R. Information security risk analysis, Third Edition – New York: CRC Press, 2010. – P. 456.

7. Resources for Security Risk Analysis, ISO 17799 / BS7799 Security Policies & Security Audit // SECURITYAUDITOR.NET: сайт компании Risk Associates. – URL: https://www.securityauditor.net/. – (дата обращения 21.10.2019).

8. Information technology — Security techniques — Code of practice for information security management // ISO/IEC 17799. – 2005.

9. Fabbri J. Risk Scoring Methodology // RISKWATCH.COM: risk watch website. – URL: https://riskwatch.com/2019/07/31/risk-scoring-methodology/. – (дата обращения 25.10.2019)

10. Комплексное решение для управления информационной безопасностью // DSEC.RU: digital security company website. – URL: https://dsec.ru/press-release/digital-security-office-2005-kompleksnoe-reshenie-dlya-upravleniya-informatsionnoj-bezopasnostyu/. – (дата обращения 02.11.2019).

11. Астахов А.М. Искусство управления информационными рисками – Москва: ДМК Пресс, 2010 – С. 312.

12. CRAMM (CCTA Risk Analysis and Management Method) // MANAGEMENTMANIA.COM: management blogs. – URL: https://managementmania.com/en/cramm-ccta-risk-analysis-and-management-method. – (дата обращения 5.11.2019).

13. Дойникова Е. В. Методики и программный компонент оценки рисков на основе графов атак для систем управления информацией и событиями безопасности // Информационно-управляющие системы. – СПБ: Политехника, 2016. – С. 54

14. Ingols K. Practical attack graph generation for network defense // Proceedings of 22nd Annual Conference on the Computer Security Applications (Miami Beach, FL, 2006). – IEEE, 2006. – P. 121.

15. Жижелев А. В., Панфилов А. П., Язов Ю. К., Батищев Р. В. К оценке эффективности защиты информации в телекоммуникационных системах посредством нечетких множеств // Известия вузов. Приборостроение. – 2003. – Т. 46. – № 7. – С. 22.

16. Kanoun W. Automated reaction based on risk analysis and attackers skills in intrusion detection systems // Proceedings of the Third International Conference on Risks and Security of Internet and Systems (28-30 Oct. 2008). – P. 117–124.

17. White Paper. 6 key risk management metrics for controlling cyber security // CDN2.HUBSPOT.NET: file sharing service. – URL: https://cdn2.hubspot.net/hubfs/91381/2019 Collateral/White Papers/Controlling

Cyber Security/6 Key Risk Management Metrics for Controlling Cyber Security .pdf. – (дата обращения 15.11.2019).

18. The Center for Internet Security. The CIS Security Metrics // The Center for Internet Security. – 2009. – P.175.

19. Дойникова Е. В. Показатели и методики оценки защищенности компьютерных сетей на основе графов атак и графов зависимостей сервисов // Труды СПИИРАН. – СПб: Наука, 2013. – Вып. 26. – С. 54–68.

20. Mell P., Scarfone K., Romanosky S. A Complete Guide to the Common Vulnerability Scoring System Version 2.0 // NIST.GOV: an official website of the USA government. – URL: https://www.nist.gov/publications/complete-guide-common-vulnerability-scoring-system-version-20. – (дата обращения 15.11.2019).

21. Stan O., Bitton R., Ezrets M., Dadon M., Inokuchi M., Ohta Y., Yagyu T., Elovici Y., ShabtaiHeuristic A. Heuristic Approach Towards Countermeasure Selection using Attack Graph // DEEPAI.ORG: artificial intelligence researches. – URL: https://deepai.org/publication/heuristic-approach-towards-countermeasure-selection-using-attack-graphs. – (дата обращения 29.11.2019).

22. Cuppens F., Autrel F., Bouzida Y., Garcia J., Gombault S., Sans T. Anti-correlation as a criterion to select appropriate counter-measures in an intrusion detection framework // Annales Des Télécommunications. – 2006. – Vol. 61. – Iss. 1-2. – P.197-217

23. Lippmann R. P. Validating and restoring defense in depth using attack graphs // MILCOM 2006 - 2006 IEEE Military Communications conference. – 2006.

24. Минаев В.А., Модестов А.А, Кухаренко Д.И. Основные теоретические положения структуризации частных показателей защиты информации с целью интегральной оценки защищенности информационно-телекоммуникационных систем // CYBERLENINKA.RU: исследования в области информационной безопасности. – URL:

https://cyberleninka.ru/article/n/osnovnye-teoreticheskie-polozheniya-strukturizatsii-chastnyh-pokazateley-zaschity-informatsii-s-tselyu-integralnoy-otsenki. – (дата обращения 14.12.2019).

25. Lyon G. Remote OS Detection // The Official Nmap Project Guide to Network Discovery and Security Scanning. – 2011.

26. Strongly connected components // GEEKSFORGEEKS.ORG: a computer science portal. – URL: https://www.geeksforgeeks.org/strongly-connected-components/. – (дата обращения 16.12.2019).

27. Cyber threat map [Электронный ресурс]. – 2019. – Режим доступа: https://cybermap.kaspersky.com/stats.

28. ГОСТ Р 53114-2008 Защита информации. Обеспечение информационной безопасности в организации. Основные термины и определения [Электронный ресурс]. – 2009. – Режим доступа: http://docs.cntd.ru/document/1200075565

29. Mell P., Karen S., Romanosky S. A Complete Guide to the Common Vulnerability Scoring System Version 2.0 [Электронный ресурс]. – 2007. – Режим доступа: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51198.

30. Федорченко А. В., Чечулин А. А., Котенко И. В. Исследование открытых баз уязвимостей и оценка возможности их применения в системах анализа защищенности компьютерных систем и сетей // Информационно-управляющие системы. – 2014. – № 5. – С. 72–79.

31. Банк данных угроз безопасности информации [Электронный ресурс]. – 2019. – Режим доступа: https://bdu.fstec.ru/vul.

32. About CVE [Электронный ресурс]. – 1999. – Режим доступа: https://cve.mitre.org/about/.

33. Bellis E. Using database to automate Assessment and Remediation [Электронный ресурс]. – 2013. – Режим доступа: https://www.kennasecurity.com/blog/using-databases-automate-assessment-remediation/.

34. N. Mansourov, D. Campara, System assurance: beyond detecting vulnerabilities. – Burlington: Elsevier, 2010. – P 161.

35. Secunia Research [Электронный ресурс]. – 2019. – Режим доступа: https://www.flexera.com/products/operations/software-vulnerability-research/secunia-research.html.

36. Гнедин Е., Сценарий для взлома. Разбираем типовые сценарии атак на корпоративные сети [Электронный ресурс]. – 2017 – Режим доступа: https://xakep.ru/2017/04/10/hacking-attack-types/.

37. Network Security: Vulnerabilities vs. Exploits [Электронный ресурс]. – 2018. – Режим доступа: https://www.gizmosphere.org/network-security-vulnerabilities-vs-exploits/.

38. Эксплойты [Электронный ресурс]. – 2015. – Режим доступа: https://www.anti-malware.ru/threats/exploits.

39. Top 10 Exploit Databases for Finding Vulnerabilities [Электронный ресурс]. – 2018. – Режим доступа: https://null-byte.wonderhowto.com/how-to/top-10-exploit-databases-for-finding-vulnerabilities-0189314/.

40. Муханова А. А. Классификация угроз и уязвимостей информационной безопасности в корпоративных системах / А. А. Муханова, А. В Ревнивых, А. М Федотов // Вестник НГУ. Сер.: Информационные технологии. – 2013. – Т. 11. – № 2. – С. 55-72.