

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО РЫБОЛОВСТВУ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Калининградский государственный технический университет»

Балтийская государственная академия рыбопромыслового флота

В.В. Подтопельный

**АУДИТ
ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ**

Учебное пособие
для студентов специальности 10.05.03
«Информационная безопасность
автоматизированных систем»
всех форм обучения

Часть 1

Калининград
Издательство БГАРФ
2023

УДК 004.56 (075)

Подтопельный, В.В. Аудит информационной безопасности. Часть 1: учебное пособие для студентов специальности 10.05.03 «Информационная безопасность автоматизированных систем» всех форм обучения / В.В. Подтопельный; БГАРФ ФГБОУ ВО «КГТУ». – Калининград: Издательство БГАРФ. – 2023. – 171 с. – Библиогр.: с. 170–171. – ISBN 978-5-7481-0514-9. – Текст: непосредственный.

ISBN 978-5-7481-0514-9

Учебное пособие включает в себя первую часть комплекса рассмотрения теоретических вопросов в области защиты информации и практических заданий по дисциплине «Аудит информационной безопасности».

Пособие предназначено для студентов 4 – 5 курсов специальности 10.05.03 «Информационная безопасность автоматизированных систем» и смежных специальностей, содержит теоретические сведения о процессах аудита автоматизированных систем.

Ил. 200, табл. 6, библиогр. – 14 назв.

Рекомендовано в качестве учебного пособия редакционно-издательским советом Балтийской государственной академии рыбопромыслового флота ФГБОУ ВО «КГТУ» 23.01.2023 г., протокол № 01.

Рецензенты: *Великите Н.Я.*, кандидат физико-математических наук, доцент, зав. кафедрой информационной безопасности БГАРФ ФГБОУ ВО «КГТУ»;
Ветров И.А., кандидат технических наук, доцент, доцент ИФМН и ИТ БФУ им. И. Канта

ISBN 978-5-7481-0514-9

УДК 004.56 (075)

© БГАРФ ФГБОУ ВО «КГТУ», 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. ОСОБЕННОСТИ АУДИТА СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ	5
1.1. Аудит информационной безопасности сетевой инфраструктуры	5
1.2. Особенности аудита информационной безопасности удаленных автоматизированных рабочих мест	8
1.3. Особенности аудита информационной безопасности с применением актуальной методики оценки угроз	14
1.4. Сравнительный анализ технологий аудита информационной безопасности сетевой инфраструктуры диспетчерского уровня АСУТП	23
1.5. Особенности сбора данных в многомодульной системе обнаружения вторжений	30
1.6. Особенности аудита информационной безопасности АСУТП	33
1.7. Проблемы аудита информационной безопасности АСУТП	39
1.8. Особенности подготовки активного аудита информационной безопасности АСУТП	47
2. ПРАКТИЧЕСКИЕ АСПЕКТЫ ПРОВЕДЕНИЯ АУДИТА	54
2.1. Пен-тест компонентов ИС при помощи Infection Monkey	54
2.2. Использование PowerShell для аудита ОС Windows	71
2.3. Реализация журнала аудита с помощью PowerShell	114
2.4. Аудит событий безопасности FreeBSD	120
2.5. Изучение работы инструмента hping3, организация стресс-теста сети с его помощью. Настройка брандмауэра с помощью ufw и iptables	131
2.6. Особенности аудита ОС Linux	149
ЗАКЛЮЧЕНИЕ	169
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	170

ВВЕДЕНИЕ

Аудит информационной безопасности информационных потоков предприятия включает сбор и анализ разнотипной информации, в том числе, сетевой информации. Существует несколько способов организации процессов аудита. Выбор способа зависит не только от доступных ресурсов, но и от специфики информационных систем организаций. Также необходимо внимательно отнестись к процессам аудита, к методам и средствам оценки безопасности инфраструктур промышленных предприятий. Подобные средства позволят ускорить реализацию процедуры испытаний защищенности информационных систем (ИС).

При проведении аудита нужно периодически организовывать сбор и анализ транслируемой и обрабатываемой информации. Современные методики и системы аудита безопасности информационных систем (ИС) и, в том числе, безопасности автоматизированных систем управления технологическими процессами (АСУТП) на предприятиях критической инфраструктуры должны создаваться с учетом требований существующих стандартов и руководящих документов, изданных государством (приказ ФСТЭК России № 31 от 14.03.2014, руководящий документ «Базовая модель угроз безопасности информации в ключевых системах информационной инфраструктуры» (утв. ФСТЭК России 18.05.2007)) и др. Это необходимо для успешной интеграции средств безопасности с учетом специфики информационных систем.

В связи с появлением новых угроз и уязвимостей, приводящих к потере информации, ее компрометации, с учетом особенностей информационных систем, в том числе относящихся к системам критической инфраструктуры, для достижения целей аудита (выявление уровня защищенности информационных блоков и подсистем АСУТП предприятия) превентивного (определение угроз и уязвимостей) и детектирующего характера требуется решить ряд задач, среди которых особо следует отметить задачи, вызывающие затруднения, а именно:

- провести исследования сетевой инфраструктуры;
- произвести поиск уязвимостей различными методами;
- произвести анализ рисков, оценить уровень защищенности системы в состоянии «as is», и при необходимости определить наличие соответствия существующим стандартам ИБ и выработать ряд рекомендаций по повышению уровня защищенности [1; 2];
- рассмотреть их комплексно с возможным использованием тестирования на проникновение.

Особенностям решения указанных задач посвящено это пособие.

1. ОСОБЕННОСТИ АУДИТА СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

1.1. Аудит информационной безопасности сетевой инфраструктуры

Почти во всех компаниях аудит проводится по одному сценарию, состоящему из 7 этапов:

1) определение тех задач, которые нужно решить в рамках нашего проекта. Собирается вся информация, проводится оценка организации действий по подготовке к началу аудита;

2) согласование условий, на которых будет произведен аудит, а также установка временных границ, за какой срок должен быть проведён аудит;

3) обработка полученных результатов;

4) проводится тестирование на проникновение;

5) анализ всех рисков, процедур оценки уровня защиты предприятия и системы, которую выстроили;

6) анализ соответствия стандартам по нормативным документам;

7) разработка рекомендаций в ходе проведения экспертного аудита.

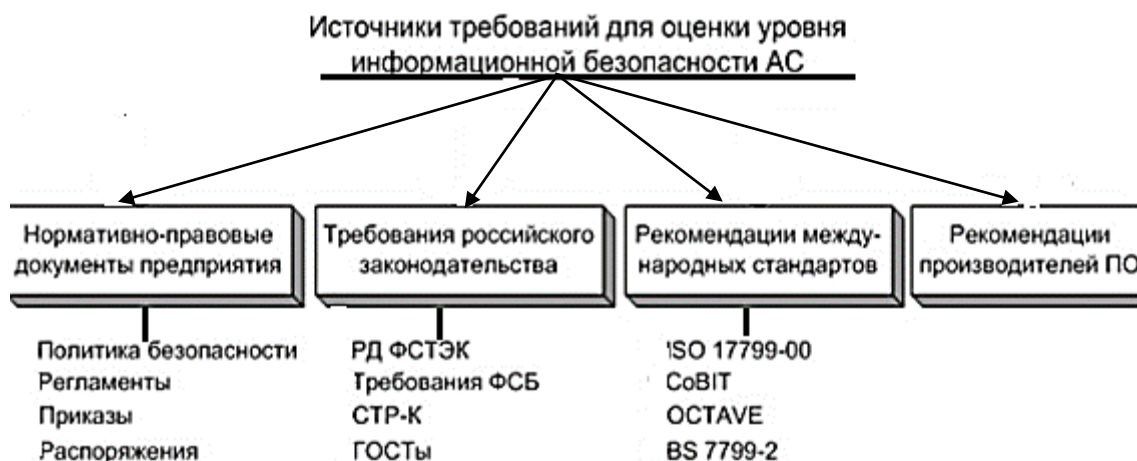


Рис. 1. Источники требований информационной безопасности

Аудит делится на внешний и внутренний. Внутренний аудит – это процесс, регламентированный нормами и правилами установленными самой компанией.

Внешний аудит – это аудит, проводимый независимой аудиторской организацией на основании договора заключенного между компанией и аудиторской организацией.

Так же используется другая классификация, которая включает следующее:

1. **Активный аудит** – это доскональное рассмотрение состояния защищенности подсистемы информационной безопасности.

Активный аудит включает в себя:

- настройку элементов системы, анализ архитектуры;
- прохождение опросов лиц, работающих в системе;
- осуществление проверок, включающих в себя подсистемы информационной безопасности.

При более детальном рассмотрении активного аудита, можно сказать, что первый раз он упоминался в начале 80-х годов. В это время вся регистрационная информация обрабатывалась вручную, без применения какого-либо специального программного обеспечения, людьми, называемыми *штат аудиторов*.

Работа людей естественно была неэффективной и по сути можно назвать её для нашего времени полностью бесполезной, но в отличие от того времени сейчас активный аудит является важным видом аудита ИБ. У активного аудита существует две архитектуры, одна из них – *локальная*, вторая – *глобальная*.

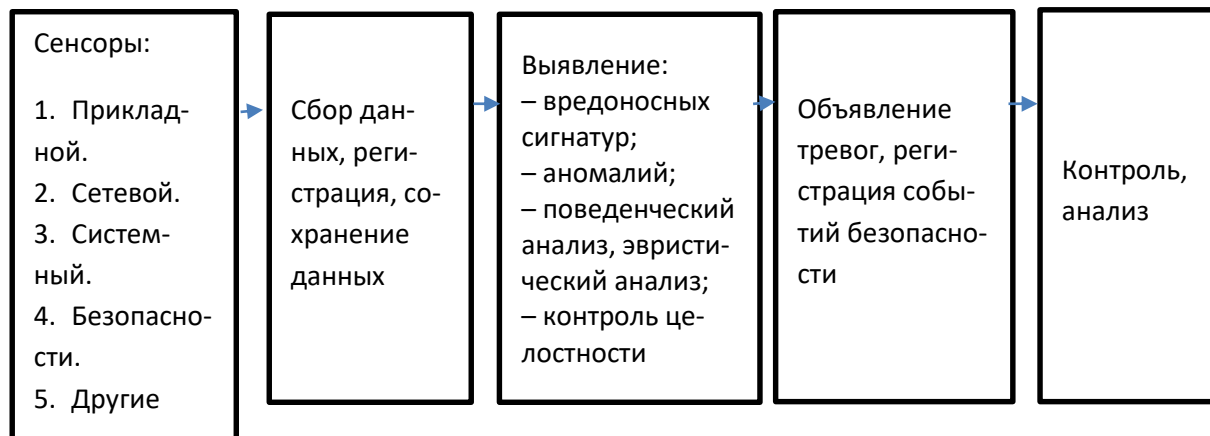


Рис. 2. Локальная архитектура аудита

Основной особенностью локального аудита является то, что основные элементы локальной архитектуры связаны между собой [2].

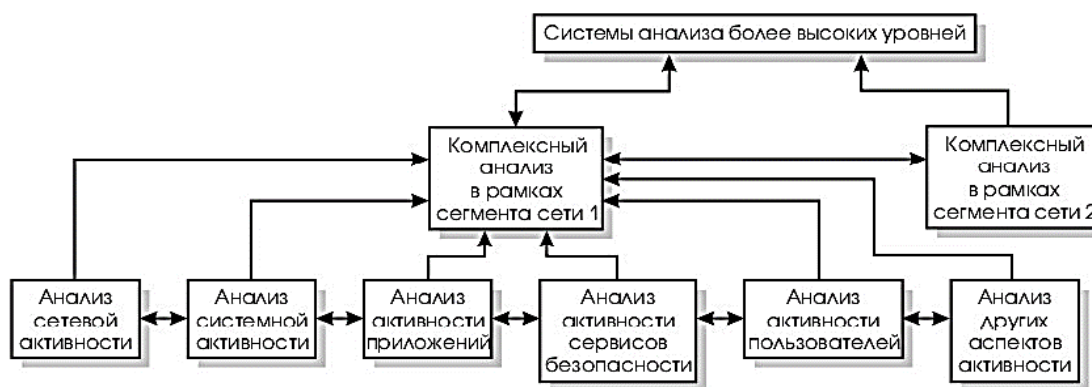


Рис. 3. Глобальная архитектура аудита

Особенностью глобальной архитектуры является то, что устанавливаются связи: *одноранговые* и *двухранговые* между локальными системами.

2. Экспертный аудит – это оценка состояния информационной безопасности системы на сегодняшний день, по уровням нормативно-методологическим, организационно-управленческим и процедурным.

Такой аудит выполняется специалистами области системного управления.

При рассмотрении данного аудита можно сказать, что он производится в несколько этапов:

- 1) определение тех задач, которые нужно решить в рамках нашего проекта. Собирается вся информация, проводится оценка организации действий по подготовке к началу аудита;
- 2) согласование условий, на которых будет произведен аудит, а также установка временных границ, за какой срок должен быть проведён аудит;
- 3) обработка полученных результатов;
- 4) проводится тестирование на проникновение;
- 5) анализ всех рисков, процедур оценки уровня защиты предприятия и системы, которую выстроили;
- 6) анализ соответствия стандартам по нормативным документам;
- 7) разработка рекомендаций в ходе проведения экспертного аудита.

3. Аудит на соответствие стандартам ИБ. В ряде случаев проводится аудит на соответствие стандартам ИБ. Он подразумевает проведение экспертного аудита. Также могут использоваться элементы активного аудита, и в результате получают следующую информацию:

- степень соответствия проверяемой ИС выбранным стандартам;
- количество и категории полученных несоответствий и замечаний.

1.2. Особенности аудита информационной безопасности удаленных автоматизированных рабочих мест

Отдельным аспектом аудита сейчас является тестирование удаленных рабочих мест. Для достижения целей технического инструментального аудита удаленных АРМ превентивного (определение угроз и уязвимостей) и детектирующего характера (уровня защищённости удаленных рабочих узлов, находящихся в зоне подконтрольной политики безопасности и вне данной зоны), требуется решить следующие задачи:

- определить границы контролируемых зон;
- меру контролируемости удаленных АРМ (при этом требуется дифференцировать компоненты зон по уровню доступности);
- скорректировать задачи технического аудита с учетом меры подконтрольности АРМ;
- определить методику проведения пассивного и активного технического аудита и произвести поиск уязвимостей различными методами.
- произвести анализ рисков.

Так же необходимо разделить исследуемую инфраструктуру на компоненты:

- корпоративная информационная система;
- удалённый АРМ с клиентом VPN;
- удалённый АРМ с применением сервисов HTTP, HTTPS, VNC и других [3; 4].

Первая проблема, возникающая при аудите, это определение компонентного состава тестируемых объектов, глубины и последовательности проверок и т. п., т. е. модели аудита, которая должна учитывать всю специфику как объекта исследования, так и условий проведения аудита, возможностей, доступности функционала средств аудита:

- особенности доменной архитектуры;
- тип и версии операционных систем;
- специфика ЛВС, к которой подключен удаленный АРМ;
- наличие стороннего трафика, наличие посторонних и промежуточных узлов [4; 5].

Для того, чтобы разработать модель технического аудита, требуется предварительно создать формальные описания объектов разных сегментов КИС и затем отдельно определить процессы технического аудита для каждого сегмента инфраструктуры предприятия [3].

Необходимо отметить, что в данном случае фактически формируется две общие модели технического аудита (при описании, в том числе, актуальных угроз):

1) модель аудита, в которой учитывается доступ к компонентам КИС в границах локальной инфраструктуры предприятия (стандартная модель);

2) модель аудита КИС с сегментами, включающими удаленные АРМ. Аудит предполагает исследование удаленных сетевых узлов без локального доступа. Данная модель состоит из стандартной модели и дополнительных уникальных моделей, которые формируются в зависимости от специфики технологических платформ, удаленных АРМ, способов связи. Подобные уникальные модели могут содержать операции двух типов:

– исследование с использованием инфраструктуры защищенного трафика;

– исследование без использования в инфраструктуре защищенного трафика.

Общие для всех моделей работы предполагают два этапа:

– *заочное*, аудит производится без непосредственного доступа к инфраструктурным компонентам локальной инфраструктуры предприятия, по сведениям итогов обследования (частичного или полного), при этом специалисты предприятия высылают аудиторов;

– *очное*, при котором обследование всех сегментов будет ограничено технологической базой сегмента.

Заочный аудит полностью применим как к компонентам КИС, так и к удаленной АРМ, но достоверность сведений по аудиту удаленных узлов ниже результатов обследования КИС.

Модель технического аудита удаленных АРМ имеет две разновидности в соответствии с типом исследования:

– модель аудита в инфраструктуре защищенного трафика;

– модель аудита вне инфраструктуры защищенного трафика.

Подобное разделение приводит к необходимости подбирать особый инструментарий исследования, учитывающий инфраструктурные особенности анализируемого объекта. Таким образом, во второй модели следует особо учитывать специфику доступа к удаленному АРМ, так как доступ к нему может осуществляться через промежуточные узлы (роутеры). Кроме того, модель технического аудита, удаленного автоматизированного рабочего места, должна отражать ту специфику удаленной сети вне ДМЗ, к которой подключён АРМ, а также учи-

тивать стабильность этой сетевой инфраструктуры, поскольку она неподконтрольна политике безопасности ДМЗ. В этом случае активный аудит, если он конечно возможен, будет весьма избирателен.

Соответственно, необходимо разграничить сведения, однозначно интерпретируемые при сборе данных о состоянии безопасности удаленной АРМ, и те сведения, которые либо неоднозначно интерпретируются, либо могут быть изменены в короткий период времени, либо на них могут повлиять сторонние факторы. При этом, очевидно, нужно проверять сведения, предоставляемые пользователями об удалённых АРМ и о тех инфраструктурах, в которых они используются.

Учитывая своеобразие обследования всех разновидностей сетевых узлов, при реализации модели уникального аудита можно выделить два основных этапа:

- сетевой (первый) этап;
- сетевой с внутренним локальным доступом (второй) этап.

При реализации первого этапа, при техническом обследовании удаленного узла сотрудника, необходимо проверить доступность АРМ с помощью сетевых средств. При формировании порядка обследования нужно учитывать наличие или отсутствие защищённого соединения, либо канала VPN. В последнем случае при проверке могут использоваться те же самые средства, которые применяются и для узлов в демилитаризованной зоне предприятия. В противном случае модель предполагает исследование с помощью средств сканирования и сетевой инвентаризации и разведки (nmap, сканер SSS, nc, IVRE и др.). Последовательность применения механизмов сбора данных приведенных в таблице №... следующая:

- a) сетевые сканеры (nmap, nc);
- b) определение специфики доступа (nmap, IVNE);
- c) определение уровня защищенности с помощью тестирующих программ SSS, Nessus, nmap (в режиме --scan (vulscan));
- d) определение сервисов сетевого доступа для подключения к службам КИС.

На этом этапе проверяются уязвимости сетевых интерфейсов удаленного узла, возможности подключения к ним сторонних пользователей.

При реализации второго этапа требуется установка системы удаленного доступа на основе протоколов VNC, либо использование RDP-службы, ее аналогов. Используя их, можно получить доступ к исполняемой среде ОС АРМ и провести локальное сканирование с помощью

программ, подобных ScanOval. Таким образом, можно получить достоверную информацию по эксплуатируемому программному обеспечению пользователя и инфраструктурному окружению узла сети. Однако, поскольку рабочая среда пользователя может постоянно меняться в силу отсутствия или нестабильности политики безопасности ДМЗ, при сканировании следует сосредоточиться на безопасности использования сетевых сервисов, необходимых для взаимодействия с сервисами корпоративной информационной системы предприятия.

В результате проведенного анализа были выявлены различия в собираемых при инструментальном обследовании сведениях о состоянии безопасности сетевых узлов. При этом учтен фактор гарантии достоверности сведений (приведены в табл. 1). Последовательность приведенных сведений указывает на требуемый порядок сбора данных при обследовании сетевых узлов.

Таблица 1

Порядок сбора данных при обследовании сетевых узлов

№	Данные о АРМ удаленного типа для инвентаризации инфраструктуры и выявления уязвимостей	Общие КИС	Удаленные АРМ	
			в КИС	вне КИС
1	Общая информация об организации	гарантир.	гарантир.	гарантир.
2	Взаимосвязи между компонентами КИС	гарантир.	гарантир.	гарантир.
3	Программно-аппаратное обеспечение	гарантир.	+	гарантир.
4	Параметры операционной системы	гарантир.	гарантир.	гарантир.
5	Серверные задачи	гарантир.	гарантир.	–
6	Сетевые параметры КИС	гарантир.	гарантир.	–
7	Документация программно-аппаратному обеспечению	гарантир.	–	–
8	Распределение серверов, рабочих станций по сегментам сети, наличие на них критичной информации	гарантир.	гарантир.	–
9	Используемые Internet-сервисы и ресурсы	гарантир.	гарантир.	–
10	Системы управления сетевым оборудованием, системное сетевое программное обеспечение, в том числе наименование, полная версия, полная версия «заплат» (patch)	гарантир.	–	–

№	Данные о АРМ удаленного типа для инвентаризации инфраструктуры и выявления уязвимостей	Общие КИС	Удаленные АРМ	
			в КИС	вне КИС
11	Типы сетевого оборудования, версии прошивок/операционных систем	гарантир.	—	—
12	Документация производителей на сетевое оборудование, собственная технологическая документация	гарантир.	—	—
13	Использование встроенных криптографических средств защиты информации трафика	гарантир.	гарантир.	—
14	Прикладное ПО с ассоциацией к подконтрольным узлам	гарантир.	гарантир.	—
15	Учетные записи, политики доступа, средства предотвращения НСД	гарантир.	гарантир.	—
16	Специальные возможности прикладного программного обеспечения	гарантир.	—	—
17	Наличие критичных для предприятия процессов электронной обработки и передачи данных	гарантир.	гарантир.	—
18	Имеющиеся средства архивирования, режим их работы, места хранения архивов	гарантир.	гарантир.	—
19	Системы протоколирования	гарантир.	гарантир.	—
20	Межсетевые экраны	гарантир.	гарантир.	—
21	Системы мониторинга безопасности	гарантир.	гарантир.	—
22	Системы сканирования безопасности сети и передаваемой информации	гарантир.	гарантир.	—
23	Криптографические средства	гарантир.	гарантир.	—

Из приведенных данных видно, что количество достоверных сведений удаленных узлов, как и мера их достоверности, вне ДМЗ, гораздо меньше, чем внутри контролируемой зоны.

Для повышения меры достоверности требуется перед реализацией второго этапа полностью перехватить управление домашним компьютером пользователя, который выступает в качестве удаленного АРМ. Очевидно, сотрудник, в силу различных обстоятельств (наличие личной информации, персональных данных), имеет право отказать в перехвате управления узлом. Подобные операции требуется согласовывать перед аудитом. В противном случае процедуры активного исследования удаленного АРМ ограничатся процедурами первого этапа.

При проведении обследования сетевой инфраструктуры с включенными в общую систему удаленными АРМ следует учитывать сегментированный состав инфраструктуры. Классическая модель системы при проведении инструментального аудита с учетом моделирования угроз предполагает связь элементов основных множеств инфраструктурных компонентов (по расширенной модели Клементса-Хоффмана)[2]. Таким образом, процесс сбора информации должен учитывать барьеры защиты, разделяющие зоны контроля: КИС, Интернет-сеть, домашняя сеть сотрудника с удаленным АРМ. Соответственно, процесс активного аудита как эмулирование атакующих процессов, реализуется в системе, описываемой пятимерным кортежем $S = \{O, U, M, V, B\}$ (где O – множество элементов защиты, U – множество вероятных угроз, M – множество механизмов защиты, V – множество уязвимых мест, представляющих собой пути проникновения в систему, B – множество барьеров защиты)[2].

Внесение в систему удаленных АРМ с комплексом инфраструктурных элементов, расположенных вне контекста ДМЗ, усложняет описание анализируемой системы. Фактически появляется смежная система, также описываемая пятимерным кортежем, взаимодействующая с основной, но не объединенная механизмами защиты (набор барьеров, определяемый декартовым произведением $V * M: b_q = \langle u_i, o_j, m_k \rangle$). В результате объединения объектов различных сегментов при сетевом взаимодействии в единое множество возникает объединение подмножеств уязвимостей сегментов (набор уязвимых мест, определяется как подмножество декартова произведения $U * O: v_p = \langle u_i, o_j \rangle$) [2]. Также уязвимости одних сегментов КИС при сетевом взаимодействии могут компрометировать объекты других сегментов, где эти уязвимости не наблюдаются. Соответственно, возникает ситуация наследования уязвимостей, которая проявляется в том, что уязвимость на стороне КИС может привести к компрометации домашней сети сотрудника, а также его удаленного АРМ, и наоборот, уязвимости АРМ могут скомпрометировать сегмент КИС предприятия, т. е. задачи выявления уязвимостей усложняются, поскольку появляются отдельные уязвимости и составные. При этом

связь механизмов защиты возможна лишь в том случае, если удаленный узел АРМ включен в ДМЗ. Мера включенности определяет уровень защищенности общей инфраструктуры и понижает степень вероятной ее компрометации. В итоге, создаваемая модель аудита должна учитывать наличие наборов барьеров каждой стороны информационного взаимодействия, степень связанности систем защиты. При этом не связанность компонентов безопасности различных инфраструктурных сегментов может оказаться препятствием для осуществления съема информации в процессе проведения.

1.3. Особенности аудита информационной безопасности с применением актуальной методики оценки угроз

Современные системы обработки конфиденциальной информации, персональных данных (ПДн) предполагают обязательное использование средств защиты информации. Они обеспечивают надежность и безопасность функционирования информационных систем различного типа (ИС). Для определения надёжности систем безопасности, а также для оценки уровня защищенности автоматизированных информационных систем (АИС) применяются различные методики (авторские, корпоративные, государственные). Для оценки угроз в Российской Федерации используются несколько руководящих документов, последний из которых принят в 2021 году: «Методика оценки угроз безопасности информации» [2]. Предложенная в нём методика предлагает использовать при анализе угроз массивы данных об уязвимостях, а затем, ориентируясь на экспертное мнение, формировать векторы атак с возможностью заимствования различных, в том числе зарубежных, способов описания сценариев реализации угрозы (CAPEC, ATT&CK, OWASP, STIX, WASC и др.). Кроме того, документ предполагает порядок определения источников угроз, учет возможностей потенциальных злоумышленников (уровень их квалификации и подготовленности, т. е. наличие инструментария для эксплуатации уязвимости), а также предложено рассмотреть аспекты целеполагания нарушителей. При этом оценка инструментария доступного злоумышленникам и оценка возможности применения этого инструментария по отношению к обнаруженным в автоматизированной информационной системе уязвимостям сопряжена со спецификой атакуемой сетевой инфраструктуры. Таким образом, данная методика, применяемая в процедурах аудита, в основном ориентирована на рассмотрение в качестве целевого объекта распределенных информационных систем (они не предполагают

применение сетевых технологий для поддержки работоспособности системных и пользовательских компонентов), локальных уязвимостей. В методике подробно не оговаривается, каким образом осуществляется поиск уязвимостей (описаны в стандарте ГОСТ Р 56546-2015), влияющих на выбор актуальных тактик.

Применяется выявление уязвимостей инфраструктурных компонентов по списку уязвимостей на сайте регулятора (ФСТЭК). Могут применяться различные техники активного аудита (тестирование). При этом надо учитывать то, что не все испытания в информационных системах различного типа возможны. Допустим, в автоматизированных системах управления технологическими процессами (АСУТП) связанные между собой подсистемы могут быть повреждены из-за применения различных технологий тестирования, которые препятствуют режиму обмена данными real-time между диспетчерским уровнем и уровнем ПЛК [5]. Очевидно, что типы информационных систем и специфика поиска уязвимостей в этих информационных системах должны быть учтены при построении векторов атак по требованиям рассматриваемого методического документа.

При этом в самой методике, хотя и используются понятия риск, угроза, но при этом не учитывается:

- во-первых, стойкость барьеров, блокирующих реализацию угрозы злоумышленником;

- во-вторых, сам предложенный способ описания вектора атаки не учитывает вероятностные показатели путей достижения цели злоумышленником (вероятность достижения цели злоумышленником одним из путей, предполагает наличие способа эксплуатации актуальных уязвимостей и самой искомой уязвимости).

При этом в методике утверждается, что величина риска, как и другие основные показатели (оценка возможностей злоумышленника, оценка сценариев реализации угроз и т. п.) зависит от экспертного мнения (оценки) нескольких экспертов. Поскольку в методике отсутствует понятие остаточного риска у оцениваемых барьеров (средств информационной безопасности). Эксперты вынуждены либо признать полную гарантированность функций безопасности средств защиты информации, либо предложить собственные методы вычислений, позволяющих оценить стойкость барьеров защиты информации. Величина стойкости барьера напрямую влияет на оценку защищенности систем обработки информации и на эффективность системы защиты, как комплекса различных компонентов.

Кроме того, появление остаточного риска предполагает трансформацию вектора атаки, т. е. его продолжение, но в изменённом виде. Построенный вектор атаки, предполагающий гарантированное исполнение функций безопасности средством защиты по отношению к угрозе информации, и вектор атаки, который предполагает учет недостатка уровня защищённости средств безопасности, будут резко отличаться по длине (порядку действий) [6]. Последний тип вектора учитывает возможность возникновения нового раунда реализации атаки или возникновения нового сценария реализации угрозы. Таким образом, вектор атаки, построенный с учетом остаточного риска механизмов защиты, будет состоять из множества различных векторов, т. е. фактически будет являться их сверткой. Подобный способ формирования вектора выгоден при реализации аудита многоуровневых систем, поскольку будут учитываться все особенности используемых при обработке информации технологий, особенности средств защиты. Продолжение вектора или провоцирование появления нового будет зависеть от меры успешности предыдущих атакующих последовательностей и наличия требуемых уязвимостей.

Соответственно, необходимо классифицировать уязвимости по уровню применяемых технологий в автоматизированных информационных системах. Допустим в подсистемах АСУТП существуют [7]:

- уязвимости MES-систем;
- уязвимости Scada-систем и OPC-серверов (для их выявления требуется использование специализированных банков уязвимостей к ICS-CERT, NVD/CVE, SCADA Strangelove, SiemensProduct CERT).

Очевидно, что квалификация экспертов должна соответствовать требованиям рассматриваемой технологической области, должна позволить определить специфику той системы, которая исследуется при аудите. Это означает, что в обсуждении должны предположительно участвовать не только сами эксперты в области информационной безопасности, но и эксперты в области проектирования, формирования и разработки рассматриваемых информационных систем. Это касается, прежде всего, систем критической инфраструктуры, в том числе, автоматизированных систем управления технологическими процессами. Подобные системы формируются из различных компонентов технологически отличных уровней. Соответственно, в этом случае в процедурах аудита дополнительно требуется участие:

- экспертов, которые осведомлены об уязвимостях, как и самих ПЛК, так и средств съёма данных датчиков;
- программистов, которые обеспечивают разработку систем контроля и агрегирования данных;

– программистов, которые участвуют в создании систем административного уровня.

Таким образом, количество экспертов, которые должны участвовать в обсуждении проблем безопасности и оценивании способов реализации различных угроз увеличивается. При сегментировании подсистем оцениваемых АИС и, соответственно, при сегментировании процедур поиска угроз и уязвимостей может проявиться проблема несогласованности оценок различных групп экспертов занимающихся своими подсистемами и компонентами технологически различающихся уровней.

В помощь специалистам по информационной безопасности, оценивающим угрозы ИС, т. е. фактически проводящим аудит (плановый и внеплановый) с учетом требований актуальных методических руководств ФСТЭК, можно предложить ряд дополнительных решений в области оценки риска, позволяющих более точно проанализировать угрозы безопасности и выявить специфику реализации атакующих последовательностей.

Предварительно требуется рассмотреть все аспекты действий, связанных с особенностями проведения аудита, результаты которого позволяют выявить уязвимости и определить их взаимосвязи с актуальными угрозами. В некоторых системах (АСУТП и т. п.) сбор информации об уязвимостях программного обеспечения, программно-аппаратных платформ осложнен спецификой используемых в платформах технологий. Заочный аудит в этом случае не позволит определить полностью технологические спецификации исследуемых программно-аппаратных платформ без непосредственно рассмотрения таковых, и поэтому будет неточен. Соответственно, применять заочный аудит можно ограниченно, с учетом недостоверности или неполноты инвентаризационных данных.

Применение способов активного обследования, в том числе пентеста, могут негативно повлиять на работоспособность исследуемых систем и, следовательно, могут нанести непоправимый ущерб компонентам систем подобных АСУТП. Поэтому предварительно требуется разработать модель аудита, в том числе проведения процедур поиска уязвимостей. При создании модели нужно учитывать то, что зачастую невозможно корректно экстраполировать полученные в одном сегменте результаты на другие сегменты (причинами этого могут быть технологические различия используемых компонентов систем как на аппаратном уровне, так и на программно-аппаратном).

В частности, правила и критерии оценки работоспособности систем поддержки технологических процессов могут зависеть от наличия распро-

странённых технологий обмена данными (присутствует ли или отсутствует поддержка протоколов стека TCP/IP и на каком уровне). Использование технологий промышленных протоколов передачи данных (HART, CAN и т. п.) может вызвать проблемы в понимании того, что является фактом компрометации. Во-первых, критерии компрометации трафика и компонентов систем следует описывать с учетом отличительных особенностей технологических платформ, что может вызвать затруднение из-за недоступности актуальных данных об уязвимостях. Во-вторых, требуется понять, что может в рассматриваемой платформе считаться компрометацией, нарушением политики информационной безопасности. Наиболее показательным признаком нарушения безопасности является наличие факта нарушения режима трансляции данных real-time на диспетчерском уровне и уровне ПЛК. Подобные факты указывают на нелегитимные изменения в маршрутизации трафика, наличие непредусмотренного настройками трафика, в том числе, негативно влияющего на функциональность программно-аппаратных компонентов (паразитный трафик может указывать на внешнее воздействие и/или на наличие в системе компрометирующих программных компонентов, т. е. программных закладок или вредоносных объектов). В итоге, модель аудита может включать в свой состав процедуры, ориентированные на различные подходы к анализу риска, в том числе относящиеся к различным стандартам ИБ.

Однако выбор стандартов предполагает четкое следование предъявляемым в них требованиям, что ограничивает применимость аналитических моделей. В отличие от сопоставительного анализа, направленного на проверку соблюдения требований безопасности информации в тестируемой системе, пен-тест позволяет определить уровень защищённости систем. При этом применяемые задания в тестах должны предусматривать все варианты поведения технологической инфраструктуры для того, чтобы сделанные на основе проведенного тестирования выводы о степени защищенности исследуемой системы были достоверны [6]. В этом случае задания активного аудита можно разделить в соответствии с делением системы на сегменты или технологические уровни с учетом функций компонентов уровней/сегментов [6]. Допустим, в модели аудита сегментов и уровней АСУТП функции, работоспособность которых тестируется, могут быть следующими:

- регистрация событий компонентов сегмента, классифицирование данных о событии в процессе контроля;
- архивация регистрируемых данных процессов ПЛК;
- сигнализация о событиях разного класса;

- регистрация событий безопасности смены политик безопасности;
- регистрация событий в АСУТП, SCADA-системах OPC-серверов;
- передача трафика с использованием протоколов HTTP, HTTPS, SSH;
- применение XML- файлов для управления передачей данных и др. [7].

Таким образом, будет учтена уровневая специфика функций, закреплённых за различными сегментами технологической сети. При этом можно отслеживать и специфику межуровневого и межсегментного взаимодействия, используя интеграционное тестирование [8].

Для определения специфики сценария атакующих последовательностей требуется определить остаточный риск, возникающий после воздействия угрозы на уязвимый объект системы (значение, показывающее вероятность отражения атаки средством защиты информации). Оценить качество защиты можно различными способами. Наиболее простые основаны на использовании методов теории надёжности. Отказ системы защиты трактуется как результат успешного воздействия угрозы на систему ИБ в любой степени. Тогда интенсивность отказов системы ИБ равна сумме интенсивностей отказов каждого средства защиты, т. е. не полностью выполнившего свою функцию (не гарантировано), что фиксируются в журналах событий безопасности рассматриваемой системы [6]. В данном случае важен факт неполного отражения атаки – он указывает на компрометацию защиты информации. Сумма интенсивностей отказов рассчитывается следующим образом [10]:

$$Y = \sum_{i=1}^n y_i, \quad (1)$$

где Y – интенсивность отказов комплекса средства защиты;

y_i – интенсивность отказа отдельного средства защиты (средства ИБ можно объединять в группы, избегая излишней детализации при необходимости).

Соответственно, показателем успешности средств безопасности будет являться вероятность исправной работы средств безопасности $P(t)$ в течение определяемого при аудите заданного времени t . С учётом величины интенсивности отказов по отдельному средству безопасности вероятность гарантированного отражения угрозы определяется следующим образом [10]:

$$P(t) = e^{-Yt}. \quad (2)$$

Таким образом, учитывая особенности разных средств защиты информации, допустимо вычислить для каждого средства вероятность его исправной работы ($p_i(t)$), а затем и величину риска по отношению к каждому объекту защиты. Также можно вычислить вероятность исправной работы системы защиты как комплекса. Однако, исходя из эмпирического опыта использования единого комплекса средств защиты информации, можно отметить, что достаточного одной успешной реализации атакующей последовательности, чтобы признать систему защиты скомпрометированной полностью, т. е. компрометация одного средства в этом случае, означает компрометации всех средств, сопряженных с ним.

Продолжая рассмотрение вспомогательных методов, можно привести способы вычисления временного периода успешной работы средств защиты между двумя ближайшими по времени успешно реализованными атаками. Данный параметр называется «наработка на отказ системы защиты информации» (U). Параметр фактически указывает время гарантированно успешного функционирования системы защиты информации и, в целом, работоспособности защищаемой и защищающей системы. Данный параметр рассчитывается следующим образом [10]:

$$U = \frac{1}{Y}. \quad (3)$$

Дополнительно, можно отметить, что время восстановления U_r будет оцениваться как время, которое потребуется для восстановления функциональности системы ЗИ. Эта величина будет указывать также на время поиска и нивелирования канала НСД (несанкционированного доступа).

Поскольку при рассмотрении защищающей и защищаемой системы выделяются множества различных параллельно и последовательно связанных объектов, будет логично представить сегментированные системы обработки информации в виде множеств компонентов, классифицируемых по специфике их функционального предназначения. Логическую организацию исследуемых систем можно привести в виде набора элементов нескольких множеств (элементы имеют различные взаимосвязи), которые и следует учитывать при разработке модели аудита. Выделяются следующие типы множеств [3]:

- множество объектов защиты;
- множество потенциальных угроз;
- множество средств обеспечения информационной безопасности;

- множество уязвимостей;
- множество «барьеров» защиты информации, которые функционально и топологически определяют локализацию и специфику перехвата угрозы, направленной на уязвимость (реализация функции защиты с учетом специфики защищаемого объекта). При этом наборы тактик злоумышленника будут включать в свой состав каждую ветку направленного графа атаки, вершинами которого будут являться элементы описанных множеств.

Если рассматривать тактики более абстрактно, без специфики организации защищаемой системы, в кортеж тактик злоумышленника будут включены элементы следующих множеств:

- множество потенциальных угроз (наборов тактик);
- множество уязвимостей.

Успешность работы средств безопасности определяется, в данном случае, величиной сохранения риска безопасности, т. е. величиной остаточного риска (успешность тактики злоумышленника), которая связана с осуществлением отдельной атакующей последовательности (угрозы) при преодолении средства обеспечения безопасности информации по отношению к конкретному объекту, указанному в кортеже элементов множеств, описывающих систему. Значение успешности связанных веткой тактик (остаточного риска) определяется следующим образом:

$$R_{\text{ост.}i} = P_i C_i (1 - P_{\text{сопр.}i}), \quad (4)$$

где P_i – вероятность появления набора тактик (угрозы) y_i (можно рассматривать вероятности отдельных тактик, но в этом случае будет затруднительно определять величину ущерба, поскольку не каждая тактика предполагает схожие по способу описания потери); C_i – величина потерь при успешной реализации вектора атаки (набора тактик) y_i в отношении защищаемого объекта o_j ; $P_{\text{сопр.}i}$ – степень сопротивляемости барьера b_q , характеризующаяся вероятностью его преодоления.

Уровень защищенности системы можно определить следующим образом [3]:

$$U = \frac{1}{Y \sum_{(\forall b_{\text{сопр.}i} \in B)} P_i C_i (1 - P_{\text{сопр.}i})}, \quad (5)$$

где U – уровень защищенности системы; B – множество средств блокирования угроз $b_{\text{сопр.}i}$ с учетом наличия уязвимостей.

Для уточнения специфики реализации сценариев атаки (атакующей последовательности), а точнее, для определения наиболее выгодных для злоумышленника тактик по отношению к элементам кортежа, которые обозначают целевые объекты, можно использовать методику построения графа компрометации. Данная методика подразумевает формирование трех альтернативных путей реализации сценария атаки [6]. Реализация каждого пути зависит от вероятности успешной реализации тактики по отношению к уязвимостям системы. При этом учитывается время реализации злоумышленниками своих атакующих действий. Наименьшее время, которое злоумышленник затратит, является наиболее выгодным для атакующей стороны и, соответственно, указывает на наиболее успешный сценарий атаки и выбранную для его реализации наиболее успешную тактику. Предполагается, что в таком графе компрометации будет учитываться состояние системы, которое достигает злоумышленник при использовании той или иной тактики. Однако в отличие от рекомендаций оригинальной методики графа компрометации, где рассматривается пять основных состояний системы (пять этапов атаки), в графе, который формируется с учетом требований актуальной методики оценки угроз ФСТЭК, предполагается десять достигаемых состояний (по числу тактик) [6].

Определение наиболее эффективного пути и, соответственно, последовательности выбранных злоумышленниками наиболее выгодных тактик (сценария атаки) не вызывает затруднений. Выгодность последовательности тактик зависит от выделяемого для ее реализации времени: чем меньше время затрачивается на реализацию атаки, тем выше вероятность успешного достижения злоумышленником своих целей.

Методика предполагает рассмотрение трех вариантов реализации тактик (сценария атаки) в зависимости от доступности средств эксплуатации уязвимостей, известности уязвимости:

- первый вариант предполагает ситуацию, когда найдено и доступно для эксплуатации не менее одной уязвимости, и при этом злоумышленник знает тактику и имеет средства для ее реализации (исключает второй вариант);

- второй вариант предполагает ситуацию, когда найдено и доступно для эксплуатации не менее одной уязвимости, и при этом злоумышленник знает тактику, но не имеет средства для ее реализации (исключает первый вариант);

- третий вариант предполагает ситуацию, когда найдено и доступно для эксплуатации не менее одной уязвимости, и при этом злоумышленник

находится в поиске наиболее актуальной тактики (соответственно, находится в поиске новых уязвимостей, средств их реализации).

Каждый из вариантов сценариев атаки характеризуется своим распределением вероятностей и временем их осуществления с учетом величин сопротивляемости барьеров защиты.

Таким образом, описание последовательности тактик с учётом их выгоды, а также с учётом стойкости барьеров можно повысить степень точности оценки угроз безопасности и, в том числе, помочь экспертам, занимающимся созданием моделей угроз.

1.4. Сравнительный анализ технологий аудита информационной безопасности сетевой инфраструктуры диспетчерского уровня АСУТП

Для проведения аудита информационной безопасности сетевой инфраструктуры требуется осуществить сбор и анализ большого массива данных за определенный период времени. Кроме того, требуется периодически осуществлять повторение процедур сбора и анализа сетевых данных для более полного представления о возможных проблемах в сети. Сейчас большое распространение получили системы типа SIEM (Security information and event management), которые позволяют осуществлять процедуры аудита сетевой инфраструктуры в процессе ее эксплуатации непрерывно. Целью данного типа аудита является определение того, насколько сетевая инфраструктура соответствует предъявляемым к ней требованиям информационной безопасности (ИБ). Таким образом, определяется так же уровень защищённости информационных сетевых компонентов корпоративной информационной системы (КИС) предприятия.

Для достижения целей аудита (превентивного и детектирующего типа) информационной безопасности сетевой инфраструктуры АСУТП требуется решить ряд задач:

- осуществить поиск уязвимостей сетевых систем различными методами;
- оценить риски с учетом результатов исследования активности в сетевой инфраструктуре организации (отслеживание легитимных и вредоносных запросов);
- оценить уровень защищенности с учетом всех данных и всех факторов, влияющих на состояние инфраструктуры;
- определить уровень соответствия стандартам ИБ и выработать ряд рекомендаций по повышению уровня защищенности.

Для решения поставленных задач требуется изучить состояние сети, определить особенности ее инфопотоков и, если позволяет специфика инфраструктуры, провести тестирования на проникновение, т. е. реализовать активный аудит (pen-тест). Однако в сетях АСУТП на диспетчерском уровне pen-тест, как правило, не допускается, поскольку подобные действия могут снизить работоспособность системы, что негативно скажется на работоспособности предприятия в целом.

Поскольку при аудите и последующем анализе его результатов следует учитывать влияние уже встроенных в инфраструктуру средств информационной безопасности, нужно изучить, какие компоненты систем защиты активны на рассматриваемом уровне АСУТП. Защита может обеспечиваться следующим:

- системами распределения прав доступа;
- системами контроля трансляции данных через пользовательский интерфейс ОРС-сервера, сервер базы данных и подсистемы сбора и хранения данных;
- антивирусными системами;
- системами обнаружения вторжений.

Разграничение доступа осуществляется средствами операционной системы. Также могут применяться специальные программные и программно-аппаратные средства для защиты операционной системы, позволяющие контролировать разрешения политик безопасности на критически важных компонентах сети. Следует учитывать, что при аудите цели злоумышленника не очевидны, и часто проявления деструктивных влияний на компоненты уровня могут оказываться непоказательными, поскольку не затрагивают целевые объекты атаки.

Однако большую сложность представляет контроль среды передачи данных на диспетчерском уровне АСУТП. Сейчас Ethernet используется как единая среда передачи данных для АСУТП, хотя на диспетчерском уровне могут транслироваться данные подключенных ПЛК, которые используют протоколы Modbus/TCP, EtherNet/IP, PROFINet и др. При этом рекомендуется применять те устройства передачи данных, которые обеспечивают защиту от подмены IP-адресов узлов сети, а также защиту от утечки данных. Соответственно, требуется определить те методы и технологии, которые могут, учитывая специфичность сетевой инфраструктуры уровня системы, использоваться для реализации полноценного аудита. Необходимо понять, какие при аудите слабые и сильные стороны существуют у техник анализа данных с учетом технологических ограничений.

Одним из важнейших направлений контроля состояния АСУТП является внедрение и «тонкое» (т. е. учетом функциональных особенностей компонентов сети) использование систем обнаружения сетевых атак как в сетевой, так и в хостовой реализации. Известные сетевые системы обнаружения и предотвращения вторжений могут использоваться как средства защиты и, одновременно, как средство сбора данных с последующей сигнализацией об аномальном состоянии системы или данных трафика.

Такие системы, размещенные на диспетчерском (среднем) уровне, используются для защиты нижних уровней АСУТП, на которых работа средств контроля трафика может привести к задержкам в трансляции данных, что критически скажется на работе ПЛК.

Принцип работы средств аудита и контроля защищенности строится на основе централизации процедур журналирования (системные журналы и журналы аудита безопасности) и выявления критичных для системы событий с оповещением о них администраторов безопасности. При этом следует учитывать проблемы, которые могут возникнуть при эксплуатации данных систем. Главная проблема – это несовместимость технологий обработки данных и способов их трансляции. Технологические различия могут проявляться в использовании или неиспользовании OPC-сервера в системе, касаться различий операционных платформ. Проблемы несовместимости могут быть связаны с использованием протоколов, уникальных для технологической сборки уровня ПЛК.

Несовместимость средств аудита и исследуемых систем может возникнуть по следующим причинам:

- влияние на поток данных, что ограничит функционал SCADA, работающей в реальном режиме времени, с последующим вызовом коллизий;
- возможен эффект исчерпания ресурсов при активном аудите сервисов и/или сетевых служб SCADA.

Таким образом, средства аудита и контроля трафика следует встраивать на диспетчерский уровень, во-первых, учитывая технологическую специфику конкретно взятого диспетчерского уровня АСУТП системы, во-вторых, учитывая разделение технологий передачи данных на два уровня: уровень стандартных протоколов, используемых в корпоративных информационных системах и уровень технических протоколов типа CANopen, HART, Modbus. Так же требуется соблюдать осторожность при перехвате данных для анализа в процессе аудита, поскольку это может вызвать критическую для системы нижнего уровня АСУТП задержку.

Для проведения аудита информационной безопасности на диспетчерском уровне необходимо определить основной компонентный состав уровня, функциональные свойства его компонентов, а так же выявить средства информационной защиты, применение которых возможно с учетом технологической специфики уровня. Необходимо учитывать особенности организации АСУТП, а именно: подсистемы диспетчерского и полевого (нижнего) уровня должны быть связаны с корпоративной сетью и компонентами административного управления. Часть системы, при эксплуатации полевого уровня, может работать в реальном режиме времени, который часто совмещается с виртуализированной обработкой данных на уровнях общего управления. Таким образом, исследуя инфраструктуру предприятия при аудите, требуется определить, к какому уровню относятся следующие компоненты:

1. Межсетевые экраны (МЭ), расположенные на границах уровней.
2. Системы удаленного доступа позволяющие контролировать производственные процессы.
3. Модемы, предназначенные для связи между MTU-терминалами и периферийными устройствами.
4. Маршрутизирующие устройства, выполняющие функции соединения сетей LAN с WAN-сетями, MTU-терминалов и RTU-устройств.

Кроме того, следует проанализировать промышленную сеть, которая включает сенсоры, датчики и элементы оборудования с программируемым логическим контроллером (ПЛК).

Учитывая эти факторы, можно выделить те технологии, при работе с которыми проблем совместимости возникнуть не должно. В рамках решения задач аудита применимы системы обнаружения (СОВ) типа OSSEC. Эта система классифицируется как HIDS (система хостового типа), и имеет клиентско-серверную архитектуру. В ней в качестве клиентов применяются агенты, установленные на контролируемые узлы сети. Агенты передают данные на сервер для последующего анализа с использованием специально созданных для заданного уровня инфраструктуры АСУТП правил аудита. Также данная СОВ решает задачи контроля целостности среды, отслеживает доступ привилегированных пользователей. Эта технология позволяет не использовать интеллектуальные методы анализа данных, и объем анализируемой информации не ограничен только сетевыми данными.

Перед этапом внедрения СОВ нужно было определить принципы встраивания механизмов OSSEC HIDS и принципы формирования правил аудита с учетом топологии сети и технологических особенностей

диспетчерского уровня. Эти принципы были сформулированы следующим образом:

1. Функционал OSSEC HIDS не должен влиять на работоспособность АСУТП. Это может быть достигнуто за счет избирательной установки агентов на специально отобранные узлы с учетом специфики сетевых сервисов диспетчерского уровня. При этом COB будет функционировать на уровне предупреждений в режиме Real-Time.

2. Необходимо обеспечить зеркалирование трафика через SPAN-порт (Switched Port Analyzer) в том случае, если есть вероятность возникновения коллизии в трансляции данных при работе OSSEC.

3. Порт 1514/udp должен быть свободен, так как агенты подключаются к серверу через этот порт. Поскольку агент взаимодействует с сервером через UDP-порт 1514, OSSEC может работать на узле и как сервер и как агент. В этом случае он не будет влиять на трафик, но будет исключен из общей сети агентов, что снизит уровень централизации аудита.

Основываясь на указанных принципах, целесообразно формировать компонентную сборку механизмов COB следующим образом: узлы MPB (монитор реального времени) и Web-сервисов будут включать в свой состав и серверы OSSEC, и агенты, поскольку при работе с данными компонентами опасность задержки в передаче сетевых пакетов будет критична для системы. Использование полноценной клиентско-серверной архитектуры COB, т. е. когда сервер будет отделен от агентов сбора данных, возможно, если агенты будут интегрированы на отдельные узлы АРМ и серверы архивов, поскольку в этом случае допустима небольшая задержка при трансляции данных.

Правила должны учитывать специфику компонентной организации механизмов COB, а также требования трансляции служебных данных при работе в инфраструктуре диспетчерского уровня. При этом, у каждого правила OSSEC есть уникальный идентификатор (идентификаторы в диапазоне от 100 000 до 109 999 для собственных правил). Правила сгруппированы в соответствии с целью их применения. Так же у каждого правила есть уровень критичности (level) от 0 до 15. При значении 0 событие игнорируются, а 15 означает максимальный уровень критичности [1]. Это позволяет задать приоритеты при анализе событий. Всего для рассмотрения процедур аудита 11 типов правил:

1. Угроза внедрения вредоносного кода или данных.
2. Угроза перехвата данных, передаваемых по вычислительной сети.
3. Угроза подделки записей журнала регистрации событий.

4. Угроза удаленного запуска вредоносного кода в обход механизмов защиты операционной системы.
5. Угроза воздействия на программы с высокими привилегиями.
6. Угроза сканирования веб-сервисов.
7. Угроза подбора пароля.
8. Угроза «кражи» учётной записи доступа к сетевым сервисам.
9. Угроза обнаружения хостов.
10. Угроза использования механизмов авторизации для повышения привилегий.
11. Угроза внедрения кода или данных.

Таким образом, каждое правило отнесено к определенному типу узлов диспетчерского уровня, учитывает архитектурные особенности построения серверно-агентной системы HIDS и имеет уровень критичности в соответствии с оценкой опасности угрозы. При этом надо отметить, что данные правила могут работать с протоколами технических систем уровня ПЛК, такими как ModBus.

Как дополнение к системам SIEM для объективной оценки рисков сетевой инфраструктуры можно использовать интеллектуальные методы и, соответственно, механизмы анализа. Реализация данных методов имеет свою специфику. Один из наиболее эффективных интеллектуальных методов связан с построением аналитических моделей, которые используются для анализа данных. Применяя их, можно понять закономерности проявления нарушений и в дальнейшем прогнозировать появление потенциальных угроз. Данные модели можно непрерывно обучать. В целом, эта методика предназначена для нахождения комбинации математических уравнений, которые лучше всего предсказывают результат.

Перед обучением модели необходимо сформировать обучающую выборку. Ее можно получить, снимая данные в определенный период времени с подконтрольной сети перед началом процедур аудита, или же заимствовать данные из набора COB OSSEC в той части, где фиксируются атаки. Однако и в первом и во втором случае данные необходимо привести к тому виду, который требуется для обработки в модели. Для этого в модуле анализа обязательно должны присутствовать:

- подмодуль обработки и анализа типа Pandas;
- инструментарий для прогнозного анализа данных типа Scikit-learn (он включает в себя библиотеки классификаторов, которые будут использоваться для анализа) [12].

Анализатор захватывает весь сетевой трафик, но поскольку обучение касается протоколов модели OSI, то будут интерпретироваться

только кадры Ethernet. В выбранной модели заранее задаются исключения на отбор пакетов технологического уровня, что позволит сохранить требуемую скорость передачи для технических данных. После интерпретации кадра, пользователю демонстрируется информация заголовков кадра и происходит сравнение значения поля «EtherType» (Ethernet_protocol). Согласно стандарту, IEEE 802.3 полю IPv4 соответствует значение 0x0800. При несоответствии поля заданным параметрам будет указано: «неопределённый сетевой протокол». Такой пакет не будет обрабатываться.

Следующий необходимый компонент – программа Argus, которая генерирует информацию о состоянии трафика. Она обрабатывает полученные пакеты и генерирует сводные данные о сетевом потоке. Argus может работать на отдельном граничном узле, фильтруя весь сетевой трафик. Кроме того, эта программа может работать как автономный модуль сбора и анализа сетевых данных. Используя включенные в свой состав алгоритмы, Argus находит признаки для формирования шаблона. Каждый признак и его атрибуты, используемые для обучения модели и создания решения, нужно анализировать вручную, чтобы найти ошибки в наборах: требуется исключить неактуальные признаки, произвести масштабирование числовых атрибутов и преобразование категориальных атрибутов (для приведения всех данных к требуемому единому масштабу атрибутов используется два типа масштабирования: масштабирование по минимуму или нормализация (min-max scaling / normalization) и стандартизация (standardization)). Таким образом, можно сформировать шаблоны отбора для конкретной сети диспетчерского уровня, который, очевидно, характеризуется большей технологической специфичностью и большим количеством ограничений, чем стандартная корпоративная сеть.

Далее необходимо провести определение зависимостей между признаками. Для этого требуется вычислить стандартный коэффициент корреляции (коэффициентом корреляции Пирсона(r))[4].

$$r_{xy} = \frac{\sum_{i=1}^m (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^m (x_i - \bar{x})^2 \sum_{i=1}^m (y_i - \bar{y})^2}} = \frac{cov(x,y)}{\sqrt{s_x^2 s_y^2}}, \quad (6)$$

где $x^m = (x_1, \dots, x_m)$, $y^m = (y_1, \dots, y_m)$ – выборки, \bar{x}, \bar{y} – выборочные средние x^m, y^m , s_x^2, s_y^2 – выборочные дисперсии, $r_{xy} \in [-1, 1]$.

После обработки набор данных имеет n -количество числовых признаков (в проводимом эксперименте 53 признака, т. е. 37 из исходных данных и 16 признаков были сформированы при кодировании категориальных атрибутов). Таким образом, каждый анализируемый сетевой пакет будет представлен как некий параметр в n -мерном векторном пространстве. При этом опытным путем было выяснено, что для классификации сетевых пакетов в задачах анализа данных, лучшей моделью оказался: градиентный бустинг (GradientBoostingClassifier). Таким образом, формируется модель, позволяющая, с учетом всех возможных закономерностей, которые характерны для выбранной сетевой инфраструктуры, классифицировать данные при анализе трафика в процессе аудита.

Однако выбранные интеллектуальные методы ограничены спецификой обучаемой модели и необходимостью приведения к требуемому виду данных, т. е. данные должны быть одного порядка. С другой стороны, подобный механизм позволяет посредством обучения и выработки приемлемой модели классификации учесть динамику изменений сетевых угроз с учетом их специфики, что в системе HIDS, основанной на статических правилах, сделать невозможно. При этом, учитывая специфику встраивания OSSEC в сеть диспетчерского уровня и принципы формирования правил, сложно с помощью хостовой системы обнаружения создать единый контур аудита, т. е. когда управление и анализ отслеживаемых событий будут полностью централизованным. Однако подобные системы позволяют учесть все виды событий в сети и, более того, позволяют частично контролировать трафик уровня контроллеров.

1.5. Особенности сбора данных в многомодульной системе обнаружения вторжений

Перспективные системы обнаружения сетевых вторжений (СОВ) разрабатываются как программное обеспечение, функционал которого можно расширять или модифицировать за счет дополнительных модулей обнаружения сетевых атак. Эти модули могут использовать различные методики и принципы анализа. Они функционально изолированы, поэтому их подключение не вызовет проблем. Кроме того, они функционально взаимодополняют друг друга.

Для уменьшения нагрузки на вычислительные мощности аппаратной платформы системы, уменьшения времени сбора и анализа параметров трафика требуется определить избыточный набор собираемых данных, который пригоден для первоначального поиска признаков

угроз. И также нужно сформировать отдельный набор, который включает данные для анализа на предмет выявления конкретных атак.

Данные признаки используются для того, чтобы выявить характеристики угроз и построить связи между правилами и сами правила. Поэтому главной задачей при разработке набора правил распознавания было выявление наиболее частотных параметров (маркеров) сетевых угроз.

Маркеры сетевой разведки поделены на два класса: *явные* и *косвенные*. Для построения правил необходимы *явные признаки*. *Косвенные признаки* используются для увеличения или уменьшения вероятности правильной гипотезы о наличии разведки.

Были выявлены параметры, характеризующие состояние. К таким параметрам и характеристикам (условиям) сети при ведении разведки можно отнести:

- а) флаги TCP-пакетов;
- б) если порт узла-источника равен 0, то это является нарушением, так как нельзя использовать 0 порт;
- в) несоответствие указанных контрольных сумм пакетов с их оригинальными суммами.

По результатам проведенных экспериментов было выявлено, что в TCP-трафике можно заметить явный рост нагрузки при реализации сетевого исследования. Этот рост нагрузки отличается следующим:

- количеством принятых пакетов;
- продолжительностью по времени.

Если рассматривать трафик в целом, то можно выделить следующие признаки атаки:

- запрос к закрытым портам;
- неправильная последовательность флагов при открытии соединения или в ходе соединения (если соединение отсутствует и при этом принимается пакет с флагом отличным от SYN, то это означает нарушения порядка соединения, что указывает на ведение атакующих действий);
- множественная попытка открытий соединений за один промежуток времени, возможно, с одного IP-адреса;
- неправильные контрольные суммы TCP- и UDP- пакетов;
- если полученный пакет имеет неправильную комбинацию флагов;
- неправильная последовательность флагов при открытии соединения или в ходе соединения, например: ACK пакет до SYN;
- не бывает пакетов с одним FIN-флагом или без флагов;
- попытки множественного открытия соединений;

- обращение с портов, которые открыты для передачи данных;
- IP-адрес назначения совпадает с IP-адресом хоста;
- последовательный опрос портов.

Из приведенных наборов параметров, которые постоянно встречаются при анализе в различных модулях, за исключением атаки ARP-spoofing, можно выделить следующие данные и особенности (условия) их анализа:

1. Количество принятых пакетов.
2. Фиксация размера пакетов.
3. IP-адреса сетевых узлов (с учетом статистики их подключения).
4. ICMP-пакеты.
5. Сбор значений TTL.
6. Общее количество пакетов TCP.
7. TCP-пакеты с ACK-флагом.
8. TCP-пакеты с SYN- флагом.
9. TCP-пакеты с FIN- флагом.

При анализе атак ARP-spoofing следует учитывать специфику протокола, которая резко сужает набор параметров. Однако и в этом случае требуется работать с данными по протоколу IP.

К уникальным собираемым данным и особенностям анализа, которые следует учитывать в отдельных случаях, можно отнести следующие:

1. ICMP-ответов без запроса.
2. UDP-пакеты.
3. Запросы к закрытым портам.
4. Статистика с неправильными комбинациями флагов.
5. Множество открытия соединений.
6. Последовательный опрос портов.
7. Фиксация в TCP-пакете порта источника равен 0.
8. Число полуоткрытых соединений.
9. Пары IP-адрес и MAC-адрес в ARP-reply и операции сравнения.

Отдельно следует учитывать TCP-пакеты с URG-флагом, поскольку пакет TCP-протокола фиксируется вне зависимости от содержащегося внутри флага.

Таким образом, результаты рассмотрения собираемых параметров трафика указывают на целесообразность включения в набор данных модуля первичного сбора и анализа следующих функций и наборов данных: количество принятых пакетов, функции отдельного анализа TCP-пакетов с FIN-флагом, фиксации размеров пакетов. Также для разгрузки вычислительных возможностей следует из первичного анализа исключить: статистические данные по запросам UDP-запросов с разных IP,

упростить анализ пакетов с флагом АСК (не учитывать пакеты без предварительного установления соединения), упростить анализ ICMP-пакетов на наличие ICMP-ответов без запроса.

1.6. Особенности аудита информационной безопасности АСУТП

Информационная защита автоматизированных систем, применяемых в технологических комплексах различных предприятий, должна создаваться с учетом требований руководящих документов, изданных государством (приказ ФСТЭК России № 31 от 14.03.2014, руководящий документ «Базовая модель угроз безопасности информации в ключевых системах информационной инфраструктуры» (утв. ФСТЭК России 18.05.2007) и др.). Соответственно и процедура проведения аудита должна строиться с учетом заявленных в документах требований. Так же следует учесть то, что специфика организации АСУТП не всегда позволяет реализовать все методы исследования, которые применяются при аудите корпоративных информационных систем, основанных на клиентско-серверной архитектуре.

Основной целью проведения исследовательских работ в процессе аудита является определение степени надежности (защищенности) функционирования и надежности связи информационных блоков и подсистем. При этом необходимо учитывать, что подсистемы управления АСУТП должны совмещаться с информационными подсистемами пользовательского типа, которые применяются на верхних уровнях общей автоматизированной системы управления предприятием или организацией. В итоге это приводит к необходимости применять не только комплексный подход, предполагающий исследование информационных систем и определение класса ее защищенности с учетом разных позиций (организационной, программно-аппаратной, и др.). Требуется использовать многоуровневый подход, предполагающий исследование системы управления по уровням технического и информационного обеспечения с учетом функциональной нагрузки подсистем. Это позволит эффективно осуществить поиск уязвимостей, реализация которых может вызвать сбой в функционировании как всей системы информационного сопровождения, так и отдельных ее частей.

Своеобразие аудита АСУТП связано с тем, что исследуемая система частично может работать в реальном режиме времени, который может быть совмещен и с виртуализированной обработкой данных на

уровнях общего управления и контроля. Нужно учитывать особенности технологического обеспечения, а именно контуры управления, которые представлены как автоматизированными системами, так и автоматическими.

При аудите должны исследоваться и тестироваться следующие уровни: уровень планирования, уровень управления, уровень диспетчерского управления, уровень автоматического управления, полевой уровень. При этом необходимо учитывать, что уровни, относящиеся к планированию и управлению производством, лишь косвенно участвуют в работе АСУТП и фактически представляют собой отдельные части КИС. Далее требуется учесть особый характер передаваемой информации, которая связана с определенными правилами трансляции и обработки данных уровня системы. Если критически важную информацию для систем уровня КИС возможно выделить стандартными методами, то информацию, нижних уровней (ПЛК и др.), к которой относятся измерительные, сигнализационные данные (т. е. информация быстро меняющаяся), а также управляющая информация, дифференцировать по критерию критичности сложно в силу специфики применяемых технологий. Важность быстроменяемой информации может зависеть от динамики самой системы, требований по скорости передачи сообщений, поскольку это связано с процессом точного контроля передаваемой информации. Иные критерии в той же самой системе применяются к определению важности медленно меняющейся информации (к ней можно отнести информацию административного уровня). Кроме того, нужно учитывать наличие статической информации, представленной в виде архивов и конфигурационных файлов. Более того, аудит АСУТП должен охватывать вопросы, связанные с возможностями повреждения оборудования в результате передачи скомпрометированной информации (информационного пакета или данных настройки).

При подготовке к проведению аудита необходимо предварительно собрать информацию по возможным уязвимостям в системе из открытых источников. Необходимо выделить два класса уязвимостей. Первые относятся к уровню систем MES и типологически сходны с теми уязвимостями, которые характерны для корпоративных сетей на основе стека протоколов TCP/IP. Вторые необходимо отнести к уровню SCADA-систем и ниже. Для выяснения того, какие уязвимости характерны для исследуемой технологической инфраструктуры и ее программного сопровождения необходимо использовать различные специализированные банки данных: ICS-CERT, NVD/CVE, SCADA Strangelove, Siemens Product CERT. Так же необходимо просматривать

наборы известных эксплойтов, которые можно использовать для моделирования ситуаций вторжения и компрометации (SAINTexploit, Metasploit Framework, Immunity Canvas). При сборе данных об уязвимостях требуется учитывать специфику технологий производителя программного обеспечения АСУТП. Кроме того, нужно иметь ввиду возможное наличие сопряжений известных разноуровневых уязвимостей.

К подготовке аудита нужно отнести определение перечня руководящих документов различных организаций (ФСТЭК, ФСБ, ИСО), а также требования, которые предъявляются к описанию угроз и уязвимостей в исследуемой организации. При рассмотрении требований учитывают следующие аспекты обеспечения безопасности:

- способы и технологии обеспечения процедур аутентификации и идентификации;
- управления контроля доступа;
- обеспечения целостности системы, обрабатываемой и хранимой информации;
- организационные меры защиты на предприятии и непосредственно на рабочих местах; операторов всех уровней систем обработки и управления информацией;
- физические способы и средства защиты информации;
- средства обнаружения и изъятия программ разрушающего воздействия;
- способы и технологии управления конфигурацией систем обеспечения информационной безопасности;
- средства защиты подсистем сопряжения разноуровневых систем обработки информации;
- способы и средства защиты информации по открытым каналам связи.

Отдельно следует рассматривать процессы сопровождения:

- анализ угроз ИБ;
- оценка рисков ИБ АСУТП;
- планирование и управление непрерывностью производственных процессов.

Важно четко отметить то, на основании чего угроза, связанная с уязвимостью, признается актуальной: на основе коэффициента опасности или на основе вероятности реализации. Должна быть прописана методика определения актуальных угроз и базовая модель угроз. При этом

сопряжения разноуровневых уязвимостей и угроз здесь носит гипотетический характер. Все приведенные аспекты безопасности и сопровождающие процессы следует учитывать и непосредственно при обследовании АСУТП.

Процесс обследования разделяется на два этапа: заочное, т. е. по полученным от организации сведениям о программной и технологической базе, и очное обследование всех уровней систем обработки информации. На данном этапе определения уязвимостей имеет особую специфику. Методы интенсивного исследования, т. е. методы реп-теста, могут нанести непоправимый ущерб подсистемам АСУТП. С другой стороны, их неприменение приведет к снижению качества результатов обследования, что в итоге обесмыслит весь процесс аудита. Поэтому важно правильно определить те компоненты системы, которые можно тестировать методами агрессивного сбора информации, и те, которые не стоит подвергать воздействию. Для них должна быть выбрана другая, более щадящая методика, основанная на принципах виртуализации исследуемых процессов, построения полигонов, моделируемых сред.

При решении вопросов тестирования на проникновение удобно применять уровневую модель дифференциации систем управления предприятием. Те системы, которые относятся к верхним уровням (уровень планирования, уровень управления), построенные на модели сервер-клиент с использованием стандартных протоколов ТСР/ІР, как правило, достаточно устойчивы к реп-тесту, и в случае сбоя легко реанимируются. И совершенно обратная ситуация с нижними уровнями (уровень автоматического управления, полевой уровень). Здесь требуется избирательный подход. И совершенно особый случай представлен средним уровнем (уровнем диспетчерского управления). Именно на этом уровне находится большинство систем сопряжения разных информационных сред предприятия. Изолированность обычно задается с помощью различных средств маршрутизации и криптографической обработки. Однако это не означает полное отсечение разных компонентов разделенных по уровням подсистем. Фактически диспетчерский уровень выступает не только в роли управляющего компонента, но и выполняет функции межуровневого и межсистемного шлюза. Поэтому в связи со спецификой, отличающий данный уровень от других, его целесообразно при анализе угроз рассматривать отдельно, учитывая возможности схождения в нем разноуровневых уязвимостей при реализации атак на других уровнях.

Кроме того, необходимо отметить возможное наличие уязвимости систем и интерфейсов управления операторских станций. Системы

среднего уровня, так же, как и верхнего, построены на основе клиентско-серверной архитектуры. При этом клиентами серверной части могут быть компоненты данного уровня, так и верхних и нижних уровней. Особое внимание требуется уделить системам подключения клиентов SCADA и удаленным рабочим столам. На этом уровне частично все еще доступны стандартные средства защиты информации, в том числе, системы фильтрации трафика и шифрование передаваемых данных. Так же на данном уровне используются и такие технологии доступа коммутаторов, как веб-интерфейсы. Количество элементов, которые могут обладать в какой-то степени характеристиками сопряжения уровней в каждой АСУТП разное. Оно зависит от инфраструктуры предприятия, типа проекта, технологии, предъявляемых требований. Поэтому такие элементы целесообразно определять исходя из их функционального назначения. Функции могут быть следующими:

- контроль состояния технических платформ и компонентов, отображение данных в процессе контроля (графические и иные инструменты контроля);
- регистрация и архивация данных по рабочим процессам технических компонентов;
- оповещение с учетом эксплуатации модемов или веб-контента;
- управление пользователями: назначение и ограничение прав пользователей;
- формирования отчетов с учётом эксплуатации серверов документации SCADA-систем;
- передача данных с использованием протоколов HTTP, HTTPS, а также веб-служб;
- передача данных в файлах XML-формата;
- применение скриптов для автоматизации эксплуатационных процессов различных компонентов системы управления.

Чтобы определить, какие программные компоненты можно тестировать и в каких режимах, предварительно необходимо их дифференцировать по следующему критерию: в каком направлении движется исходящий сигнал от компонента среднего уровня. Предполагается что сигналы, пакеты данных, направленные на нижние уровни, указывают на системные компоненты чрезвычайно уязвимые к тестам на вторжение. Такие системы целесообразнее моделировать в специальных виртуальных средах и на стендах, включающих реальные или виртуально представленные ПЛК и элементы, связанные с ними. Для системных компонентов диспетчерского уровня, передающих данные с верхних и отправляющие данные к компонентам верхних уровней (при выяснении

всех особенностей функционирования подсистем), проведение активного тестирования возможно. При рассмотрении возможностей компрометации в этих условиях необходимо предварительно создать модель угроз и модель нарушителя, с развернутым перечнем графов атаки для данного уровня систем. Можно выделить следующие направления атак:

- нарушение работоспособности подсистем SCADA и смежных с ней систем;
- захват контроля энергосети;
- компрометация интерфейсов и терминалов защиты;
- перехват управления техническими устройствами нижнего уровня системы управления с возможным последующим повреждением (обход, модификация блокировок; передача недокументированных, запрещённых команд).

В целом, на стенде могут быть смоделированы следующие типы атак (при этом необходимо учитывать, что некоторые атаки могут сочетаться, могут носить комплексный характер):

1. Атака «человек посередине» с перехватом трафика. Возможна при наличии клиентско-серверной архитектуры и наличии протоколов по типу ARP. Используется для негласной фильтрации и сбора данных, а также может являться частью подготовки к более серьезному воздействию на системные компоненты.

2. Подмена трафика между компонентами системы. Данная атака может быть развитием атаки «человек посередине» и позволяет выполнять различные команды, модифицировать данные трафика.

3. Внедрение программ Rootkit и РПВ в компоненты системы управления.

4. Отказ в обслуживании компонентов системы управления (DoS-атака). Возможна в клиентско-серверной архитектуре SCADA-систем.

5. Внедрение эксплойтов и использование коллизий в системном ПО.

Объектами атак могут являться следующие компоненты системы управления:

- модули контроля технологического процесс;
- клиентские модули и системная среда АРМ- оператора;
- технические компоненты (ПЛК);
- промышленные сетевые устройства;
- модуль защиты информации и подсистемы идентификации и аутентификации субъектов доступа.

Аналогичные стенды можно создавать и для нижних уровней. Они будут в целом сходны со стендами среднего уровня, поскольку функционально с ними тесно связаны. Поэтому в состав компонентов стенда можно включать компоненты и среднего и нижнего уровня:

- АРМ-оператора;
- механизмы конфигурирования системы управления;
- сервер регистрации событий;
- серверы ввода/вывода (по типу OPC);
- промышленный коммутатор;
- ПЛК с требуемыми характеристиками;
- коммутационное оборудование, блоки питания.

Выявленные в результате обследования данные с учетом трехуровневого деления уязвимостей и угроз, следует учитывать и на следующих фазах аудита:

- оценка состояния защищенности системы;
- общий анализ угроз и уязвимостей;
- разработка рекомендаций.

В итоге, при оценке защищенности АСУТП нужно учесть множество различных аспектов функционирования подсистем: режимы работы управляемых устройств, типы той информации, которой они управляются, уровни размещения подсистем и устройств, межсистемные связи. Учет всех рассмотренных факторов при аудите позволит подобрать наиболее эффективные средства снижения риска возникновения сбоев, реализации угроз в системах управления.

1.7. Проблемы аудита информационной безопасности АСУТП

Требуется учитывать особенности архитектуры АСУТП: системы и подсистемы, созданные на основе различных технологий, объединяются в одну. Это означает, что подсистемы диспетчерского контроля и полевого применения должны быть связаны с информационными подсистемами административного управления. Поэтому методы решения поставленных задач аудита будут отличаться в зависимости от свойств исследуемых подсистем на разных уровнях, а именно на уровнях:

- организации и бизнес-процессов;
- систем управления (менеджмента) и корпоративных информационных систем;
- технических систем.

При этом часть системы подобного рода, за счет эксплуатации полевого уровня, может работать в реальном режиме времени. Он может совмещаться с виртуализированной обработкой данных на уровнях общего управления. Поэтому процедуры аудита, связанные с реализацией активного исследования, будут различаться в зависимости от технологических основ подсистем. Кроме того, нужно учитывать особенности технического обеспечения, а именно контуры управления, которые представлены как автоматизированными системами, так и автоматическими. Очевидно, в таких системах не всегда возможно реализовать тестирование. При этом выявленные результаты требуется рассматривать, используя методологии вычисления и анализа рисков с учетом особенностей исследуемых подсистем и уровней анализируемого объекта. Таким образом, первое затруднение, с которым сталкивается аудитор на предприятии с внедренным АСУТП – это создание модели аудита, которая учитывает при реализации процедур проверки все технологические различия подсистем, объединенных в едином комплексе. Соответственно, чтобы разработать модель аудита, требуется предварительно отдельно создавать формальные описания объектов разного уровня (КИС, АСУТП уровни), описания актуальных угроз, и отдельно определять процессы аудита для каждого сегмента инфраструктуры предприятия. Подобным образом формируется модель аудита.

В модели следует учитывать следующие уровни и сегменты:

- уровень планирования;
- уровень управления;
- уровень диспетчерского управления;
- уровень автоматического управления;
- полевой уровень.

Необходимо отметить, что фактически формируется две модели аудита (при описании, в том числе, актуальных угроз). Очевидно, будут прослеживаться резкие отличия моделей аудита уровней, относящиеся к планированию и управлению производством, от моделей уровней и сегментов АСУТП (эти отличия могут быть также связаны с определенными правилами передачи и обработки данных уровней системы). Это означает, что будут прослеживаться разные подходы к проведению аудита в разных сегментах системы предприятия.

Разность подходов к аудиту обусловлена и тем, что контролируемую информацию, важную для систем корпоративного уровня, можно обработать стандартными методами, в то время как данные ПЛК, сервер-

ров АСУТП, измерительные, сигнализационные данные, т. е. информацию, поставляемую в режиме реального времени, следует анализировать, используя специальное оборудование и программное обеспечение, способное разбирать и интерпретировать пакеты данных протоколов промышленного типа. Соответственно, критерии оценки критичности информации различных уровней АСУТП будет также различаться, как и методы анализа и оценки информации, что сказывается на выборе общего теоретического подхода к проведению аудита.

Особенности работы с АСУТП проявляются уже при подготовке к проведению аудита, когда требуется предварительно собрать информацию об уязвимостях в системе из открытых источников. В соответствии с делением на типы сегментов делятся и классы уязвимостей:

- уязвимости уровня MES и схожие с ними (они в основном связаны операционными средами и средствами, программным обеспечением, передачи данных на основе стека протоколов TCP/IP);
- уязвимости уровня АСУТП, в том числе SCADA-систем, если они присутствуют, и уязвимости полевого уровня (для выяснения этого можно использовать специализированные банки данных: ICS-CERT, NVD/CVE, SCADA Strangelove, SiemensProduct CERT).

При проведении инвентаризации требуется учитывать наличие клиентско-серверной модели, актуальной для всех уровней инфраструктуры предприятия со встроенным АСУТП. Это предполагает в большинстве случаев применение стека протоколов TCP/IP (при активном аудите, как правило, проблем с поиском уязвимостей, в силу распространенности не вызывает). Иная ситуация складывается с протоколами нижних уровней (уровень автоматического управления, полевой уровень). Здесь сосредоточены как средства принадлежащие к верхним уровням, так и компоненты контроля элементов полевого уровня со своим спектром решений в области трансляции данных. Здесь требуется избирательный подход. На границе нижних и верхних уровней могут находиться технологические решения со своей структурной спецификой, сопрягающие эти технологии (различные решения SCADA, OPC-серверы).

Если компоненты подсистем среднего и полевого уровня строго локализованы, то компоненты верхних уровней могут быть распределены. Поэтому при инвентаризации ресурсов и активов предприятия с АСУТП отдельно следует учитывать различные средства межуровневого сопряжения по трафику и средства изолирования компонентов одного уровня от другого. Это необходимо будет реализовать для того, чтобы

четко понимать границы аудита технологической сети с ее специфическими правилами функционирования, и корпоративной сети с привычной и стандартизированной структурой организации компонентов систем.

Фактически диспетчерский уровень выступает не только в роли управляющего компонента, но и выполняет функции межуровневого и межсистемного шлюза. Поэтому, в связи со спецификой, отличающий данный уровень от других, его целесообразно при анализе угроз рассматривать отдельно, учитывая возможности схождения в нем разноразрядных уязвимостей при реализации атак на других уровнях.

Угрозы у перечисляемых уровней и сегментов разные, поэтому при аудите вызывает затруднение описание ассоциативных связей угроз и инвентаризируемых компонентов. Это в свою очередь влияет на порядок проведения аудита. В этом случае процесс обследования, который разделяется на два этапа (заочное, т. е. по полученным от организации сведениям о программной и технологической базе, и очное обследование всех уровней систем обработки информации) будет ограничен, поскольку:

- во-первых, заочный аудит не применим к технологической сети из-за ее специфики функционирования;

- во-вторых, методы интенсивного исследования, т. е. методы реп-теста, могут нанести непоправимый ущерб подсистемам АСУ ТП, что в итоге обесмыслит весь процесс аудита.

И с этих позиций активный аудит, если он конечно возможен, будет весьма избирателен. Это в итоге может поставить под сомнение объективность исследования. Кроме того, полученные результаты нельзя экстраполировать на компоненты технологической сети в силу их ограниченной достоверности: они будут свидетельствовать о состоянии конкретного ресурса отдельного сегмента.

Соответственно в модель аудита требуется отдельно внести те сегменты и компоненты, которые на этапе активного аудита можно использовать в качестве целевых ресурсов, и особо отметить те, которые не стоит подвергать воздействию. Для них должна быть выбрана другая методика, без агрессивного воздействия или предполагающая использование стендовых решений. И поскольку правила и критерии определения работоспособности подобных систем отличаются от распространенных систем на основе стека IP/TCP, то следующей проблемой становится определение критериев нарушения в технологической сети. Основным показателем реализации угрозы являются задержки при транс-

ляции пакетов с техническими данными или в работе комплекса программно-аппартной платформы. Они могут свидетельствовать как о появлении паразитной нагрузки на трафике, так и о воздействии на сегменты, участвующие в трансляции, внешних источников или о несогласованности работы скомплектованного оборудования, что уже не является областью ИБ. В этом случае процесс аудита будет осложнен дополнительными процедурами из-за потребности отличать признаки собственно технических неисправностей и признаки проявления инцидента ИБ, которые в технологической сети, на диспетчерском и полевом уровне схожи.

Все эти особенности влияют на формирование модели нарушителя при ее формализации, в том числе, на формирование критериев нарушения, что необходимо при реализации аудита и сценариев действий нарушителей различных категорий.

При решении вопросов активного аудита можно использовать уровневую модель дифференциации систем управления АСУ предприятия (при выявлении уязвимостей). В этом случае могут помочь при моделировании атак наборы известных эксплойтов (SAINTexploit, MetasploitFramework, ImmunityCanvas).

Отдельно стоит обратить внимание на возможное наличие уязвимости систем и интерфейсов управления операторских АРМ (автоматизированных рабочих мест). Системы диспетчерского уровня функционируют на основе клиентско-серверной архитектуры. При этом клиентами серверной части могут являться как компоненты того же уровня, так и верхних уровней, включая службы удаленного управления и контроля технологических процессов. Это означает, что технология тонких клиентов и удаленных рабочих столов широко распространена на современных предприятиях. На этом уровне возможна установка стандартных средств защиты информации, в том числе, системы фильтрации трафика и шифрования передаваемых данных. Однако ее применение приводит к задержкам при передаче данных технологического характера, как внутри управляемого сегмента, так и при передаче технической информации за его пределы. Более того, могут использоваться веб-интерфейсы для доступа к управляющим компонентам на месте осуществления технических операций. Поэтому, при аудите трафик и элементы, информация от которых должна поставляется на АРМ, в том числе удаленных, фактически в режиме реального времени, как и доступ к технологической сети самих АРМ, целесообразно отделять от остального массива компонентов системы предприятия.

Соответственно, требования к процедуре и критерии безопасности необходимо каждый раз корректировать при столкновении с новой технологической базой. Это в свою очередь осложняет расчет рисков при формировании итогового отчета. Более того, сопоставление разноуровневых уязвимостей и угроз будет носить гипотетический характер. Поэтому требуется, чтобы методика практического анализа результатов аудита отражала специфику исследуемого объекта. В целом, можно использовать комбинирование анализа рисков и анализа стандартов ИБ, при условии, что эти стандарты безопасности, учитывая специфику технологий, в том числе зарубежной, применимы, существуют или являются актуальными. Это сужает возможности аудита на соответствие стандартам и, наоборот, расширяет поле анализа на основе аудита рисков. При этом нужно четко понимать и ограничивать возможности тестовой компрометации узлов сети, в соответствии с требованиями обеспечения надежности ее работы в реальном времени. Это так же сужает поле применения методик анализа рисков.

Такой проблемы не возникает, когда выполняются задачи аудита по отношению к административному уровню. На нем требования к информации по скорости передачи существенно ниже (к ней можно отнести информацию административного уровня). Кроме того, нужно учитывать наличие на ЭВМ статических наборов данных, представленных в виде архивов и конфигурационных файлов. Контроль целостности можно осуществлять периодически.

Принимая во внимание специфичность требований предъявляемых к технологической сети, целесообразно первоначально проводить тестирование на устойчивость к отказам при помощи наблюдения за ее работой в конечном наборе специально выбранных ситуаций. При этом тесты должны быть четко дифференцированы по возможностям воздействия на состояние системы, и те, которые вызывают подозрения должны проводиться на специально сформированном стенде. Таким образом, еще одним осложнением, возникающим при аудите, является создание специализированных лабораторных стендов, учитывающих специфику технологической сети конкретного предприятия. Можно выделить следующие задания, связанные проверкой на стенде устойчивости системных компонентов:

- проверка надежности обеспечения управления питанием энергосети;
- проверка надежности обеспечения работоспособности интерфейсов и терминалов защиты;

– проверка надежности обеспечения управления техническими устройствами нижнего уровня системы управления с возможным последующим повреждением (обход, модификация блокировок; передача недокументированных, запрещённых команд) [7].

Тестовые испытания направлены на проверку требований к исследуемой системе. При этом используемые задачи в тестах должны обеспечивать проверку всех возможных вариантов поведения технологической сети и ее компонентов, иначе выводы о качестве системы, сделанные на основе проведенного тестирования, будут недостоверны [6].

Тестирующие задания легче всего построить и классифицировать на основе функций компонентов системы. Функции могут быть такими:

– контроль состояния компонентов технологической сети, отображение данных в процессе контроля (в том числе, применение графических и иных средств отображения процесса контроля);

– регистрация и архивация данных по рабочим процессам технических компонентов;

– оповещение о различных событиях с учетом передачи данных через модемы;

– контроль распределения прав доступа пользователей на различных уровнях технологической сети;

– формирование отчетов с учётом эксплуатации серверов документации серверов АСУТП, SCADA-систем, OPC-серверов;

– трансляция данных по протоколам HTTP, HTTPS;

– использование файлов XML-формата для трансляции данных и их хранения [7].

Если рассматривать аудит по структуре АСУТП, то на технологическом уровне при активном аудите целесообразно применять модульное (компонентное) тестирование. Оно позволяет проверить безопасность функционирования отдельно взятого элемента исследуемого объекта (например, программного или аппаратного модуля, подпрограммы, отработки должностных обязанностей по защите информации отдельным должностным лицом и т. д.). Методика определения рисков в этом случае будет наиболее выгодна, поскольку позволяет проанализировать покомпонентно состав сегмента системы и, более того, учесть последовательность поэтапного воздействия на анализируемый компонент при его тестовой компрометации. Особенно данная методика проявляет себя хорошо при работе с испытательными стендами. Но, очевидно, учесть системные связи межуровневого характера при использовании тестового стенда можно только в редуцированном виде.

Таким образом, межуровневый характер отношений сегментов и компонентов технологической сети можно проследить в аудите, используя интеграционное тестирование (взаимодействия между элементами объекта проверяется путем реализации методов тестирования «сверху вниз», «снизу вверх», распределения потоков управления и данных и т. д.) [6]. Однако состав сети и требования по непрерывной трансляции технических данных ограничивают данный метод в практике применения только теми компонентами сети, тестирование которых не вызовет коллизий (число и тип подобных компонентов определяется при подготовке к «активному» аудиту).

Для получения целостного представления о состоянии компонентов проверяемой инфраструктуры требуется использовать системное тестирование, которое охватывает целиком весь объект и его внешние интерфейсы, а также среду функционирования. Однако в силу разности технических платформ, особенности требований к надежности компонентов технологической сети, осуществить в целом системное тестирование чрезвычайно трудно.

Чтобы более подробно проанализировать программные компоненты системы на предмет определения особенностей и подходящих режимов тестирования, предварительно необходимо их классифицировать по следующему критерию: в каком направлении движется исходящий сигнал от компонента среднего уровня. При этом необходимо отметить, что данные, направленные на нижние уровни, во-первых, принадлежат медленному трафику, а во-вторых, направление потока данных указывает на потенциальные системные узлы и сервисы, которые окажутся в поле интересов злоумышленника, действующего извне.

Таким образом, при аудите АСУТП возникает много затруднений. Во-первых, это сложность создания модели аудита, в которой необходимо учесть все технологические особенности чрезвычайно различных подсистем. Во-вторых, потребуются использовать разные подходы к проведению аудита в технологически отличных сегментах системы предприятия. Более того, требования к процедуре и критерии безопасности необходимо каждый раз корректировать при столкновении с новой технологической базой. В-третьих, критерии оценки критичности информации различных уровней АСУТП будут также различаться. В модели аудита необходимо учитывать отдельно те сегменты и компоненты, которые на этапе активного аудита не стоит подвергать воздействию, что налагает ограничения на процедуру тестирования.

В целом, надо отметить, что процедура аудита АСУТП сегментов предприятия характеризуется чрезвычайной требовательностью к описанию и определению практик аудита фактически к каждому сегменту системы и, в частности, компоненту технологической сети.

1.8. Особенности подготовки активного аудита информационной безопасности АСУТП

Изначально, перед аудитом требуется провести комплексное исследование инфраструктуры АСУТП: определить специфику передачи информации, протоколы промышленного типа, способ распределения информация, службы, функционирующие в границах уровней АСУ. Так же необходимо распределить системные компоненты и актуальные протокольные среды по следующим уровням:

- уровню КИС (корпоративная информационная система);
- уровень (диспетчерский уровень), на котором совмещаются компоненты и протоколы промышленного типа и уровня КИС (сервисы и протоколы семиуровневой модели OSI);
- уровень, на котором совмещаются компоненты и протоколы обработки промышленной информации (полевой уровень).

Далее необходимо решить следующие задачи.

1. Сбор информации с уровневим разделением (инвентаризация).
2. Требуется провести поиск уязвимостей (при решении этой задачи предполагается не только аналитика транслируемой информации, но и активное воздействие на нее и на ее источник). При этом поиск уязвимостей может быть реализован даже в организационной составляющей комплекса ИБ. Процедура активного сбора и активного аудита технической составляющей предполагает определение:

- будут ли задержки при передаче данных;
- будет ли искажения информации при передаче данных;
- будут ли изменяться настройки систем передачи данных и т. д.

3. Анализ угроз, обнаруженных уязвимостей, и вычисление рисков, с последующим определением уровня защищенности.

Так как условия проведения аудита ИБ всей инфраструктуры предприятия разные, как и число задач на различных уровнях АСУТП предприятия, следует выделить две модели аудита:

1. Модель аудита, которая относится непосредственно к верхним уровням, модель КИС.

2. Модель аудита нижних уровней (три уровня, на которых активно используются промышленные протоколы). Она будет содержать

дополнительные задачи, связанные с вычислением успешной реализации процедур аудита, и времени его безопасной реализации. На основе анализа полученных в результате вычислений данных необходимо принимать решение о целесообразности применения активного аудита на рассматриваемых уровнях.

Поскольку основная проблема активного аудита в сетевых системах реального времени связана именно с задержкой времени передачи данных, то очевидно, что угроза, которая существует в системе, не относится к типичным видам угроз ИБ (перехват данных, подмена данных, раскрытие конфиденциальной информации). Основной угрозой при активном аудите являются именно возникновение задержки времени и/или недопустимого периода задержки пересылаемого набора данных. Это препятствует трансляции данных в режиме real-time, вызывая их недопустимое по технологическим спецификациям запаздывание. Поскольку воздействие активного аудита приводит к тем же результатам, что и целенаправленная атака, их можно отождествить, с той лишь разницей, что тип угрозы при аудите заранее известен и по способу реализации неизменен. Соответственно, перед проведением аудита требуется рассмотреть проблемы возникновения рисков ИБ, которые могут привести к повреждению системы и, соответственно, прекращению самих процедур аудита.

Таким образом, этапы анализа будут включать не только выявление угроз и уязвимостей, определение рисков, но и предварительный анализ безопасности процедуры аудита на основе априорных оценок рисков негативного воздействия инструментария аудита, возникающих при исследовании систем АСУТП. Это отличительная особенность реализации активного аудита в системах реального времени (разного типа) на предприятиях с распределенными функциональными уровнями АСУТП. Соответственно, в список задач требуется внести дополнительную задачу определения безопасности процедур аудита для исследуемой инфраструктуры.

Поскольку угроза при процедуре аудита была определена как временной период, за который система утрачивает функциональность, необходимо подобрать такую модель расчета успешности аудита, которая будет учитывать специфику используемых параметров, т. е. позволит вычислить вероятность появления задержки во времени, период ее существования и период штатного функционирования системы при активном аудите. Такая модель может быть основана на подходах определения надежности функционирования систем.

Предполагается, что период, учитываемый в расчетах, охватывает время сканирования инфраструктурных компонентов. Любая задержка трансляции информационных пакетов промышленных протоколов типа CAN, ModBus, HART приводит к нарушению режима работы real-time и, соответственно, будет считаться отказом (поскольку переданная информация не пришла к адресату вовремя). Тогда интенсивность отказов будет трактоваться как интенсивность (среднее число) фиксаций задержек при передаче данных в процессе активного аудита ИБ в режиме real-time. Несмотря на то, что угрозы задержек трансляции на каждом уровне типологически тождественны, длительность задержки, которая будет являться критичной для каждого уровня, будет своя. Соответственно, интенсивность отказа системы передачи данных по промышленным протоколам равна сумме интенсивностей задержек, фиксируемых при аудите на каждом уровне АСУТП, и рассчитывается по следующей формуле:

$$A = \sum_{i=1}^n a_i, \quad (7)$$

где A – интенсивность отказа системы передачи данных по промышленным протоколам;

a_i – интенсивность задержек при аудите.

Таким образом, вероятность $p(t)$ исправной работы в течение интервала времени проведения аудита t с учётом интенсивности отказов, т. е. задержек при передаче, определяется так:

$$p(t) = e^{-a_i}. \quad (8)$$

Учитывая уровневую сегментацию АСУТП, можно рассчитать для каждого уровня вероятность исправной работы и вычислить вероятность исправной работы в целом для системы, суммируя показатели всех существующих на производстве систем, связанных с передачей данных в режиме real-time, и так же вычислить среднее время работы взаимосвязанных систем АСУТП.

Однако, учитывая связанность уровней необходимо рассматривать влияние функционала одной подсистемы на работоспособность других. Поскольку это влияние определяется спецификой связей, которые на предприятии в основном носит нетипичный характер, то тогда показатель вероятности исправной работы должен отражать специфичность данных взаимосвязей и взаимозависимостей. С другой стороны, учитывая технологическую специфику рассматриваемых уровней, их

можно объединить в единый сегмент. Однако, решение рассматривать подсистемы реального времени как единый сегмент или как несколько взаимосвязанных зависит от специфики распространения протокольных сред и системных сервисов на следующих уровнях АСУТП:

- уровня управления;
- уровнях диспетчерского управления;
- уровня автоматического управления и полевого уровня.

Использование промышленных протоколов между уровнем диспетчерского управления и уровнем автоматического управления, уровнем автоматического управления и полевого управления подразумевает наличие взаимосвязей между их службами.

Далее, необходимо рассмотреть промежуток времени между двумя задержками при передаче пакетов по промышленным протоколам. Данный параметр является показателем времени успешной работы (H) системы при активном аудите до первого отказа (задержки) и называется «наработка на отказ трансляции данных при активном аудите и инвентаризации сетевых сервисов», и рассчитывается по следующей формуле[?]:

$$H = \frac{1}{A}. \quad (9)$$

Для каждого уровня этот показатель рассчитывается отдельно. Однако, если рассматривать три нижних уровня как единый сегмент автоматизированной системы, то и время наработки на отказ, соответственно, будет единым для данной сборки уровней. Таким образом, можно выяснить непосредственно время успешной работы систем аудита и инструментария инвентаризации сетевых сервисов, возможные периоды возникновения проблем с передачей данных и вероятность возникновения этих проблем в период активной эксплуатации промышленных протоколов в режиме real-time.

Поскольку в состоянии системы «как есть» (as-is) параметр времени восстановления H_r будет оцениваться как время, которое требуется для возобновления режима real-time, то, соответственно, данный параметр не следует рассматривать как нивелирование канала НСД (несанкционированный доступ). В данном случае в отличие от параметров вычислительной модели рисков возникновение угроз НСД рассматривается всего одна угроза (задержка трансляции данных), которая хорошо известна и перекрывается простым отключением инструментов исследования подсистем АСУТП.

При этом система в режиме real-time (в соответствии с моделью) возобновит свою работу так же, как если бы возобновляла свою работу система защиты, при условии того, что она была бы повреждена. Это позволяет определить коэффициент готовности K_w возобновления трансляции данных с требуемой скоростью и выявить насколько этот коэффициент готовности соответствует тому, который бы означал возможность возобновления работы системы без каких-либо критических повреждений таковой. Таким образом, коэффициент готовности системы возобновить трансляцию и работу в целом без лишних критических повреждений является ключевым показателем в модели расчетов.

$$K_w = \frac{H}{(H-Hr)}. \quad (10)$$

Соответственно, можно вычислить также коэффициент неготовности K_{nw} системы к возобновлению трансляции данных в режиме реального времени.

$$K_{nw} = 1 - K_w. \quad (11)$$

Совмещение работы систем в режиме real-time и работы в виртуальном временном режиме возможно на одном уровне АСУ предприятия. Поэтому всегда следует дифференцированно подходить к вопросу определения допустимого времени и количества задержек (нужно знать точно компонентный состав анализируемого оборудования, его характеристики, спецификации протокольных сред).

При выявлении специфики активного аудита и порядка его проведения предварительно не требуется анализировать модель нарушителя и применять модель нарушителя при обработке собранных данных. Это существенно облегчает работу по подготовке проведения безопасного активного исследования подсистем АСУТП при их непосредственной эксплуатации.

Пассивный аудит позволяет решать вопросы анализа состояния безопасности систем без взаимодействия с системными элементами, в то время как аудит активного типа при работе с компонентами АСУТП сильно ограничен. Трудность состоит в том, что компоненты автоматизированных систем управления, размещенные на разных уровнях, сильно различаются по технологии исполнения. Ограничения, прежде всего, связаны с технологиями, которые реализуются на уровнях компонентов обрабатывающих техническую информацию от полевых устройств. При этом другие уровни АСУ с данной информацией уже не

работают. Данные особенности следует учесть при определении способов анализа угроз и уязвимостей.

Кроме того, аудит активного типа при взаимодействии с компонентами АСУТП не должен нарушить технологию обработки данных и при этом в результате тестирования должен быть получен максимум информации о состоянии системы. Также и сама методика анализа результатов должна учитывать указанные особенности аудита активного типа. Следовательно, необходимо найти такую методику анализа безопасности подсистем АСУТП, которая бы учитывала, с одной стороны, уровневую архитектуру системы, а с другой стороны, технические особенности, как функционирования подсистем, так и процедур аудита [2].

При определении специфики тестирования выделяют следующие уровни АСУ: уровень планирования, уровень управления, уровень диспетчерского управления, уровень автоматического управления, полевой уровень. Предполагается, что каждый из этих уровней уязвим. Однако сами уязвимости различаются в соответствии с различиями применяемых на уровне АСУ технологий. Следует отметить, что на уровнях, расположенных над диспетчерским слоем, скорость передаваемой информации не критична так, как на полевом или диспетчерском уровне, поскольку компоненты этих (нижних) уровней транслируют данные по сети в режиме реального времени (с минимальной задержкой, при этом большие задержки по времени передачи не допустимы) [3]. Соответственно, время задержки в этом случае является показателем критичности.

Следуя требованию учитывать технологии уровней АСУ при тестировании, можно сделать вывод о том, что аудит будет заключаться в отдельном исследовании и анализе каждого технологически отличного уровня. И только после полного исследования состояния всех уровней, суммируя полученные данные, можно будет оценить общую уязвимость или защищенность системы.

При анализе результатов активного аудита удобно применять граф компрометации, который учитывает вероятностные показатели проникновения в систему с учетом всех ситуаций, которого могут возникнуть у злоумышленника, а также с учетом времени, которое затрачивает злоумышленник для достижения заданных им целей. Основным показателем уровня защищенности/незащищенности является время, затрачиваемое злоумышленником на достижения целей компрометации на каждом слое. При этом, чем больше время, затрачиваемое на компрометацию, тем выше вероятность отражения атаки.

Поскольку рассматривается проблематика сохранения работоспособности критически важных полевых устройств, ПЛК, диспетчерских систем, то априори в модели внешнего нарушителя предполагается, что

злоумышленник попытается проникнуть на нижние уровни АСУ и скомпрометировать размещенные на них компоненты. Таким образом, нарушитель должен выполнить ряд проникновений и преодолеть пространство всех слоев, которые окружают техническое ядро системы, т. е. уровень ПЛК. В случае внутреннего негативного воздействия на компоненты отдельного слоя начало атаки следует отсчитывать от точки вторжения инсайдера-злоумышленника в систему.

Кроме того, при составлении графа в компрометации следует учитывать все классы уязвимостей, которые характерны для исследуемых подсистем и являются точками входа злоумышленника в слой АСУ. Можно выделить на основе сходства технологий следующие классы уязвимостей:

- первый класс уязвимостей (относится к административному уровню);

- второй класс уязвимостей (относится к уровню scada-систем и целевому уровню, т. е. диспетчерскому уровню и уровню датчиков).

В данной методике особенно важен учет вероятностей трех состояний атаки (называемых подпроцессами), которые могут проявиться при прохождении каждого слоя, учитывая то, что каждый новый слой является для злоумышленника, при отсутствии информации от инсайдера, плохо известной, либо вовсе неизвестной областью (поэтому требуется каждый раз заново реализовать последовательность атаки). Для упрощения анализа результатов аудита, технологические схожие компоненты подсистемы и уровни, можно объединить в трех основных уровнях:

- уровень корпоративных информационных систем (MES, ERP);
- уровень диспетчеризации с использованием системы SCADA;
- уровень полевых устройств (включает: уровень систем управления полевыми устройствами, уровень контроллеров датчиков, уровень датчиков).

Соответственно упрощается формирование графа компрометации каждого обобщенного уровня. Таким образом, количество составленных графов компрометации и вычисленных периодов, затрачиваемых злоумышленниками на проникновение, равно количеству выделенных обобщенных слоев. Соответственно, сумма периодов взломов будет являться общей временной характеристикой проникновения в систему извне. Кроме того, нужно отметить, что данная методика позволяет использовать параметры пассивного аудита для вычисления времени проникновения без активного тестирования. В итоге, можно утверждать, что данная методика позволяет совместить при анализе параметры пассивного и активного аудита.

2. ПРАКТИЧЕСКИЕ АСПЕКТЫ ПРОВЕДЕНИЯ АУДИТА

2.1. Пен-тест компонентов ИС при помощи Infection Monkey

Пен-тест – это анализ системы на наличие уязвимостей. Это метод оценки безопасности информационной системы путем имитации атаки злоумышленников. Пен-тестинг проводится с позиции потенциального злоумышленника и может предполагать активное использование системных уязвимостей.

Целью тестирования является выявление возможных уязвимостей и недостатков, которые могут привести к нарушению конфиденциальности, целостности и доступности информации, спровоцировать неправильную работу системы или привести к отказу в обслуживании, а также прогнозировать возможные финансовые потери и экономические риски. Тестирование влияет как на виртуальный уровень, так и на физический.

По результатам тестирования на проникновение дается оценка возможностей текущего уровня безопасности противостоять попытке вторжения потенциального злоумышленника, данные о количестве времени и ресурсов, необходимых для успешной атаки на клиента. В случае выявления уязвимостей список рекомендаций по устранению вышеуказанных уязвимостей является обязательным.

Суть работы заключается в имитации действий злоумышленника, который намеревается получить доступ к информационным системам заказчика и нарушить целостность, конфиденциальность или доступность информации, принадлежащей заказчику. Наиболее частыми объектами исследования являются:

1. Системы управления базами данных.
2. Сетевое оборудование.
3. Сетевые службы и сервисы (например, электронная почта).
4. Средства защиты информации.
5. Прикладное программное обеспечение.
6. Серверные и пользовательские операционные системы.

Infection Monkey – это бесплатный инструмент с открытым исходным кодом для моделирования атак и взломов, разработанный Guardicore.

Infection Monkey предназначена для безопасного тестирования устойчивости вашей сети и центра обработки данных к внешним взломам и внутреннему заражению вычислительных ресурсов. Так же это

самораспространяющийся инструмент, который способен автоматически находить и визуализировать наиболее простые пути движения злоумышленника в вашей сети.

Высокоуровневая концепция Infection Monkey проста. Инструмент предназначен для поиска компьютеров, доступных в сети, и попытки их проверки с использованием различных методов, включая интеллектуальный выбор пароля и безопасные эксплойты, т. е. имитацию действий реального злоумышленника, а не автоматического сканера. Это делается для того, чтобы все системы безопасности могли обнаружить Infection Monkey.

В процессе заражения Infection Monkey предоставляет подробную информацию о конкретной уязвимости, которая была использована, а также о влиянии уязвимых сегментов на вашу сеть. Любой успех Infection Monkey указывает на сбой в вашей системе безопасности, который следует исправить.

Вот неполный список сценариев, которые могут быть реализованы с помощью Infection Monkey:

- проверка уровня компетентности вашего SOC и способности детектировать типичные таргетированные атаки;
- выявление типичных ошибок конфигурации параметров безопасности операционных систем;
- построение карты внутренней сети компании глазами хакера, анализ подверженности сети латеральному движению;
- анализ работоспособности новых средств защиты до их покупки, проверка корректности настройки и эффективности существующих СЗИ, в том числе антивируса, EDR, ханипотов и систем Desertion, систем анализа сетевого трафика, поведенческой аналитики (UEBA);
- межсетевых экранов и так далее, проверка конфигурации политик Zero Trust, проверка вашей способности детектировать и блокировать различные техники матрицы MITRE ATT&CK.

Первым шагом является установка в среде ОС, которое имитирует C&C-сервер для агентов.

После установки управляющего C&C-сервера можете запустить агент и начать анализ непосредственно с этого же сервера, или скачать и запустить агент на любой другой машине.

Запуск агента имитирует появление злоумышленника, который взломал одну из машин, или инсайдера, который начал проявлять активность в вашей сети.

После заражения машины система отправляет на свой C&C-сервер телеметрию и запрашивает конфигурацию с инструкциями по дальнейшим действиям. Если ответа нет, то она строит туннели до C&C-сервера через другие подобные системы. В случае ошибки используется встроенная отказоустойчивая конфигурация.

Затем Infection Monkey пытается украсть учетные записи, сохраненные на текущей машине, а также начинает сканирование разрешенных диапазонов IP-адресов, выявляя все доступные машины и их открытые сервисы.

Собранные данные передаются на C&C-сервер, а также используются для дальнейшей атаки и распространения на доступные машины, если это не запрещено конфигурацией.

Конфигурация Infection Monkey позволяет добавлять IP-адреса в черный список для запрета их сканирования, ограничивать глубину распространения агента и указывать разрешенные подсети для работы.

Infection Monkey разработана для обеспечения гарантированной безопасности, и не использует функции анализа, распространения и эксплуатации, которые могут повлиять на стабильность машин или сети. Infection Monkey использует несколько векторов атаки и позволяет увидеть следующие вещи:

1. Уязвимые хосты. Находит узлы со слабыми паролями, старыми версиями ПО или известными уязвимостями. Вот список эксплойтов:

- а) SMB Exploiter;
- б) WMI Exploiter;
- в) MSSQL Exploiter;
- г) MS08-067 Exploiter;
- д) SSH Exploiter;
- е) ShellShock Exploiter;
- ж) SambaCry Exploiter;
- и) ElasticGroovy Exploiter4;
- к) Struts2 Exploiter;
- л) WebLogic Exploiter;
- м) Hadoop/Yarn Exploiter;
- н) VSFTPD Exploiter.

2. Запрещенное взаимодействие. Можно обнаружить взаимодействие между сетями, которое должно быть запрещено на уровне МЭ или маршрутизатора.

3. Горизонтальное распространение. Отображение перемещения вредоносного объекта в графическом виде.

Для демонстрации работы программного обеспечения Infection Monkey необходимо организовать работу сервера, где будут произведены обязательные базовые настройки. В данной работе использовалась ОС Windows Server 2012 R2.

Выполнить

1. Требуется установить операционную систему на виртуальную машину VMware Workstation Pro. Выбираем стандартную версию с графическим ядром для обеспечения более комфортного рабочего процесса.

2. Вначале необходимо изменить стандартное имя сервера на более простое и понятное. Для изменения имени необходимо зайти в свойства системы, а затем изменить параметры, после чего выбираем **Изменить**.

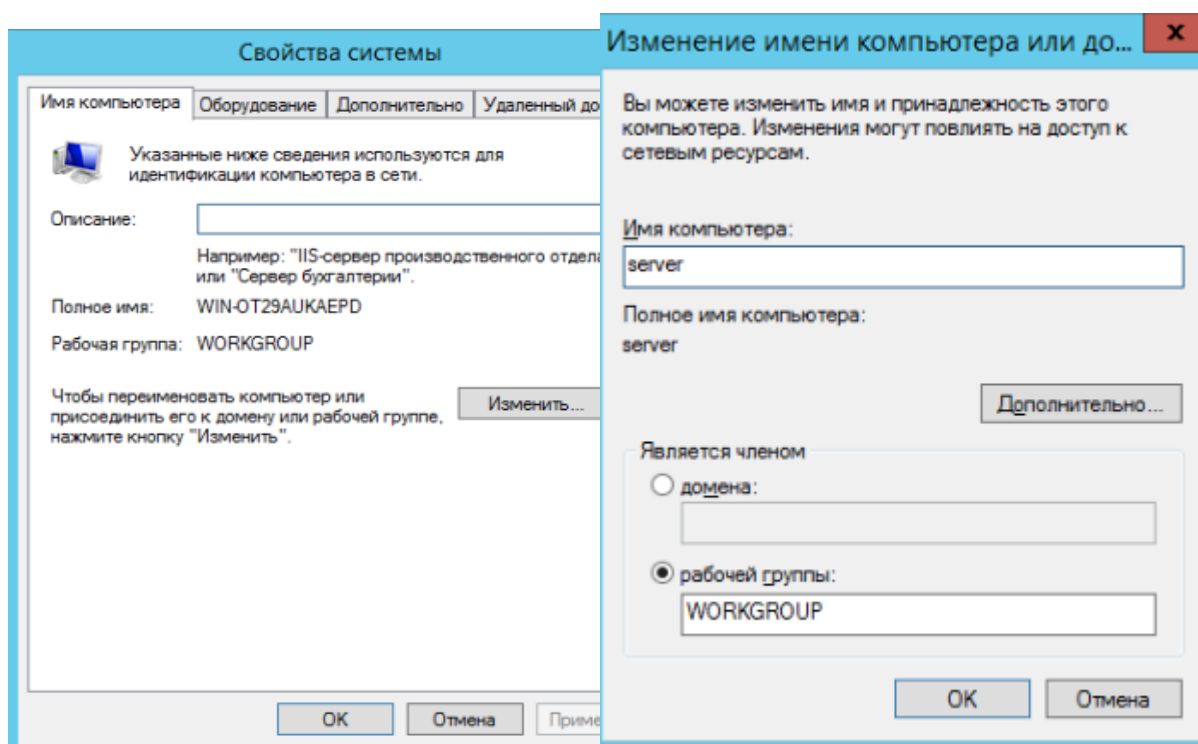


Рис. 4. Изменение имени

3. Прописываем в поле новое имя компьютера (в данном случае было выбрано имя server). А затем необходимо перезагрузить систему для того, чтобы изменения вступили в силу.

4. Теперь необходимо заняться сетевыми настройками. Узнаем информацию об IP локальной сети при помощи команды ipconfig.

5. После необходимо зайти в настройки сетевых подключений, для этого используем ncra.cpl.

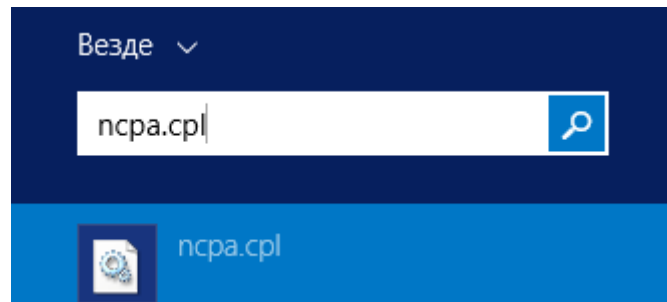


Рис. 5. Настройки сетевых подключений

6. Выбираем и переходим в свойства протокола TCP/IPv4. Вводим статические IP-адреса, маску, шлюз и предпочитаемый DNS-сервер.

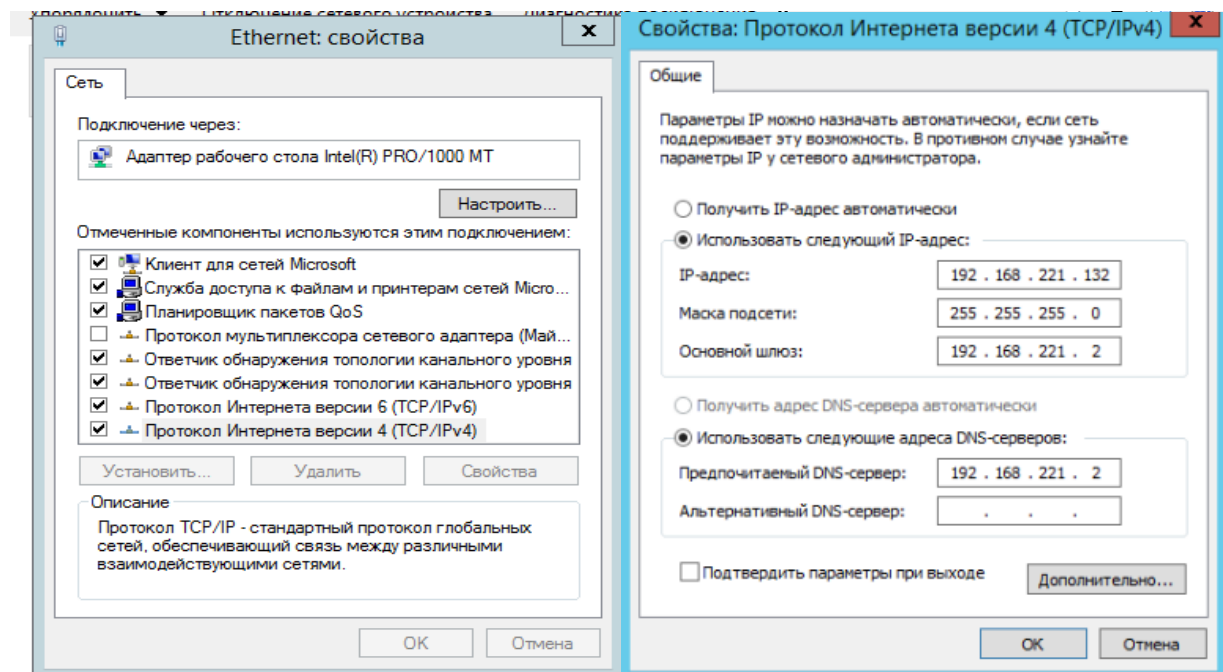


Рис. 6. Свойства протокола TCP/IPv4

7. Теперь переходим к основной настройке сервера. Открываем панель мониторинга и выбираем **Добавить роли и компоненты**.

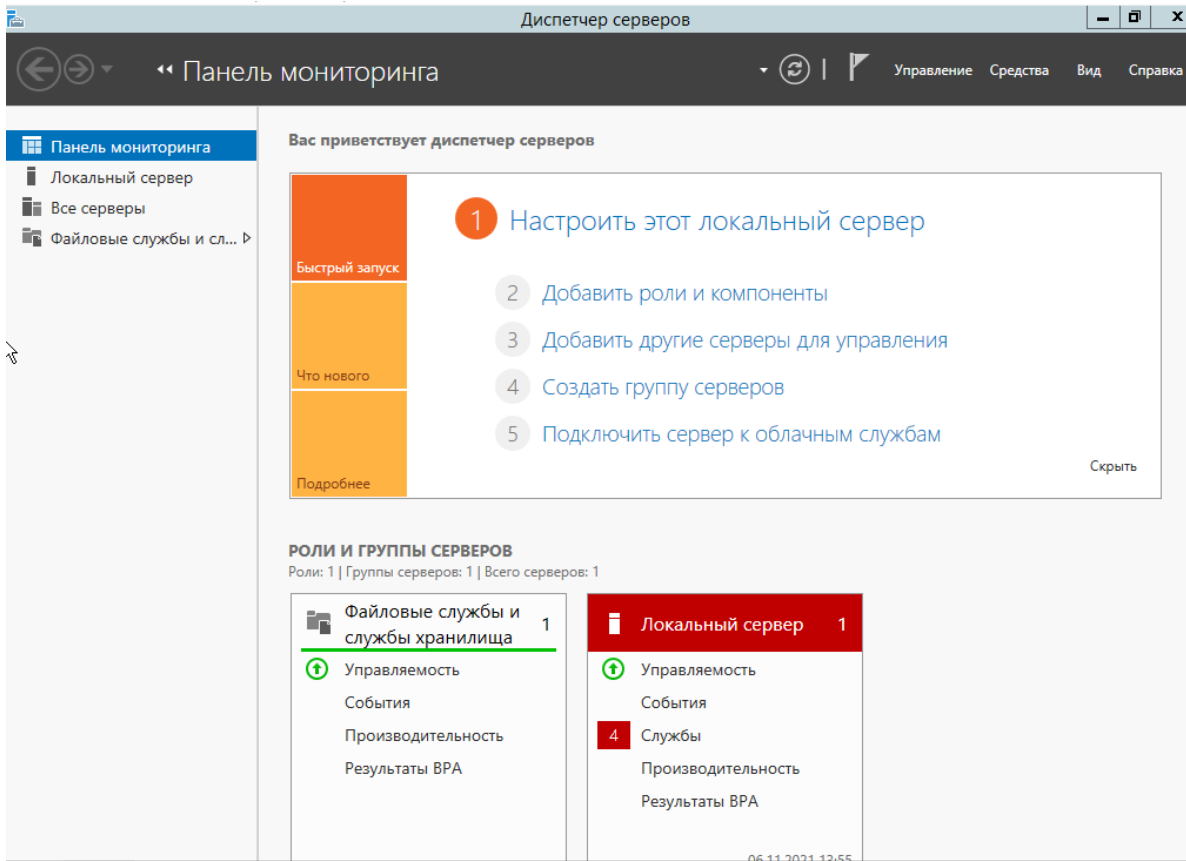


Рис. 7. Диспетчер серверов

Здесь необходимо выбрать тип установки, в нашем случае выберем **Установка ролей или компонентов**.

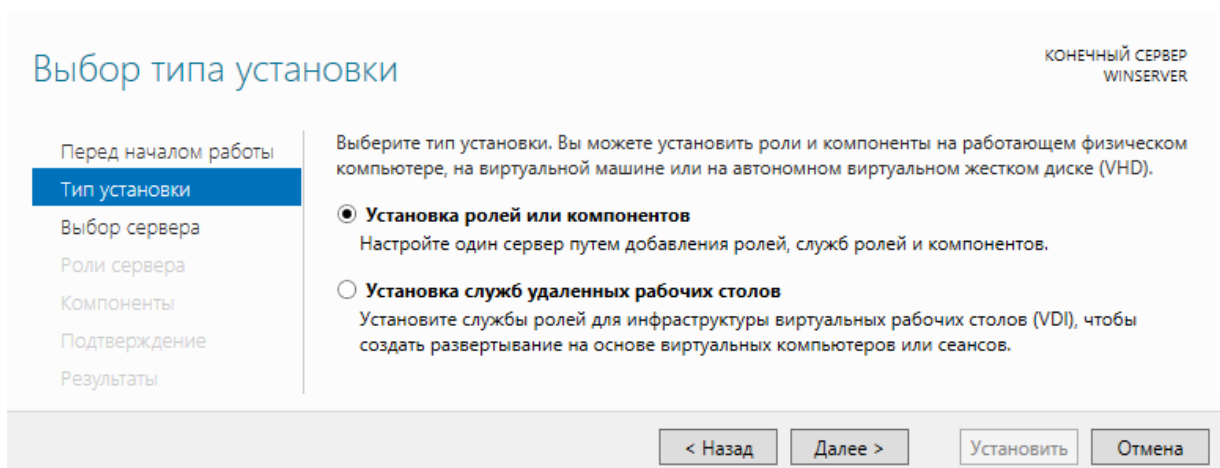


Рис. 8. Тип установки

В пуле серверов необходимо выбрать нужный сервер, который будем настраивать.

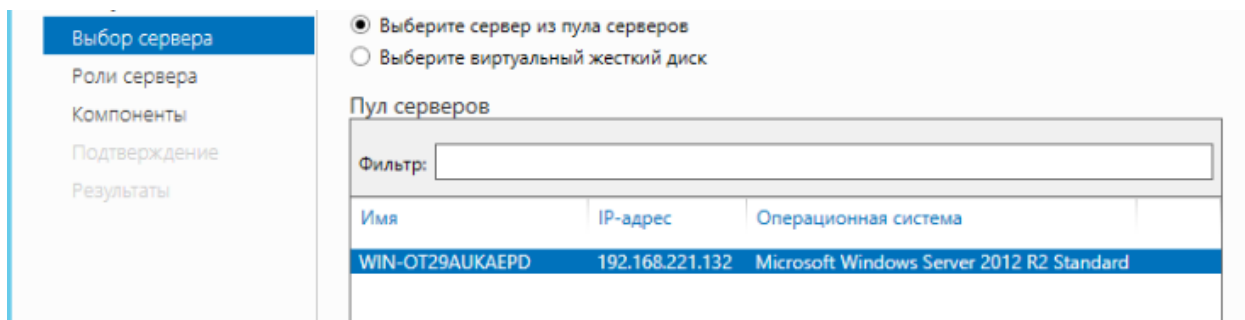


Рис. 9. Выбор сервера

8. Для базовой настройки сервера необходимо выбрать данные роли:

- а) DHCP-сервер;
- б) DNS-сервер;
- в) Доменные службы Active Directory;
- г) Службы удалённых рабочих столов;
- д) Службы хранения;
- е) также выбраны следующие компоненты.

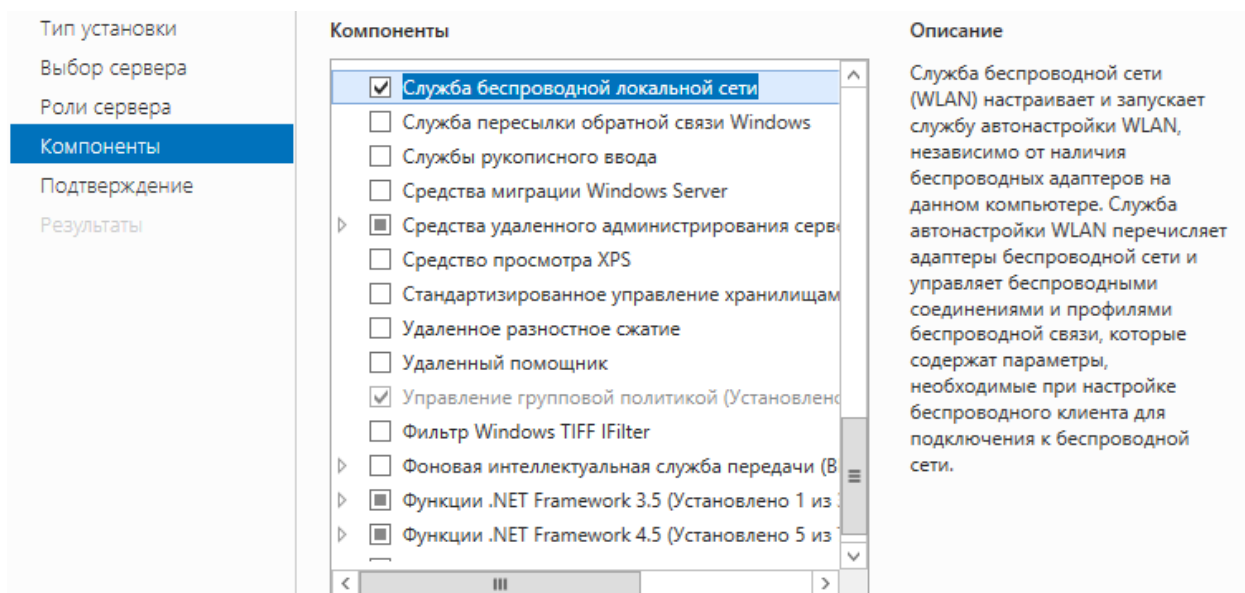


Рис. 10. Выбор компонентов

9. Определяем необходимые службы ролей.

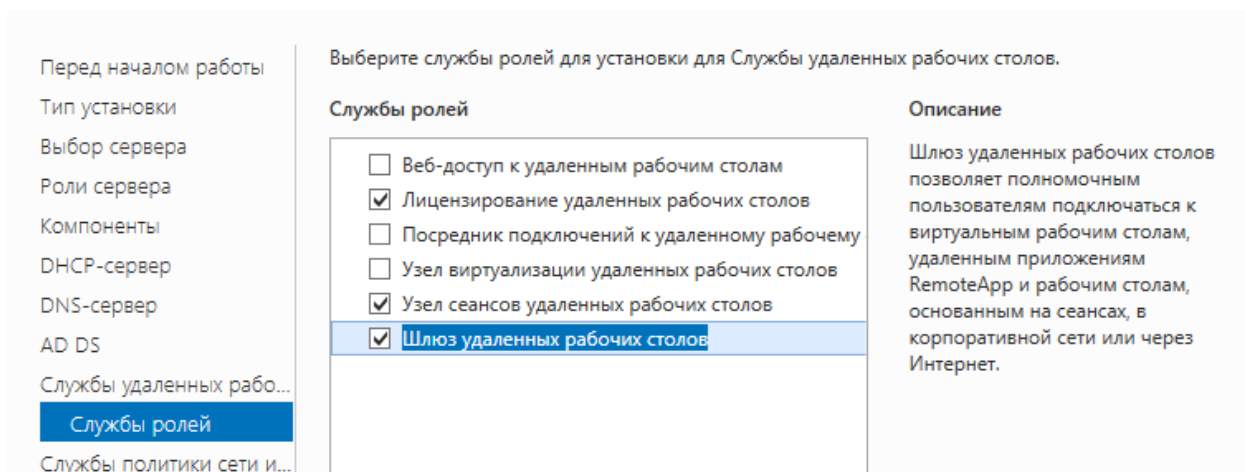


Рис. 11. Выбор служб ролей

Определяем службы ролей для политики сети и доступа.

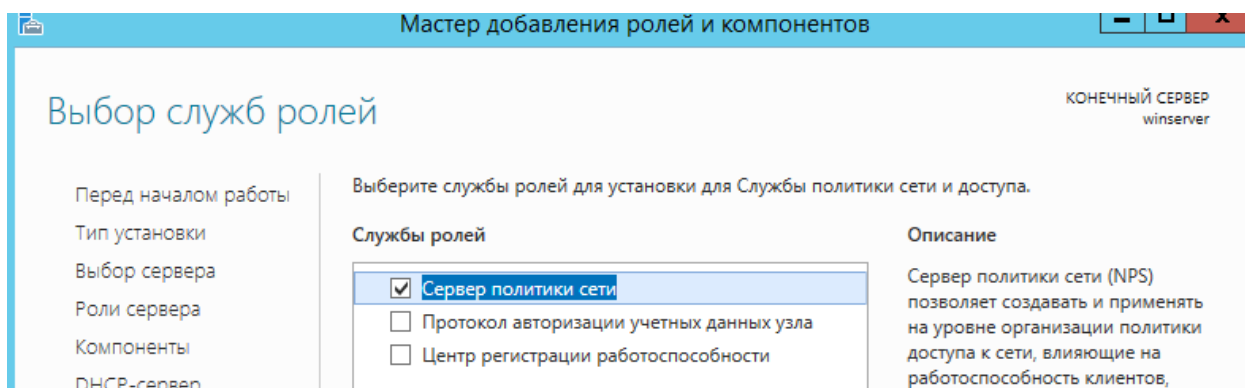


Рис. 12. Выбор служб ролей

Далее выбираем, как показано на рис. 13.

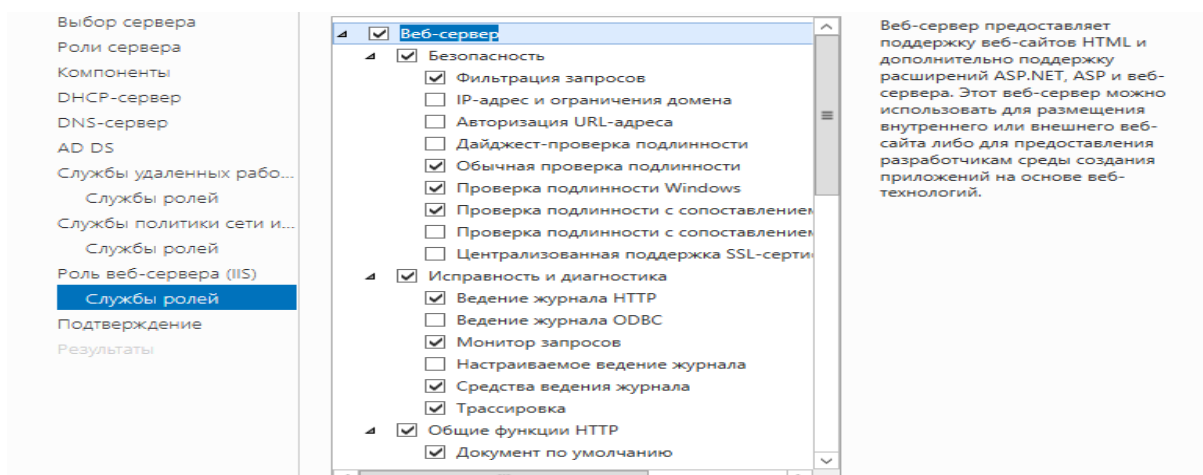


Рис. 13. Выбор служб ролей

Устанавливаем необходимые компоненты. После окончания установки необходимых нам служб перезагружаем систему.

9. Начинаем настройку сервера. Первое что необходимо сделать – это настроить DNS. Производим повышение роли сервера до уровня контроллера домена. В конфигурации развертывания выбираем **Добавить новый лес** и указываем имя корневого домена.

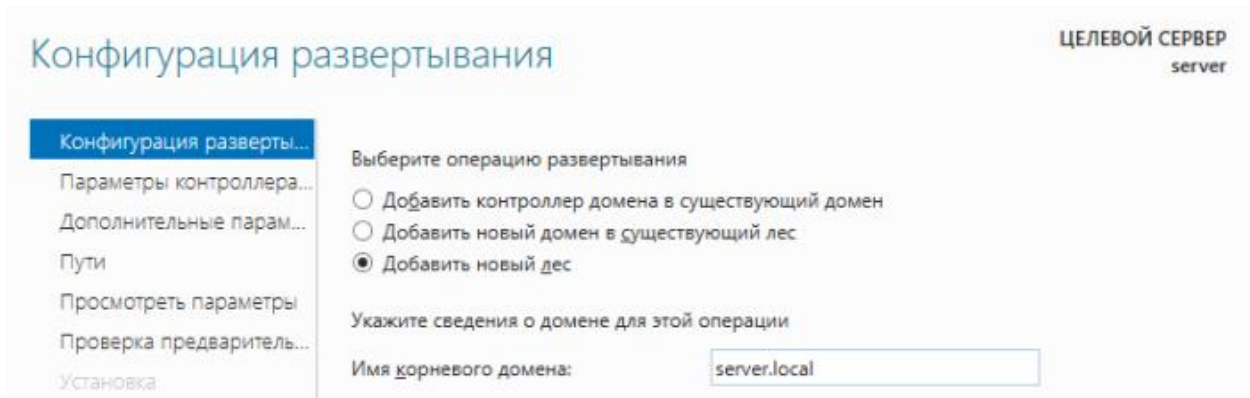


Рис. 14. Конфигурация развертывания

Придумываем пароль для домена.

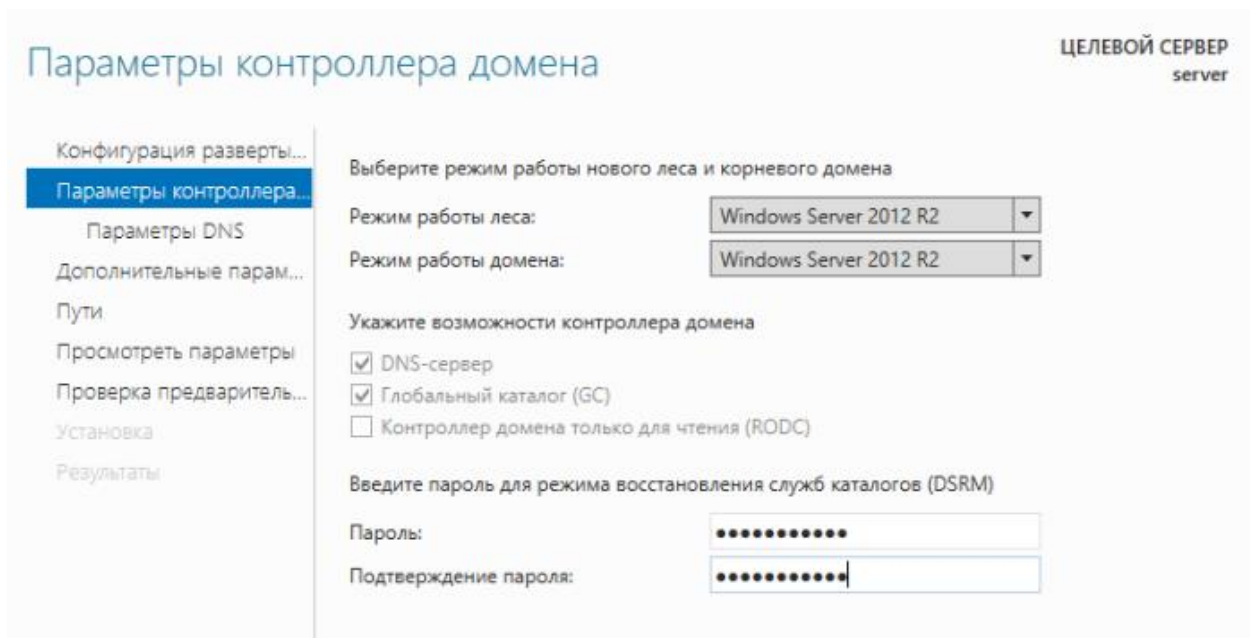


Рис. 15. Параметры контроллера

Производится проверка. Если все было сделано правильно, то установка должна проводиться корректно. После окончания установки необходимо произвести перезагрузку сервера. Производится проверка требований.

Переходим во вкладку DNS. Производится выбор DNS.

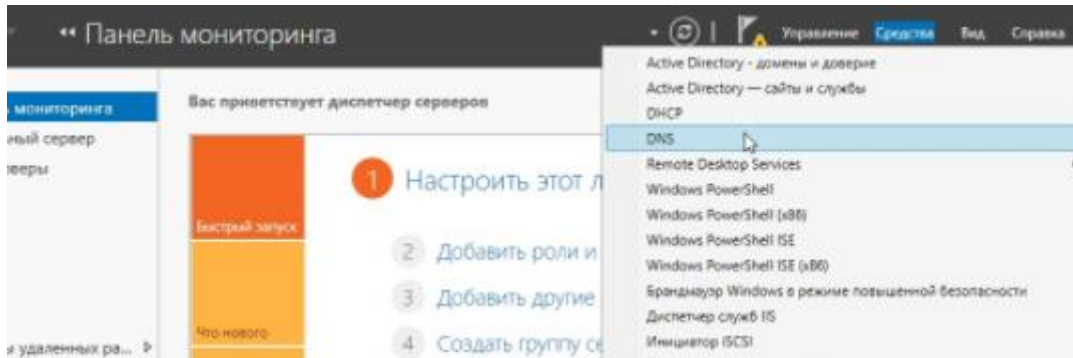


Рис. 16. Выбор DNS

10. Создаем на сервере новую зону. Выбор типа зоны (выбирается основная зона).

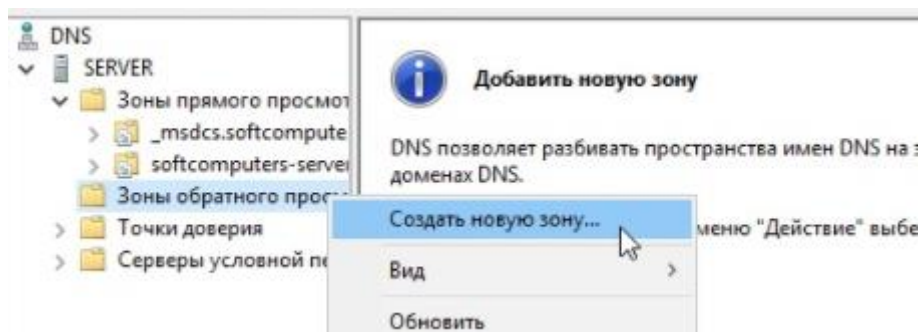


Рис. 17. Создание новой зоны

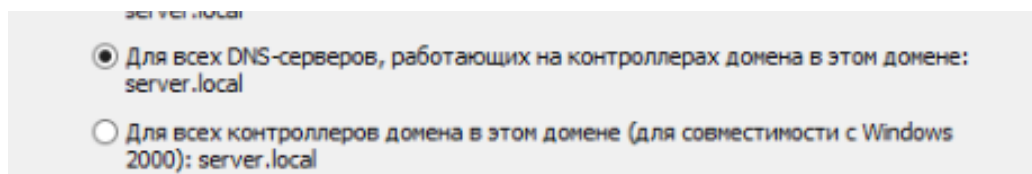


Рис. 18. Выбор области репликации

Производится выбор зоны обратного просмотра (IPv4). Выбираем диапазон IP, который будет принадлежать данной зоне. Определяем идентификатор сети.

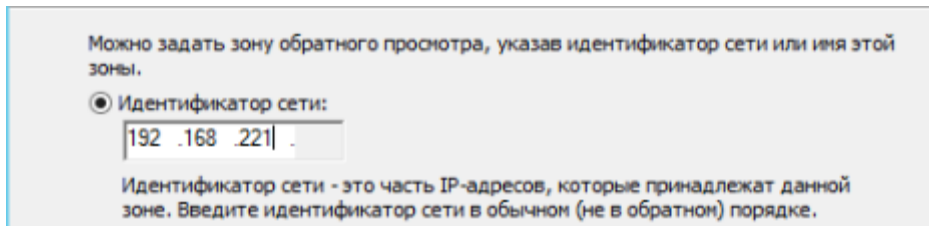


Рис. 19. Идентификатор сети

Указываем **Разрешить динамические обновления**.

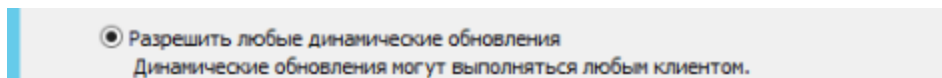


Рис. 20. Выбор типа динамического обновления

Настройка DNS завершена. Теперь переходим к DHCP. Производится настройка DHCP.

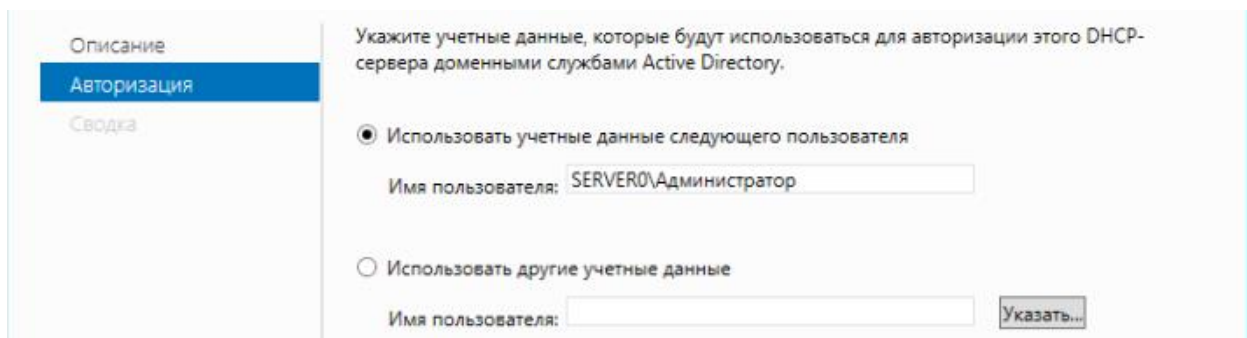


Рис. 21. Настройка DHCP

11. После того как DHCP было создано, переходим к нему для следующей настройки.

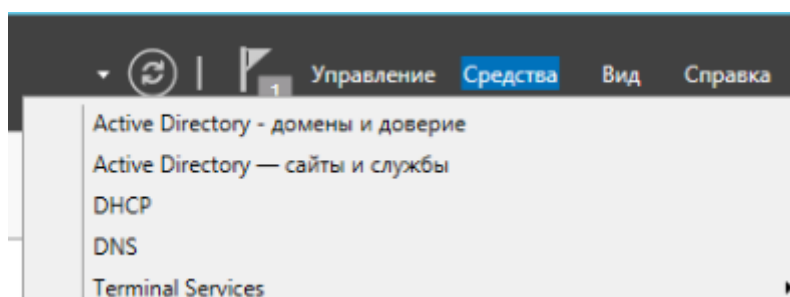


Рис. 22. Настройка DHCP

Выбираем на нашем сервере IPv4 и создаем в ней новую область.

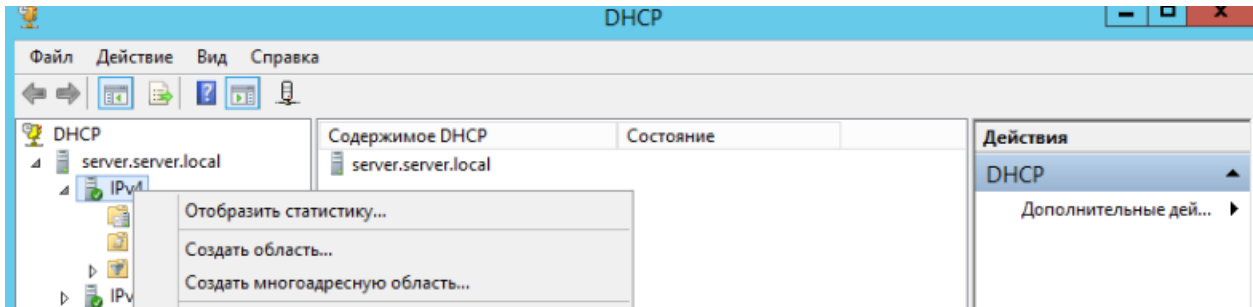


Рис. 23. Создание области

Указывается имя области.

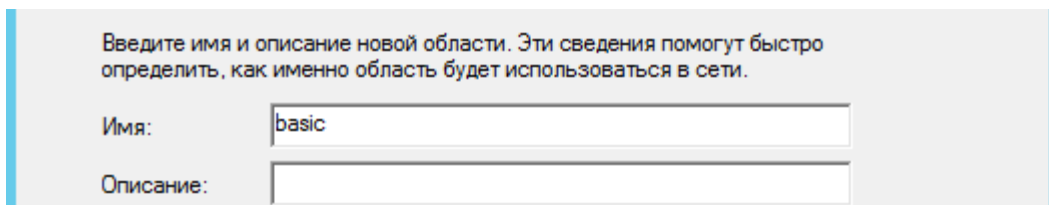


Рис. 23. Имя области

Указываем диапазон IP-адресов, которые будут использованы сервером для создания локальной сети.

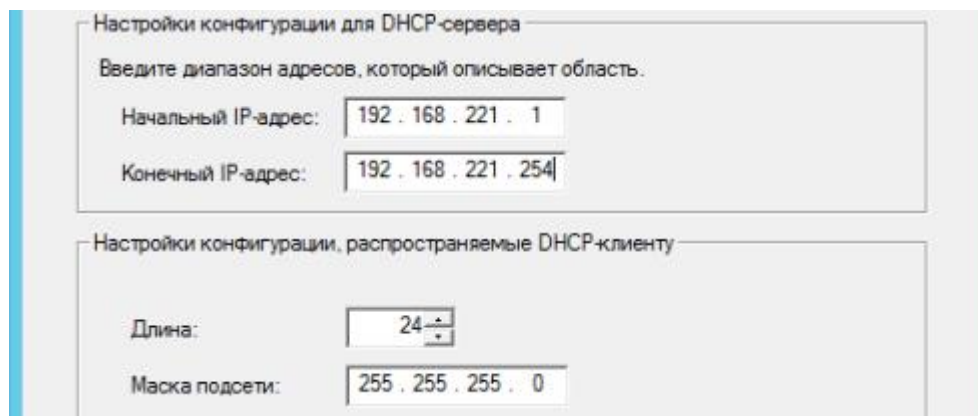


Рис. 24. Диапазон адресов

Производится настройка параметров.

Указываем IP-адрес основного шлюза. Указываем имя домена.

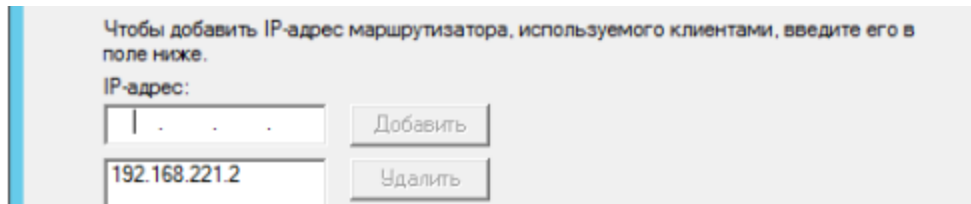


Рис. 25. IP-адрес основного шлюза

Далее настраиваем с учетом наборов адресов используемой сети.

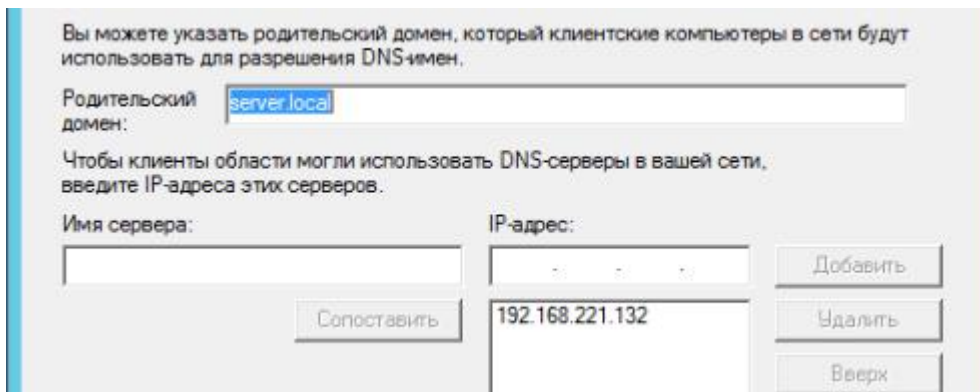


Рис. 26. Имя домена

Как видим, был создан новый диапазон адресов.

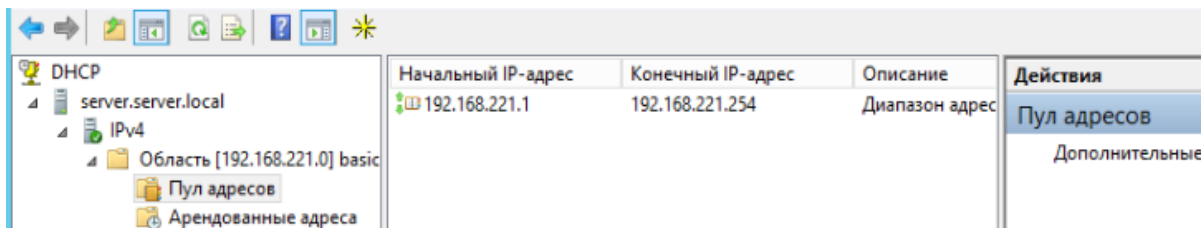


Рис. 27. Диапазон адресов

12. Переходим к созданию пользователей. Создаем подразделение на сервере, где будут содержаться наши пользователи.

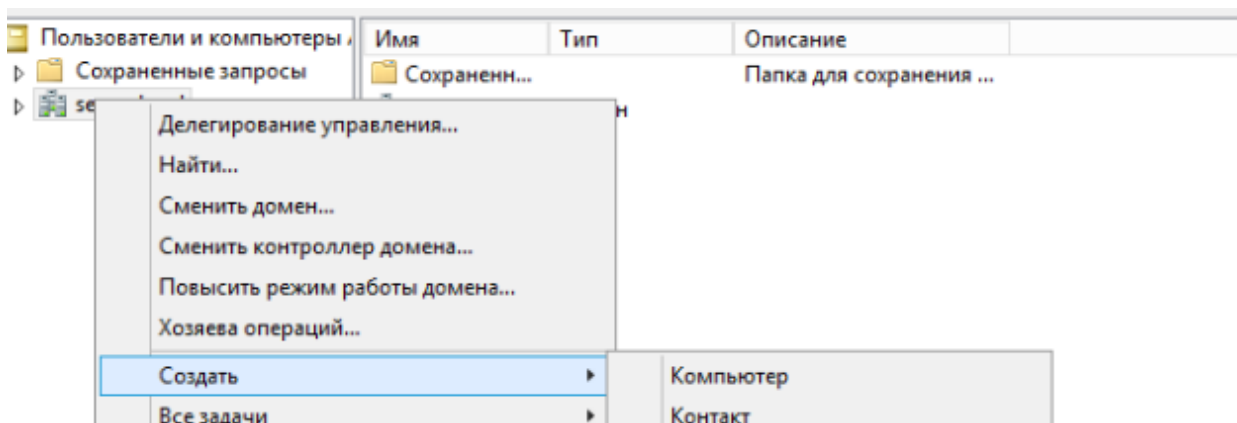


Рис. 28. Создание подразделения

Теперь в созданном подразделении создаем пользователя. Производится настройка пользователя.

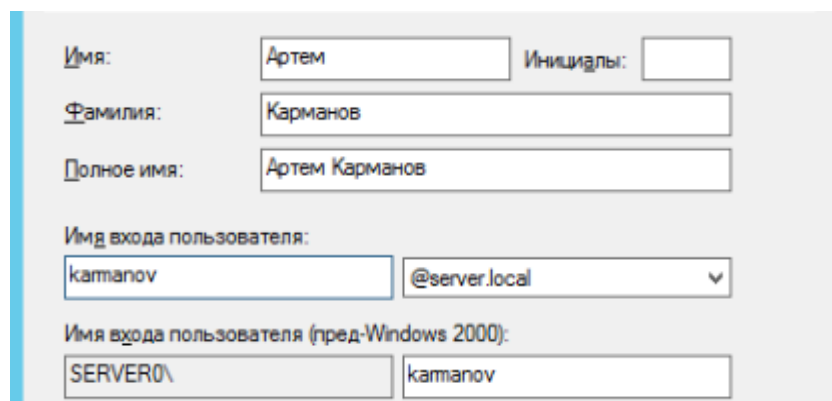


Рис. 29. Настройка пользователя

На этом настройка сервера завершена, теперь необходимо подключить к серверу рабочие станции. Для начала на ОС производим настройку IP-адреса, шлюза и DNS сервера.

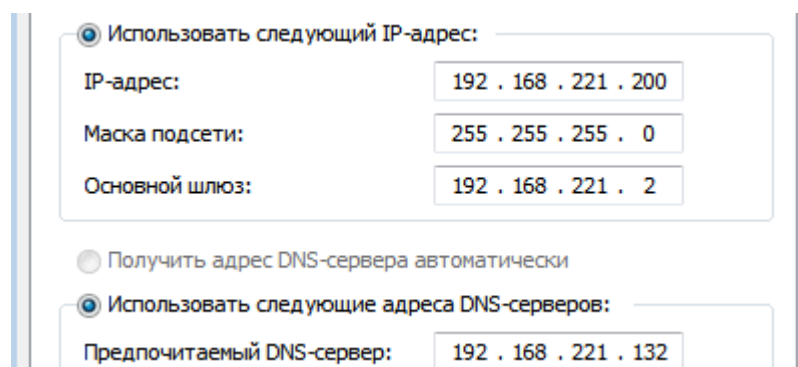


Рис. 30. Настройка IP

13. Производим проверку. Указываем домен нашего сервера.

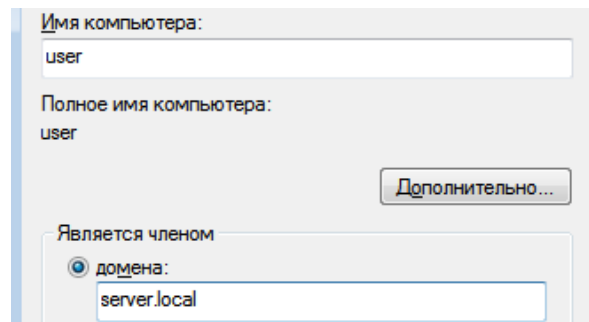


Рис. 31. Указание домена

14. Вводим имя и пароль ранее созданного пользователя. Теперь имеется полностью рабочая инфраструктура. Проведем пен-тест инфраструктуры при помощи Infection Monkey. Запускаем Infection Monkey.

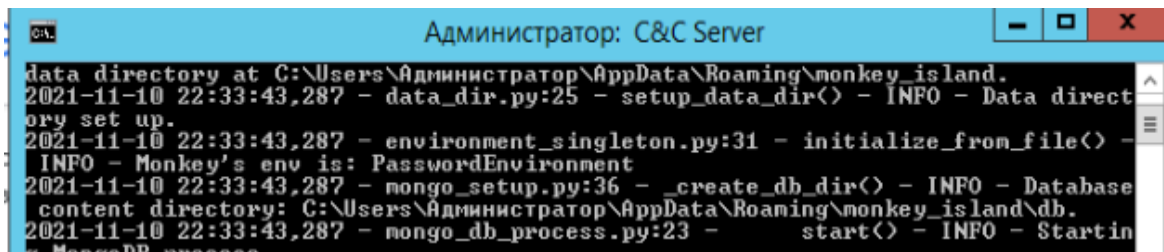


Рис. 32. Консоль Infection Monkey

Затем заходим в браузер и в адресной строке вводим <https://localhost:5000>. После загрузки системы вводим логин и пароль администратора сервера.

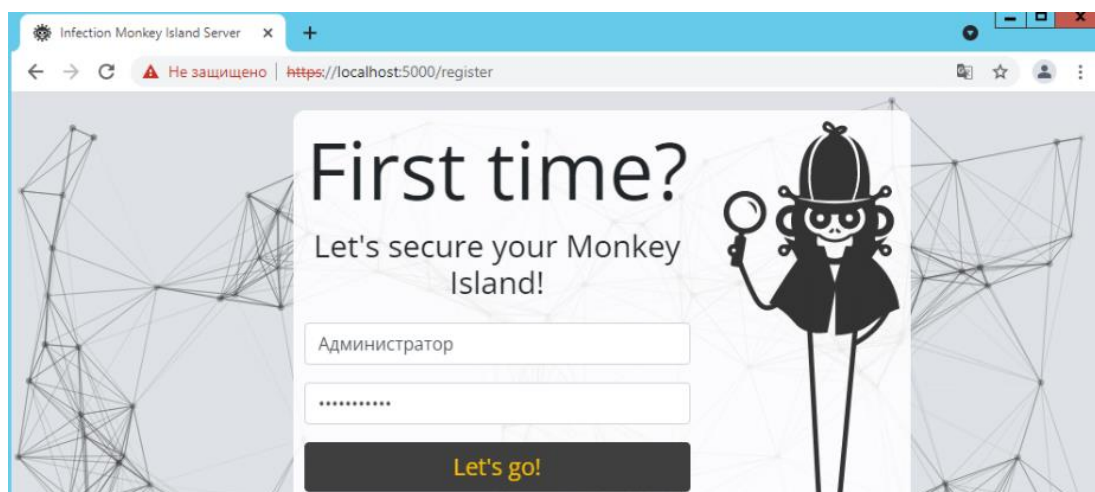


Рис. 33. Терминал Infection Monkey

Выбираем кастомный сценарий.

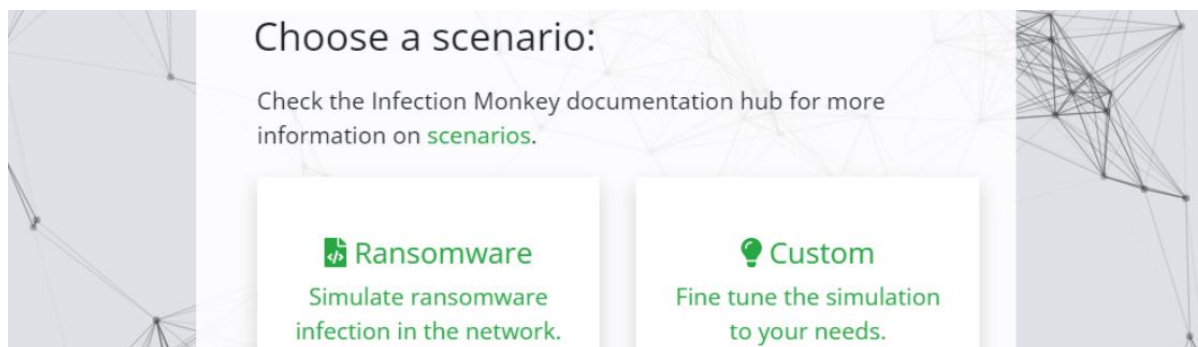


Рис. 34. Выбор сценария

15. Теперь настраиваем конфигурацию атак, которые будут проведены в сети.

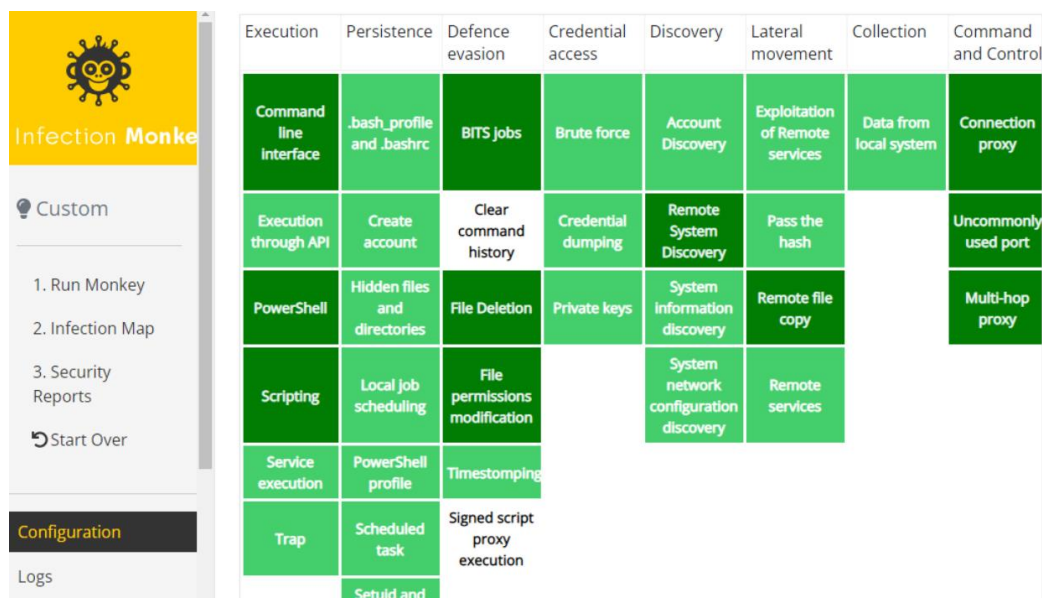


Рис. 35. Выбор конфигурации

После всех настроек запускаем Monkey. Для этого переходим в Run Monkey, где выбираем From Island.

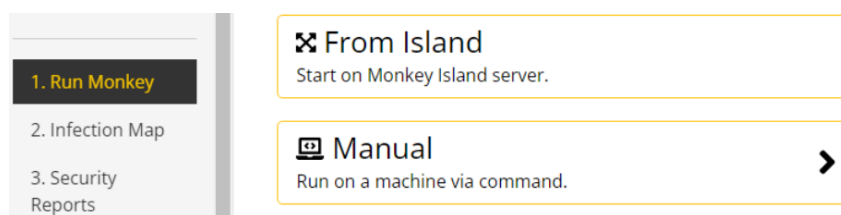
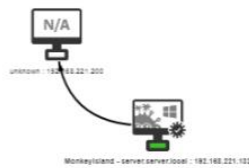


Рис. 36. Запуск Infection Monkey

После окончания проверки будет получен отчет о состоянии защищенности сети:

2. Infection Map

Legend: Exploit █ | Scan █ | Tunnel █ | Island Communication █



```
post breach action executed on server (192.168.221.132) machine. [8/12]
10/11/2021 22:39:36 server.server.local: Monkey discovered machine
192.168.221.200.
10/11/2021 22:40:06 server.server.local: Monkey failed exploiting
192.168.221.200 using the SSHExploiter exploiter.
```

Monkey Telemetry● Kill All Monkeys

MonkeyIsland - server.server.local : 192.168.221.132

Operating System	Windows
Status	Alive
IP Addresses	192.168.221.132
Services	
Accessible From	?
Force Kill	<input type="checkbox"/>
Download Log	Download

Рис. 37. Карта сканирования

Посмотрите на результаты распознавания Brute-force атаки.

The monkey started propagating from the following machines where it was manually installed:

- server.server.local

The monkeys were run with the following configuration:

Username used for brute-forcing:

- Administrator
- root
- user

Passwords used for brute-forcing:

- roo*****
- 123*****
- pas*****

Рис. 37 (а). Brute-force атака

Отчет об успешности реализации Zero Trust.

Summary

Get a quick glance at how your network aligns with the Zero Trust eXtended (ZTX) framework.

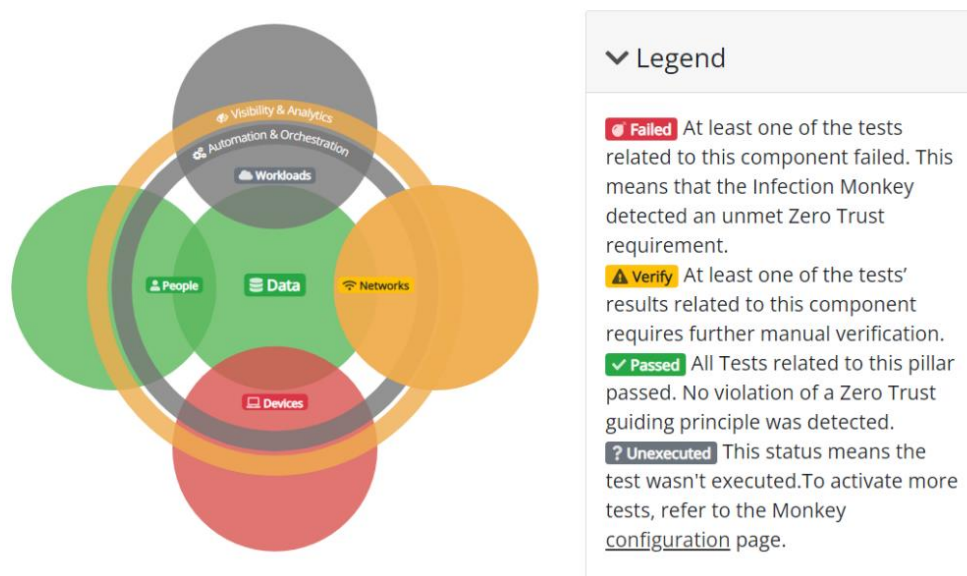


Рис. 38. Отчет об успешности реализации Zero Trust

Далее отчет о проблемах безопасности и способах их устранения. Также просмотрите отчеты «Data» и отчет «Devices».

2.2. Использование PowerShell для аудита ОС Windows

PowerShell – это оболочка командной строки с языком сценариев в одной системе. Сейчас же эта оболочка поддерживается на трех операционных системах: Windows, Linux и MacOS. Изначально она была создана на основе платформы .NET Framework, а позднее на .NET Core. В отличие от принимающих и возвращающих текстовые данные оболочек, Windows PowerShell работает с классами .NET, у которых есть свойства и методы.

Актуальность использования PowerShell связана с тем, что компания Microsoft в настоящее время позиционирует оболочку как основной элемент управления ОС и разработанными ею приложениями.

Рассмотрим с использованием этой технологии аудит ОС Windows. Административные задачи обычно выполняются при помощи командлетов, которые являются специализированными классами .NET. Также можно использовать различные хранилища данных, такие как файловая система и реестр. Windows PowerShell позволяет системным

администраторам автоматизировать большие и сложные задачи. С помощью него можно менять настройки, запускать и останавливать сервисы и службы, а также производить обслуживание некоторых приложений, устанавливать программное обеспечение, управлять процессами, устанавливать роли, компоненты и другие задачи системных администраторов.

PowerShell позволяет выполнять обычные команды, а также дает доступ к объектам COM, WMI и ADSI. Оболочка Windows PowerShell – это среда выполнения команд и сценариев на языке PowerShell. Также многие приложения для Windows предоставляют доступ к своим интерфейсам управления через PowerShell. Язык PowerShell – это объектно-ориентированный скриптовый язык программирования. Он является полноценным скриптовым языком программирования, так как имеет программные конструкции, которые присутствуют в каждом языке программирования.

Windows PowerShell позволяет:

1. Менять настройки ОС.
2. Управлять службами и процессами.
3. Настраивать роли и компоненты сервера.
4. Устанавливать программное обеспечение.
5. Управлять установленным ПО через специальные интерфейсы.
6. Встраивать исполняемые компоненты в сторонние программы.
7. Создавать сценарии для автоматизации задач администрирования.
8. Работать с файловой системой, реестром Windows, хранилищем сертификатов и т. д.

Данная оболочка имеет те же возможности что и командная строка такие как:

1. Хранение истории выполнения команд.
2. Настройка внешнего вида оболочки.
3. Завершение выполнения команд сочетанием клавиш Ctrl+C.
4. Много других возможностей, которых нет в оболочке командной строки.

Запустить оболочку PowerShell можно несколькими способами, например:

1. Из командной строки, набрав PowerShell.
2. Через диалоговое окно «Выполнить» (сочетание клавиш Win+R), также набрав PowerShell.
3. В Windows 7 – Пуск – Все программы – Стандартные – Windows PowerShell – Windows PowerShell.

4. В Windows 8.1 или Windows Server 2012 R2 – Пуск – Все программы – Служебные – Windows PowerShell.

5. В Windows 10 или Windows Server 2016 – Пуск – Все программы – Каталог Windows PowerShell (в группе W) – Windows PowerShell.

Командлет – это команда Windows PowerShell, при помощи которой можно осуществлять взаимодействие с объектами ОС с целью их управления. Они являются частью языка PowerShell и построены по принципу «глагол-существительное», сначала указываем что делать, а через дефис на чем.

Язык, который используется в PowerShell, создан для администрирования задач, но он также является полноценным скриптовым языком программирования, так как имеет программные конструкции, которые присутствуют в каждом языке программирования.

Попробуем решить такую задачу как аудит персонального компьютера с помощью PowerShell, для этого воспользуемся программой, которая есть на любом персональном компьютере Windows PowerShell. В решении данной задачи поможет Защитник Windows и возможность управлять им с помощью PowerShell. Главным преимуществом защитника является простота использования, наличие по стандарту в Windows, работа по умолчанию.

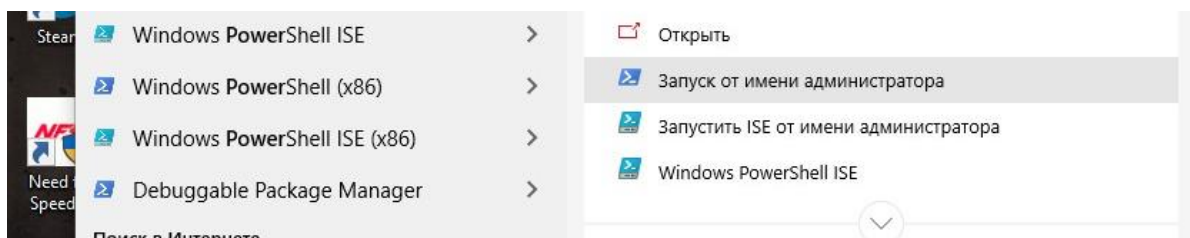


Рис. 39. Запуск Windows PowerShell

Командлеты PowerShell возвращают результаты в виде объектов, что является отличием от командной строки Windows, там команды возвращают на экран только текст.

Список командлетов

№	Командлет	Описание
1	Надстройка MpPreference	Настройка защитника Windows
2	Get- MpComputerStatus	Состояние антивирусного программного обеспечения на компьютере
3	Get-MpPreference	Получает настройки для сканирования и обновлений Защитника Windows
4	Get-MpThreat	Получает историю угроз, обнаруженных на компьютер
5	Get- MpThreatCatalog	Получает известные угрозы из каталога определений
6	Get-MpThreatDe- tection	Получает активные и прошлые вредоносные угрозы, обнаруженные Защитником Windows
7	Set-MpPreference	Удаляет исключения или действия по умолчанию
8	Запуск MpScan	Запускает сканирование на компьютере
9	Запуск MpWDOS- scan	Запуск автономного сканирования Защитника Windows
10	Запуск MpWDOS- scan	Обновляет определения анти- вредоносных программ на компьютере

Полный список командлетов можно получить с помощью команды Get-Command.

```
PS C:\WINDOWS\system32> Get-Command

CommandType      Name                                     Version      Source
-----
Alias             Add-AppPackage                         2.0.1.0     Appx
Alias             Add-AppPackageVolume                  2.0.1.0     Appx
Alias             Add-AppProvisionedPackage             3.0         Dism
```

Рис. 40. Выполнение команды Get-Command

Также у командлетов есть параметры, которые можем использовать для уточнения действия командлета. Например, параметр -Verb ищет по глаголу, а -Noun по существительному. Тем самым при помощи параметра можем сократить радиус поиска нужной нам командлеты.

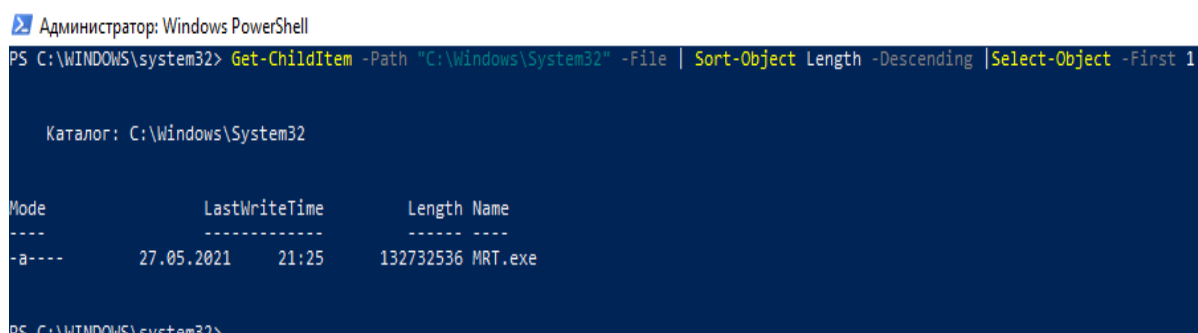
Командлеты имеют несколько ключевых особенностей:

1. Существуют системные, пользовательские и опциональные командлеты.
2. Результатом выполнения командлета будет объект или массив объектов.
3. Командлеты могут обрабатывать данные и передавать их другим командлетам с помощью конвейеров.
4. Командлеты нечувствительны к регистру, так что нет никакой разницы между `Get-ADUser`, `get-aduser` и `gEt-AdUsEr`.
5. В качестве разделителя используется символ.

Конвейер – это передача работы командлета другому командлету, через знак «|».

Конвейеры помогают выполнять сложные задачи простым и удобным способом без написания сложных алгоритмов и сценариев.

Рассмотрим пример использования конвейера:



```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Get-ChildItem -Path "C:\Windows\System32" -File | Sort-Object Length -Descending | Select-Object -First 1

Каталог: C:\Windows\System32

Mode                LastWriteTime         Length Name
----                -
-a----           27.05.2021   21:25      132732536 MRT.exe

PS C:\WINDOWS\system32>
```

Рис. 41. Пример использования конвейера

Команда «`Get-ChildItem -Path "C:\Windows\System32" -File | Sort-Object Length -Descending | Select-Object -First 1`», вывела название самого большого файла в выбранном каталоге. Используются следующие командлеты:

1. `Get-ChildItem` – командлет получения объектов в указанном каталоге.
2. `Sort-Object` – командлет для сортировки объектов, в данном примере сортировка по размеру файла.
3. `Select-Object` – командлет выбора нужных свойств объекта, в примере выводится самый большой файл.

Все командлеты отделены друг от друга знаком «|». Результат работы передается от одного командлета к другому, сначала получаем объекты в указанном каталоге, затем сортируем полученный результат и в заключении выбираем первый объект.

Выполнить

1. Получение сведений о компьютере и его ОС. Для начала определим версию операционной системы, с помощью команды "(Get-WmiObject Win32_OperatingSystem).Version", где:

А. Get-WmiObject – это важный командлет для решения общих задач управления системой, который работает через службу WMI.

В. Win32_OperatingSystem – класс, который представляет ОС Windows, установленную на компьютере.

С. Version – указывает на то, какие данные нам нужно получить. Введите команду на приведенном ниже рис. 42.

```
PS C:\WINDOWS\system32> (Get-WmiObject Win32_OperatingSystem).Version
10.0.19041
```

Рис. 42. Версия операционной системы

2. Из рис. 43 видим, что версия операционной системы 10.0.19041. Теперь узнаем название операционной системы, имя компьютера и разрядность системы.

```
PS C:\> (Get-WMIObject win32_operatingsystem).name
Майкрософт Windows 10 Домашняя для одного языка|C:\WINDOWS|\Device\Harddisk0\Partition3
PS C:\> (Get-WMIObject win32_operatingsystem).CSName
DESKTOP-Q14A4DP
PS C:\> (Get-WMIObject win32_operatingsystem).OSArchitecture
64-разрядная
```

Рис. 43. Параметры ОС

3. Свойства класса Win32_OperatingSystem включают сведения о версии ОС и пакета обновления. Эти свойства можно выбрать, используя конвейеры. Их можно получить, как в виде таблицы, так и в виде столбца. Введите команду на приведенном ниже рис. 44.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_OperatingSystem | Select-Object -Property BuildNumber, BuildType, OSType, ServicePackMinorVersion

BuildNumber BuildType      OSType ServicePackMinorVersion
-----
19041      Multiprocessor Free      18           0

PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_OperatingSystem | Select-Object -Property Build*, OSType, ServicePack*

BuildNumber      : 19041
BuildType        : Multiprocessor Free
OSType           : 18
ServicePackMajorVersion : 0
ServicePackMinorVersion : 0
```

Рис. 44. Свойства ОС

Из рис. 44 видим, что номер ОС такой же, как и в прошлой проверке, тип сборки процессора – многопроцессорная. А также OSType 18, номер основной версии последнего пакета обновления "ServicePackajorVersion", установленного на сервере 0, а также номер младшей версии последнего пакета обновления "ServicePackajorVersion" тоже 0.

Получим сведения о нашем компьютере. Введите команду на приведенном ниже рис. 45.

```
PS C:\WINDOWS\system32> Get-WmiObject -Class Win32_ComputerSystem | Select-Object UserName

UserName
-----
DESKTOP-Q14A4DP\1

PS C:\WINDOWS\system32> $userinfo = Get-WmiObject -ComputerName 'localhost' -Class Win32_ComputerSystem
>> $user = $userinfo.UserName -split '\\'
>> $user[1]
```

Рис. 45. Имя компьютера и пользователя ОС

Выведем информацию о компьютере.

```
PS C:\WINDOWS\system32> (Get-WmiObject Win32_ComputerSystem)

Domain           : WORKGROUP
Manufacturer     : ASUSTek COMPUTER INC.
Model            : VivoBook 15_ASUS Laptop X540BA
Name             : DESKTOP-Q14A4DP
PrimaryOwnerName : Пользователь Windows
TotalPhysicalMemory : 4160872448
```

Рис. 46. Информация о компьютере

Можем увидеть следующие параметры компьютера:

- а) Имя домена WORKGROUP;
- б) Производитель компьютера: "ASUSTek COMPUTER INC.";
- в) Модель: "VivoBook 15_ASUS Laptop X540BA";
- г) Имя компьютера: "DESKTOP-Q14A4DP";
- д) Имя основного владельца: Пользователь Windows;
- е) Общая физическая память равна 41608722448.

4. Далее посмотрим оценку производительности компьютера командлетом Get-CimInstance. Введите команду на приведенном ниже рис. 47.

```
PS C:\> Get-CimInstance Win32_WinSAT

CPUScore           : 8,2
D3DScore           : 9,9
DiskScore          : 5,9
GraphicsScore      : 5
MemoryScore        : 5,9
```

Рис. 47. Оценка производительности компьютера

Данный командлет вывел следующие оценки производительности:

- а) CPU Score – оценка процессора – 8,2;
- б) Disk Score – оценка для жесткого диска – 5,9;
- в) Graphics Score – оценка видеокарты – 5;
- г) MemoryScore – оценка оперативной памяти – 5,9;
- д) SPRLevel – общая оценка 5.

Теперь посмотрим список установленных исправлений с помощью класса Win32_QuickFixEngineering.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_QuickFixEngineering

Source      Description      HotFixID      InstalledBy      InstalledOn
-----
Update      Update           KB4601554     NT AUTHORITY\СИСТЕМА 19.04.2021 0:00:00
Security Update Security Update KB4577266     NT AUTHORITY\СИСТЕМА 04.12.2020 0:00:00
Update      Update           KB4577586     NT AUTHORITY\СИСТЕМА 03.04.2021 0:00:00
Security Update Security Update KB4580325     NT AUTHORITY\СИСТЕМА 10.12.2020 0:00:00
Security Update Security Update KB4586864     NT AUTHORITY\СИСТЕМА 10.12.2020 0:00:00
Update      Update           KB4589212     NT AUTHORITY\СИСТЕМА 19.03.2021 0:00:00
Security Update Security Update KB4593175     NT AUTHORITY\СИСТЕМА 14.12.2020 0:00:00
Security Update Security Update KB4598481     NT AUTHORITY\СИСТЕМА 22.01.2021 0:00:00
Security Update Security Update KB5003173     NT AUTHORITY\СИСТЕМА 28.05.2021 0:00:00
Security Update Security Update KB5003242     NT AUTHORITY\СИСТЕМА 27.05.2021 0:00:00
```

Рис. 48. Установленные исправления

Также в PowerShell можно узнать параметры рабочего стола, вывести сведения о BIOS, о процессоре и локальных пользователей компьютера.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_Desktop

SettingID Name      ScreenSaverActive ScreenSaverSecure ScreenSaverTimeout
-----
NT AUTHORITY\СИСТЕМА False
NT AUTHORITY\LOCAL SERVICE False
NT AUTHORITY\NETWORK SERVICE False
DESKTOP-Q14A4DP\1 False
.DEFAULT False
```

Рис. 49. Сведения о рабочих столах локального компьютера

Также можно вывести данные BIOS компьютера.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_BIOS

SMBIOSBIOSVersion : X540BA.303
Manufacturer       : American Megatrends Inc.
Name               : X540BA.303
SerialNumber       : JBN0GR044119466
Version            : _ASUS_ - 1072009
```

Рис. 50. Сведения о BIOS

Класс WMI Win32_BIOS возвращает довольно полные и краткие сведения о системе BIOS локального компьютера. BIOS – это базовая система ввода – вывода.

Общие сведения о процессоре можно получить, используя класс Win32_Processor.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_Processor | Select-Object -ExcludeProperty "CIM*"

DeviceID Name                               Caption                               MaxClockSpeed SocketDesignation Manufacturer
-----
CPU0      AMD A6-9225 RADEON R4, 5 COMPUTE CORES 2C+3G AMD64 Family 21 Model 112 Stepping 0 2600      P0          AuthenticAMD
```

Рис. 51. Сведения о процессоре

Также можно вывести сведения локальных пользователей и владельцев.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_OperatingSystem | Select-Object -Property *user*

NumberOfLicensedUsers NumberOfUsers RegisteredUser
-----
0                      2 Пользователь Windows
```

Рис. 52. Сведения локальных пользователей и владельцев

На этом компьютере локальный пользователь один. Используемые выше команды позволяют быстро получить сведения об определенных параметрах и сведениях системы и компьютера. Но если требуется получить все сведения о системе, то можно воспользоваться утилитой командной строки \$systeminfo, которая выдаст подробную информацию о системе, включая установленные обновления.

Общие сведения о сеансах входа в систему, связанных с пользователями, можно узнать через класс Win32_LogonSession.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_LogonSession
```

LogonId	Name	LogonType	StartTime	Status	AuthenticationPackage
999		0	28.05.2021 19:42:07		NTLM
997		5	28.05.2021 19:42:13		Negotiate
996		5	28.05.2021 19:42:10		Negotiate
1791746		2	28.05.2021 19:53:11		NTLM
1791210		2	28.05.2021 19:53:11		NTLM
98547		2	28.05.2021 19:42:11		Negotiate
98504		2	28.05.2021 19:42:11		Negotiate
71107		2	28.05.2021 19:42:07		Negotiate
71139		2	28.05.2021 19:42:07		Negotiate

Рис. 53. Сведения о сеансе входа в систему

Имя пользователя, выполнившего вход в компьютер, можно определить при помощи класса Win32_ComputerSystem. Эта команда возвращает только пользователей, выполнивших вход на рабочий стол системы.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_ComputerSystem -Property UserName
```

```
AdminPasswordStatus      :
BootupState              :
ChassisBootupState      :
KeyboardPasswordStatus   :
PowerOnPasswordStatus    :
PowerSupplyState         :
PowerState               :
FrontPanelResetStatus    :
ThermalState             :
Status                   :
Name                     : DESKTOP-Q14A4DP
PowerManagementCapabilities :
PowerManagementSupported :
Caption                  :
Description              :
InstallDate              :
CreationClassName        :
NameFormat               :
PrimaryOwnerContact      :
PrimaryOwnerName         :
Roles                    :
InitialLoadInfo          :
```

Рис. 54. Сведения о пользователе, выполнившем вход в систему

Получить информацию о дисках и разделах можно с помощью следующих командлетов:

1. `Get-PhysicalDisk` позволяет получать характеристики устройства.
 2. `Get-Disk` выводит сведения о дисках на логическом уровне.
 3. `Get-Partition` отображает информацию о разделах на всех дисках.
- Введите команду на приведенном ниже рис. 55.

```
PS C:\WINDOWS\system32> Get-PhysicalDisk
Доступно предотвращение выполнения данных: Да
Number FriendlyName      SerialNumber MediaType CanPool OperationalStatus HealthStatus Usage      Size
-----
0      TOSHIBA MQ01ABF050 58F2PFMRT   HDD      False   OK              Healthy   Auto-Select 465.76 GB

PS C:\WINDOWS\system32> Get-Disk
Number Friendly Name                               Serial Number
-----
0      TOSHIBA MQ01ABF050                               58F2PFMRT

PS C:\WINDOWS\system32> Get-Partition
```

Рис. 55. Сведения о дисках и разделах

Из рис. 14 видим:

1. Тип накопителя – HDD.
2. Общий размер диска – 465,76 GB.
3. Модель диска "TOSHIBA MQ01ABF050".
4. Название диска "C".
5. Тип диска – Basic, т. е. основной.
6. Серийный номер – 58F2PFMRT.
7. Статус здоровья – здоров.
8. Диск не доступен для добавления в пул хранения.
9. Используется автоматический выбор использования.
10. Статус разделения – GPT – таблица разделов GUID, созданная для замены MBR, и является частью UEFI, который пришел на замену BIOS.
11. Рабочее состояние онлайн.

Еще при помощи PowerShell можно посмотреть свойства диска, список дисков в системе, их модель, тип и размер, место на диске, а также логические диски на компьютере. Рассмотрим эти параметры далее.

```

Администратор: Windows PowerShell

PS C:\WINDOWS\system32> Get-PhysicalDisk | fl *

ClassName           : MSFT_PhysicalDisk
Usage               : Auto-Select
OperationalStatus   : OK
UniqueIdFormat      : FCPH Name
HealthStatus        : Healthy
BusType             : SATA
CannotPoolReason    : Insufficient Capacity
SupportedUsages     : {Auto-Select, Manual-Select, Hot Spare, Retired...}
MediaType           : HDD
SpindleSpeed        : Unknown
ObjectId            : {1}\{DESKTOP-Q14A4DP\root\Microsoft\Windows\Storage\Providers_v2\SPACES_PhysicalDisk
                    -3745-ffe48f7e68ac}
  
```

Рис. 56. Свойства дисков

Для большего удобства свойства можно посмотреть в виде таблицы.

```

PS C:\WINDOWS\system32> Get-PhysicalDisk | ft -AutoSize

Number FriendlyName      SerialNumber MediaType CanPool OperationalStatus HealthStatus Usage      Size
-----
0       TOSHIBA MQ01ABF050 58F2PFMRT   HDD       False    OK                Healthy    Auto-Select 465.76 GB
  
```

Рис. 57. Свойства

Выполнить

Введите команду на приведенном ниже рис. 58.

```

PS C:\> Get-PSDrive

Name          Used (GB)  Free (GB) Provider      Root
-----
Alias
C              191,67    272,52  FileSystem    C:\
Cert          12,62     1,83   Certificate   \
D              12,62     1,83   FileSystem    D:\
Env
Function
HKCU          Registry   HKEY_CURRENT_USER
HKLM          Registry   HKEY_LOCAL_MACHINE
Variable
WSMan
  
```

Рис. 58. Список дисков в системе

Из рис. 59 видно, что на встроенном диске "C", занято 191,67 GB, а свободно 272,52.

```

PS C:\WINDOWS\system32> Get-PhysicalDisk | sort DeviceId | ft -AutoSize DeviceId,Model,MediaType,BusType,Size

DeviceId Model          MediaType BusType      Size
-----
0       TOSHIBA MQ01ABF050 HDD       SATA         500107862016
  
```

Рис. 59. Модель, тип и размер диска

Тип кабеля в компьютере – SATA – это текущий стандарт интерфейса для жестких дисков и других устройств хранения данных в компьютере.

```
PS C:\WINDOWS\system32> Get-CimInstance -ClassName Win32_LogicalDisk -Filter "DriveType=3"

DeviceID DriveType ProviderName VolumeName Size FreeSpace
-----
C:        3              498416513024 341696778240
```

Рис. 60. Место на диске

Выполнить

Введите команду на приведенном ниже рис. 61.

```
PS C:\WINDOWS\system32> Get-WmiObject -Class Win32_LogicalDisk

DeviceID      : C:
DriveType     : 3
ProviderName  :
FreeSpace    : 292624957440
Size         : 498416513024
VolumeName    :

DeviceID      : D:
DriveType     : 2
ProviderName  :
FreeSpace    : 1960296448
Size         : 15511601152
VolumeName    : USB DISK
```

Рис. 61. Логические диски на компьютере

Выяснилось, что на компьютере один логический диск "C", его размер. Узнали его модель, тип, его рабочее состояние и другие параметры, указанные выше.

Просмотр процессов, запущенных на ПК. В Windows PowerShell при помощи командлеты Get-Process, можно узнать все процессы, запущенные на компьютере.

```
Администратор: Windows PowerShell
PS C:\> Get-Process

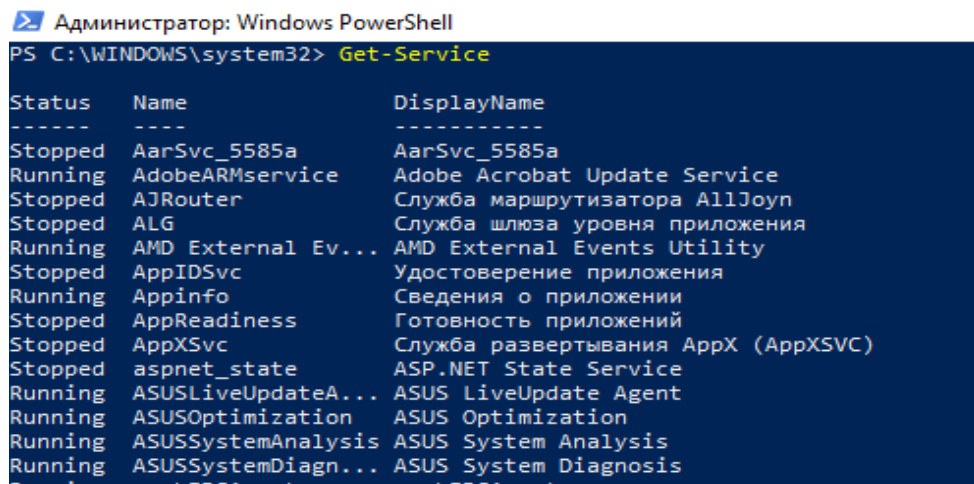
Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName
-----
152     12   2300   1152   0,17  4580 1 amdow
3683    37  164472  5076   3,56  1520 1 AMDRSServ
254     16   4972   2172   0,45  9616 1 ApplicationFrameHost
333     25   4140   4252   1,38  4516 0 ApplicationWebServer
132     9    1220   5900   0,05  2592 0 armsvc
306     12   6104   8024   0,61  4016 0 AsusLiveUpdateAgent
94      5     940   2104   0,03  4116 0 ASUSOptimization
354     13   3460   7728   1,52  4132 0 AsusSystemAnalysis
123     7    1172   2472   0,03  4168 0 AsusSystemDiagnosis
```

Рис. 62. Список процессов

Помимо процессов, запущенных на компьютере, можно посмотреть список всех сервисов и служб на компьютере, командой Get-Service. И посмотреть их состояние, запущены они или остановлены.

Выполнить

Введите команду на приведенном ниже рис. 63.



```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Get-Service

Status Name                DisplayName
-----
Stopped AarSvc_5585a         AarSvc_5585a
Running AdobeARMservice     Adobe Acrobat Update Service
Stopped AJRouter         Служба маршрутизатора AllJoyn
Stopped ALG           Служба шлюза уровня приложения
Running AMD External Ev... AMD External Events Utility
Stopped AppIDSvc      Удостоверение приложения
Running Appinfo       Сведения о приложении
Stopped AppReadiness  Готовность приложений
Stopped AppXSvc       Служба развертывания AppX (AppXSVC)
Stopped aspnet_state  ASP.NET State Service
Running ASUSLiveUpdateA... ASUS LiveUpdate Agent
Running ASUSOptimization  ASUS Optimization
Running ASUSSystemAnalysis  ASUS System Analysis
Running ASUSSystemDiagn... ASUS System Diagnosis
```

Рис. 63. Список сервисов и служб

Аудит безопасности Windows. Аудит заносит информации об определенных событиях (источник, код события, успех или отказ и т. д.) в специальные журналы, такие как журнал безопасности. Аудит может содержать сведения о взаимодействии с файлами или папками, или определенное событие, например, вход в систему или выход из нее.

Базовая политика аудита определяет категории событий, связанных с безопасностью, которые необходимо периодически проверять. Включив различные категории событий аудита, можно реализовать политику аудита, отвечающую необходимым требованиям безопасности.

Политики аудита:

1. Audit account logon events.
2. Audit account management.
3. Audit directory service access.
4. Audit logon events.
5. Audit object access.
6. Audit policy change.
7. Audit privilege use.
8. Audit process tracking.
9. Audit system events.

Также PowerShell включает в себя параметры аудита безопасности системы. С помощью командлета Get-EventLog, который получает события и журналы событий от локальных и удаленных компьютеров. По умолчанию этот командлет получает журналы с локального компьютера. Использует этот командлет для просмотра событий журнала безопасности, который называется security. Можно посмотреть весь список журнала, а также можно осуществить просмотр событий только на сегодняшний день.

Выполнить

1. Введите команду на приведенном ниже рис. 64.

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Get-EventLog security
```

Index	Time	EntryType	Source	InstanceID	Message
193585	май 27 17:10	SuccessA...	Microsoft-Windows...	4672	Новому сеансу входа назначены специальные привилегии...
193584	май 27 17:10	SuccessA...	Microsoft-Windows...	4624	Вход в учетную запись выполнен успешно....
193583	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193582	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193581	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193580	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193579	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193578	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193577	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193576	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193575	май 27 17:09	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...

Рис. 64. Список событий журнала Security

1. Введите команду на приведенном ниже рис. 65.

```
Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Get-EventLog security -after (Get-date -hour 0 -minute 0 -second 0)
```

Index	Time	EntryType	Source	InstanceID	Message
193600	май 27 17:14	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193599	май 27 17:14	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193598	май 27 17:14	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...
193597	май 27 17:14	SuccessA...	Microsoft-Windows...	5379	Учетные данные диспетчера учетных данных прочитаны...

Рис. 65. Список событий журнала Security на сегодняшний день.

Также используя конвейер, можно получить свойства выводимых командой Get-EventLog объектов, с помощью команды Get-Member. На выходе получим список свойств всех событий, выводимых Get-EventLog.

2. Введите команду на приведенном ниже рис. 66.

```

Администратор: Windows PowerShell
PS C:\WINDOWS\system32> Get-EventLog security | Get-member

    TypeName: System.Diagnostics.EventLogEntry#security/Microsoft-Windows-Security-Auditing/5379

Name      MemberType Definition
----      -
Disposed  Event      System.EventHandler Disposed(System.Object, System.EventArgs)
CreateObjRef Method     System.Runtime.Remoting.ObjRef CreateObjRef(type requestedType)
Dispose    Method     void Dispose(), void IDisposable.Dispose()

```

Рис. 66. Свойства объектов, выводимых командой `Get-EventLog`

Защитник Windows – это системная антивирусная программа. В последних версиях Windows 10 название изменилось на "Windows Defender". Также он предоставляет пользователям расширенную веб-защиту и защиту в режиме реального времени от опасных вирусов, троянов и других вредоносных программ.

В командной оболочке PowerShell можно легко обновлять защитник Windows, запускать сканирование системы и проверять текущее состояние антивируса, при помощи нескольких команд. Команды в PowerShell ускоряют процесс настройки и дают доступ к опциям, которые не доступны в интерфейсе Защитника Windows напрямую.

```

PS C:\WINDOWS\system32> Get-MpComputerStatus

AMEngineVersion      : 1.1.16900.4
AMProductVersion     : 4.18.2003.8
AMServiceEnabled     : True
AMServiceVersion     : 4.18.2003.8

```

Рис. 67. Состояние антивирусного программного обеспечения

Так как метка `AntivirusEnabled` равна `True`, то можно сделать вывод, что Защитник Windows работает и настроен правильно. Антивирусное программное обеспечение нужно регулярно обновлять, для того чтобы поддерживать актуальность определений. Обновить защитник можно командой `Update-MpSignature`.

Следующая командлета обновит настройки Защитника Windows, чтобы автоматически проверять обновления определений каждый день.

```
PS C:\WINDOWS\system32> Set-MpPreference -SignatureScheduleDay Everyday
```

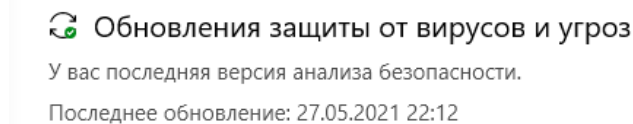


Рис. 68. Обновление настроек защитника Windows

В некоторых случаях требуется провести быструю проверку компьютера на наличие вирусов, хоть это легко можно сделать напрямую через интерфейс безопасности Windows.

```
PS C:\WINDOWS\system32> Start-MpScan -ScanType QuickScan
```

Рис. 69. Быстрая антивирусная проверка

А вот полную проверку проще делать через командную оболочку PowerShell. При полной проверке каждый файл на компьютере с Windows проверяется на наличие вредоносных программ.

```
PS C:\WINDOWS\system32> Start-MpScan -ScanType FullScan
```

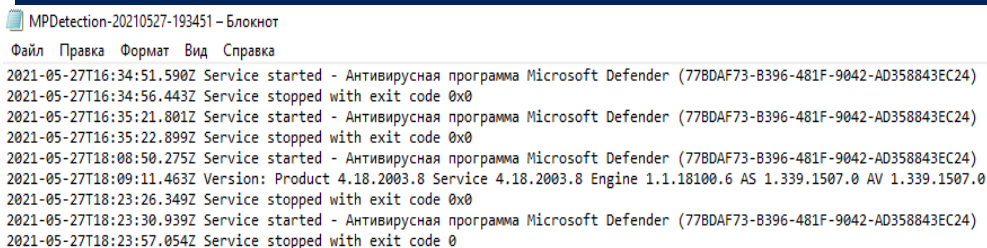


Рис. 70. Полная антивирусная проверка

Автономное сканирование – это мощная функция, которая позволяет удалять трудно обнаруживаемые вредоносные программы.

Следующая команда заставляет Windows загрузиться в автономном режиме Windows Defender и просканировать всю систему на наличие вирусов.

```
PS C:\WINDOWS\system32> Start-MpWdOScan
```

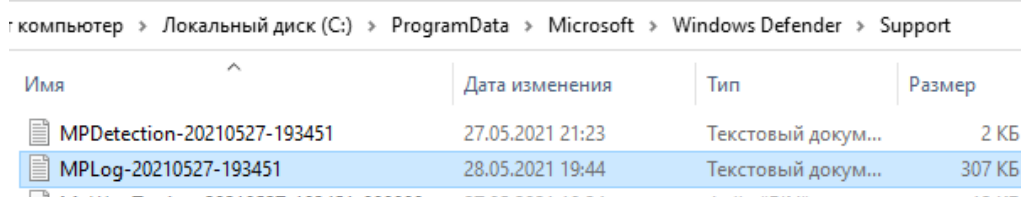


Рис. 71. Автономная проверка системы

Выполнить

1. Введите команду для антивирусной проверки, используя команду PowerShell, произведите проверку ОС.

2. Скрипт информации о действиях с учетными записями.

В этом пункте создадим скрипт, который будет контролировать следующие события:

- а) 4720 – создание нового пользователя;
- б) 4722 – включение учетной записи;
- в) 4725 – отключение учетной записи;
- г) 4726 – удаление учетной записи;
- д) 4740 – блокировка учетной записи;
- е) 4767 – разблокировка учетной записи.

Листинг 1. Скрипт информации о действиях с учетными записями

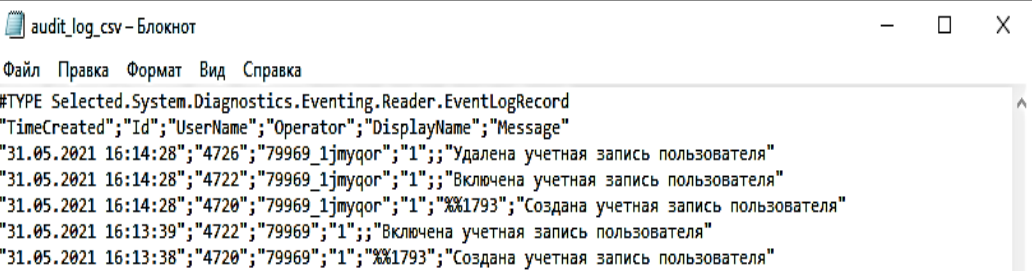
```
Get-WinEvent -EA SilentlyContinue -FilterHashtable @{
>> LogName = "Security";
>> StartTime = (Get-Date).AddHours(-1);
>> ID = 4720,4722,4725,4726,4740,4767 } |
>> select TimeCreated,
>> Id,
>> @{N = 'UserName'; E = {(([xml]$_).ToXml()).event.EventData.Data | ?
Name -eq TargetUserName').#text'}},
>> @{N = 'Operator'; E = {(([xml]$_).ToXml()).event.EventData.Data | ?
Name -eq SubjectUserName').#text'}},
>> @{N = 'DisplayName'; E = {(([xml]$_).ToXml()).event.EventData.Data |
? Name -eq DisplayName').#text'}},
>> @{N = 'Message'; E = {$_ .Message.Split('.')[0]}} | Export-Csv -Delim-
iter ';' -Encoding UTF8 -Path C:\Temp\audit_log_csv -Append
```

Введите команду на приведенном ниже рис. 72.

```

PS C:\WINDOWS\system32> Get-WinEvent -EA SilentlyContinue -FilterHashtable @{
>>   LogName = "Security";
>>   StartTime = (Get-Date).AddHours(-1);
>>   ID = 4720,4722,4725,4726,4740,4767 } |
>> select TimeCreated,
>>   Id,
>>   @{N = 'UserName'; E = {([xml]$_).event.EventData.Data | ? Name -eq 'TargetUserName'}.#text}},
>>   @{N = 'Operator'; E = {([xml]$_).event.EventData.Data | ? Name -eq 'SubjectUserName'}.#text}},
>>   @{N = 'DisplayName'; E = {([xml]$_).event.EventData.Data | ? Name -eq 'DisplayName'}.#text}},
>>   @{N = 'Message'; E = {($_.Message.Split(' ')[0])} | Export-Csv -Delimiter ';' -Encoding UTF8 -Path C:\Temp\audit_log_csv -Append
PS C:\WINDOWS\system32>

```



```

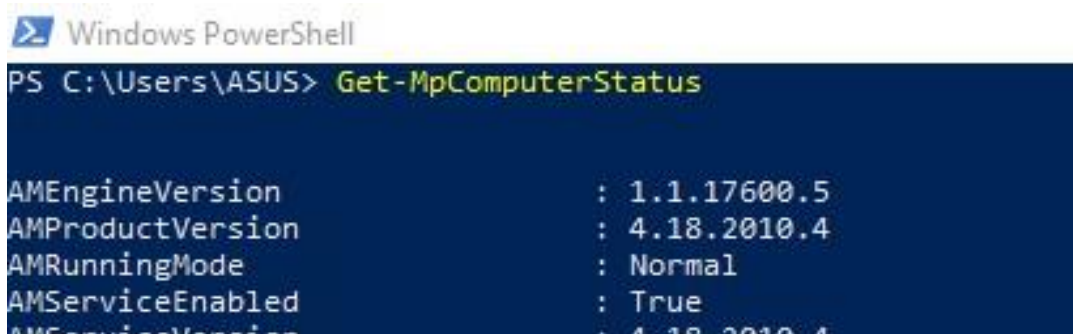
audit_log_csv – Блокнот
Файл Правка Формат Вид Справка
#TYPE Selected.System.Diagnostics.Eventing.Reader.EventLogRecord
"TimeCreated";"Id";"UserName";"Operator";"DisplayName";"Message"
"31.05.2021 16:14:28";"4726";"79969_1jmyqor";"1";"Удалена учетная запись пользователя"
"31.05.2021 16:14:28";"4722";"79969_1jmyqor";"1";"Включена учетная запись пользователя"
"31.05.2021 16:14:28";"4720";"79969_1jmyqor";"1";"%1793";"Создана учетная запись пользователя"
"31.05.2021 16:13:39";"4722";"79969";"1";"Включена учетная запись пользователя"
"31.05.2021 16:13:38";"4720";"79969";"1";"%1793";"Создана учетная запись пользователя"

```

Рис. 72. Скрипт информации о действиях с учетными записями

3. Введите команду на приведенном ниже рис. 73.

Get-MpComputerStatus –list команда выводит на экран состояние антивирусного программного обеспечения на компьютере.



```

Windows PowerShell
PS C:\Users\ASUS> Get-MpComputerStatus

AMEngineVersion      : 1.1.17600.5
AMProductVersion     : 4.18.2010.4
AMRunningMode        : Normal
AMServiceEnabled     : True
AMServiceVersion     : 4.18.2010.4

```

Рис. 73. Get-MpComputerStatus

4. Введите команду на приведенном ниже рис. 74.

Get-MpPreference – эта команда получает настройки для сканирования и обновлений защитника.

```
Windows PowerShell
PS C:\Users\ASUS> Get-MpPreference

AllowNetworkProtectionOnWinServer           : False
AttackSurfaceReductionOnlyExclusions       :
AttackSurfaceReductionRules_Actions        :
AttackSurfaceReductionRules_Ids            :
CheckForSignaturesBeforeRunningScan        : False
CloudBlockLevel                             : 1
CloudExtendedTimeout                       : 1
ComputerID                                 : 9459C3F5-BF63-4E91-A2AE-F6AD7844309A
ControlledFolderAccessAllowedApplications  :
ControlledFolderAccessProtectedFolders     :
DisableArchiveScanning                    : False
```

Рис. 74. Используем Get-MpPreference

5. Get-MpThreat – эта команда выводит на экран историю угроз, обнаруженных на компьютере.

Введите команду на приведенном ниже рис. 75.

```
Windows PowerShell
PS C:\Users\ASUS> Get-MpThreat

CategoryID           : 27
DidThreatExecute     : False
IsActive             : True
Resources            : {file:_C:\Program Files\DAEMON Tools Lite\lang\FIN.dll
RollupStatus         : 1
SchemaVersion        : 1.0.0.0
SeverityID           : 1
ThreatID             : 268654
ThreatName           : App:Daemon_Tools_Lite_BundleInstaller
TypeID               : 0
PSComputerName       :
```

Рис. 75. Используем Get-MpThreat

6. Используем Get-MpThreatCatalog.

Get-MpThreatCatalog – эта команда получает известные угрозы из каталога определений.

7. Start-MpScan – на наш взгляд можно назвать основной программой, так как она является запуском сканирования.


```
Windows PowerShell
PS C:\Users\ASUS> start-MpScan

start-MpScan
  Выполнено 0/1
 [
  Метод CIM Start объекта CIM ROOT\Microsoft\Windows\Defender\MSFT_MpScan
  This might take some time, depending on the type of scan selected.
  [ooooooooooooooooooooooooooooooooooooooooooooooooooooo

  Quick Scan
```

Рис. 76. Используем Start-MpScan

8. Команда start-MpWDOScan запуск автономного сканирования Защитника Windows.

Внимание! При начале автономного сканирования система перезагружается и появляется загрузочный экран Защитника Windows, после завершения автономного сканирования вы вернетесь автоматически на рабочий стол.

```
Windows PowerShell
Windows PowerShell
(С) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)

PS C:\Users\ASUS> start-MpWDOScan
```

Рис. 77. Запуск start-MpWDOScan

9. Команда get-MpThreat. Рассмотрев основные командлеты и разобравшись, как они работают, необходимо более подробно остановиться на командлете get-MpThreat.

```

Windows PowerShell
CategoryID      : 8
DidThreatExecute : False
IsActive       : False
Resources      : {file:_E:\EMPRESS\Binaries\Retail\EMP.dll, file:_E:\EMPRESS\Binaries\Retail\EMP.dll, file:_E:\EMPRESS\Binaries\Retail\EMP.dll, file:_E:\EMPRESS\Binaries\Retail\EMP.dll...}
RollupStatus   : 33
SchemaVersion  : 1.0.0.0
SeverityID     : 5
ThreatID       : 2147723317
ThreatName     : Trojan:Win32/Woreflint.A!c1
TypeID         : 0
PSComputerName :

CategoryID      : 8
DidThreatExecute : False
IsActive       : False
Resources      : {file:_C:\Users\ASUS\AppData\Local\Temp\is-25IA8.tmp\setup.tmp, amsiuc:_pid:00001768, file:_C:\Users\ASUS\AppData\Local\Temp\is-50VT8.tmp\setup.tmp, internalamsiuc:_27D18764FF429A6813886C5D28597524...}
RollupStatus   : 33
SchemaVersion  : 1.0.0.0
SeverityID     : 5
ThreatID       : 2147735505
ThreatName     : Trojan:Win32/Wacatac.B!m1
TypeID         : 0
PSComputerName :

CategoryID      : 8
DidThreatExecute : False
IsActive       : False
Resources      : {file:_C:\Users\ASUS\Downloads\Red.Dead.Redemption.Crackfix.V2-EMPRESS+Mr_Goldberg.7z, webfile:_C:\Users\ASUS\Downloads\Red.Dead.Redemption.Crackfix.V2-EMPRESS+Mr_Goldberg.7z|https://s102sas.storage.yandex.net/rdisk/97c46ab78b19066789c1ff5a216fb12038ddc0246f126eb0ccac3c08aeb5eafa/5f957af0/GREbRrrZ7hv2LSojwUfiyKGbM32ckYsrjND7eoRUGxBZ-JAMBDWys0ejZDpOmRxtjuyGn8NwHBoNYkXb1QJuA=?uid=524757312&filename=Red.Dead.Redemption.Crackfix.V2-EMPRESS%2BMr_Goldberg.7z&disposition=attachment&hash=&limit=0&content_type=application%2Fz-compressed&owner_uid=524757312&fsize=38421046&hid=5971df727854c5811203d57848a5d120&media_type=compressed&tknv=v2&etag=f27b1cca43fdc9352fda7169bda31269&rtoken=BJjjWiMvQ6CL&force_default=yes&yrcid=na-57e5e5d911d3629f244748c4302fecb2-downloader18e&ts=5b27ea3209c00&s=62874446e911cb3bda1df3967064ca0e5cfd63f51e24cd93c491b199bf3e4fda&pb=U2FsdGVkX1980h1cGHYKpIWCfDC-UydBHkU4qVooGR7sm_zQFPY1n5ewQ5wyezuBNgwAWAghD823KCyqGQCUs8N8XN0C8gSvF6SUVovr-IWo|pid:10648,ProcessStart:132480910681253816}
RollupStatus   : 1
SchemaVersion  : 1.0.0.0
SeverityID     : 5
ThreatID       : 2147760506
ThreatName     : Trojan:Win32/CryptInject!m1
TypeID         : 0
PSComputerName :

```

Рис. 78. Команда `get-MpThreat`

Строка `CategoryID` на рис. 78 указывает нам на тип вредоносного ПО. Строка `SeverityID` указывает на степень угрозы от 1 до 5. `ThreatID` номер, который был назначен вредоносной программе / угрозе в качестве формы идентификации. `ThreatName` имя, данное вредоносной программе, соответствует номеру `ThreatID`.

Таблица 3

Идентификаторы (ID) вредоносных объектов

1	Advvare
2	Spyware
3	Passvordstealer
4	Trojan Down loader
5	Червь
6	Черный ход

7	Remoteaccesstrojan
8	Троянский конь
9	Emailflooder
10	Keylogger
11	Dialer
12	Monitoringsoftware
13	Browsermodifier
14	cook
15	Browserplugin
16	Aoexploit
17	Huker
18	Securitydisabler
19	Jokeprogram
20	Hostileactivexcontrol
21	Softwarebundler
22	Stealthnotifier

Использование Get-EventLog при помощи скрипта. В данном примере необходимо вывести на экран пользователей, которые производили вход в систему. Чтобы лучше понять, как работает скрипт, воспользуемся таблицами значений EventID и EntryType. Из поставленной задачи ясно, что нужно рассматривать события, где eventid = 528 и entrytype = 10.

Выполнить

Команда будет выглядеть так:

```
Get-EventLog security -message "*Тип входа:?10*" -after (Get-date -hour 0 -minute 0 -second 0) | ?{$_ .eventid -eq 528 }
```

Результатом будет вывод таблицы. Изменим команду так, чтобы в таблице видели время, имя пользователя, IP-адрес. После этого уже создадим объект и запишем в него нужные нам данные.

Воспользуемся скриптом, он представлен ниже.

Листинг 2. Скрипт Get-Eventlog

```
$Events = Get-EventLog security -message "*Тип входа:?10*" -after (get-date -hour 0 -minute 0) | ?{$_ .eventid -eq 528 }
$Data = New-Object System.Management.Automation.PSObject
$Data | Add-Member NoteProperty Time ($null)
$Data | Add-Member NoteProperty UserName ($null)
$Data | Add-Member NoteProperty Address ($null)
$Events | % {
```

```

$Data.time = $_.TimeGenerated
$message = message.split("n") | % { $_.trimstart() } | % { $_.trimend() }
$Data.UserName = ($message | ?{ $_ -like "Пользователь:*" } | % { $_ -replace "^.+:" } )
$Data.Address = ($message | ?{ $_ -like "Адрес сети источника:*" } | % { $_ -replace "^.+:" })
$data
}

```

Запускаем скрипт командой: ...\. (имя скрипта.ps1).

Рассмотрим параметры скрипта, его переменные в процессе создания кода:

1. `$Events = Get-EventLog security -message "*Тип входа:?10*" -after (get-date -hour Q -minute 0 -second 0) | ?{ $_.eventid -eq 528 }` – заносим результаты выборки по событиям в переменную, чтобы в дальнейшем было удобно с ней работать.

2. Далее создадим «шаблон» будущей таблицы, содержащей три значения: время, имя пользователя и адрес.

```

$Data=      New-Object System.Management.Automation.PSObject
$Data|      Add-Member      NoteProperty      Time ($null)
$Data|      Add-Member      NoteProperty      UserName ($null)
$Data|      Add-Member      NoteProperty      Address ($null)

```

3. `$ Events | % { }` – пройдемся по каждому объекту который будет в результатах отбора `$Data.time = TimeGenerated` – заносим время (`^message = message.split("n") | % { $_.trimstart() } | % { $_.trimend() }` – в переменную `message` заносим массив строк, которые разделяются символом переноса строки `fn`), функции `trimstart`, `trimend` убирают все лишние символы в конце и в начале строки, функция `split` разделяет строку).

4. `JData.UserName = ($message | ?{ $_ -like "Пользователь:*" } | % { $_ -replace "A.+:" })`

5. Далее во вновь образовавшемся массиве мы ищем совпадения по строке «Пользователь:» и «Адрес сети источника:», а `-replace` в дальнейшем удаляет эти регулярные выражения, оставляя саму информацию (`JData.Address = ($message | ?{ $_ -like "Адрес сети источника:*" } | % { $_ -replace "A.+:" })`)).

Если скрипт не запускается, то скорее всего powershell не настроен на выполнение скриптов. Можно попробовать воспользоваться командой `set-executionpolicy remotesignet`.

Стоит заметить, что данный скрипт является шаблоном. Путем изменения параметров значений EntryType и EventID можем настроить скрипт на выполнение той задачи, которая нам требуется. В самом скрипте меняем значения на нужные значения, которые берем из параметров значений EntryType (рис.18) и EventID.

Рассмотрим задачи аудита (особенности аудита в ОС Windows). Политика аудита системы Windows определяет, какой тип информации о системе вы найдете в журнале безопасности. Windows использует девять категорий политик аудита и пятьдесят подкатегорий политик аудита, чтобы предоставить вам более детальный контроль над тем, какая информация регистрируется.

Есть три ведущих типа журналов событий в Windows:

1. Журнал приложений (Application Log) – в данный журнал приложения записывают свои сообщения о событиях.

2. Системный журнал (System Log) – тут находятся события от компонентов операционной системы.

3. Журнал безопасности (Security Log) – имеет информацию, необходимую для проведения аудита безопасности.

Все журналы размещены в папке %Systemroot%\System32\Config. Рассмотрим категории политики аудита. Каждая система Windows имеет девять категорий политик аудита и пятьдесят подкатегорий политик, которые вы можете включить или отключить. (Windows NT имеет только семь категорий; Windows 2003 и версии выше имеет девять категорий, но не имеет подкатегорий.)

В локальной политике безопасности можно увидеть настройки только для основных категорий:

1. Аудит событий входа в аккаунт.
2. Аудит событий входа.
3. Аудит управления учетными записями.
4. Аудит доступа к службе каталогов.
5. Аудит доступа к объектам.
6. Изменение политики аудита.
7. Использование привилегий аудита.
8. Аудит отслеживания процессов.
9. Аудит системных событий.

Рассмотрим **характеристики аудита**. Параметры политики аудита безопасности в разделе «Конфигурация политики аудита Settings\advanced безопасности» может помочь соблюдать требования безопасности аудита, путем отслеживания определенных действий.

Правильный системный список управления всем доступом (SACL) применяется ко всем файлам реестра, папкам, разделам реестра на компьютере.

Политика имеет девять подкатегорий:

1. Вход в систему.
2. Выход.
3. Блокировка аккаунта.
4. Основной режим IPsec.
5. Быстрый режим IPsec.
6. Расширенный режим IPsec.
7. Специальный вход.
8. Другие события входа/выхода.
9. Сервер сетевой политики.

Аудит управления учетными записями – позволяет определить события, связанные с управлением учетными записями на компьютере.

Политика аудита учетных записей, которую можно использовать для отслеживания изменений в учетных записях и группах пользователей, полезна для аудита действий администраторов и сотрудников службы поддержки. Эта политика регистрирует сброс пароля, вновь созданные учетные записи и изменения в членстве в группах; одна из подкатегорий категории «Управление учетными записями», «Другие события управления учетными записями», регистрирует изменения в политике блокировки и паролей. Политика имеет пять подкатегорий:

1. Управление учетной записью пользователя.
2. Управление учетными записями компьютеров.
3. Управление группой безопасности.
4. Управление распределительной группой.
5. Управление группой приложений.

Аудит доступа к службе каталогов – определяет, принадлежит ли пользователю событие, при доступе пользователя к объекту каталога Active Directory. Основная цель политики доступа к службе каталогов аудита – предоставить низкоуровневый контрольный журнал изменений объектов в AD. Политика имеет четыре подкатегории:

1. Доступ к службе каталогов.
2. Изменения службы каталогов.
3. Репликация службы каталогов.
4. Подробная репликация служб каталогов.

Аудит входа в систему – фактически контролирует категорию «Вход/Выход из системы». Основная цель политики – записывать все попытки использовать учетную запись домена или локальную учетную запись для входа в локальный компьютер или выхода из него.

Политика, например, не отслеживает пользователя, который использует учетную запись домена для входа на рабочую станцию.

Политика имеет четыре подкатегории:

1. Проверка учетных данных.
2. Служба аутентификации Kerberos.
3. Kerberos Service Ticket Operations.
4. Другие события входа в аккаунт.

Аудит доступа к объектам – обрабатывает доступ ко всем объектам, находящимся вне активного каталога. Определяет, подлежит ли аудиту событие попытки доступа пользователя к объекту (например, к файлу, папке, разделу реестра, принтеру и т. д.), для которого указан собственный список контроля системного доступа (SACL).

Политика имеет десять подкатегорий:

1. Файловая система.
2. Реестр.
3. Kernel Object.
4. SAM.
5. Сертификационные услуги.
6. Application Generated.
7. Handle Manipulation.
8. Файловый ресурс.
9. Фильтрующая платформа Drop Drop.
10. Подключение платформы фильтрации.

Аудит изменения политики – основной целью политики изменения политики аудита является уведомление вас об изменениях в важных политиках безопасности в локальной системе. Такие изменения включают изменения в политике аудита системы или, если локальная система является DC, изменения в доверительных отношениях.

Политика имеет пять подкатегорий:

1. Изменение политики аудита.
2. Изменение политики аутентификации.
3. Изменение политики авторизации.
4. Изменение политики на уровне правил MPSSVC.
5. Фильтрация изменений политики платформы.

Аудит использования привилегий – отслеживает реализацию прав пользователей. Политика имеет три подкатегории:

1. Чувствительное использование привилегий.
2. Использование нечувствительных привилегий.
3. Другие события использования привилегий.

Аудит отслеживания процессов – записывает события в категорию подробного отслеживания. Основная цель этой политики – отслеживать каждую программу, которая выполняется системой или конечными пользователями. Политика имеет четыре подкатегории:

1. Создание процесса.
2. Завершение процесса.
3. Деятельность DPAPI.
4. RPC события.

Аудит системных событий – определяет, подлежат ли аудиту события перезагрузки или отключения компьютера, а также события, влияющие на системную безопасность или на журнал безопасности. Политика имеет четыре подкатегории:

1. Изменение состояния безопасности.
2. Расширение системы безопасности.
3. События целостности безопасности.
4. События драйвера IPsec.

Средства Аудита безопасности Windows позволяют контролировать доступ к файлам, папкам, ключам реестра и другим объектам, и другим системным объектам, у которых есть SACL. Мониторинг доступа к файлам для файл-сервера может быть важной задачей и средства аудита безопасности Windows помогают администраторам в этом.

Аудит доступа к файлам и реестру позволяет обнаруживать попытки несанкционированного доступа к файлам и предотвращать или отслеживать изменения конфигураций системы и программ. Наиболее общими типами событий для аудита ОС Windows являются:

1. Доступ к таким объектам, как файлы и папки.
2. Управление учетными записями пользователей, групп, программ, Интернет-ресурсов.
3. Вход пользователей в систему и выход из нее.
4. Доступ и работа с процессами.

Событие в журнале безопасности ОС Windows имеет ключевое слово для аудита успеха или аудита отказа. Когда включается политика аудита, можно включить политику, чтобы регистрировать события успеха, события сбоя или оба события, в зависимости от политики. Все девять политик аудита генерируют события успеха, но только некоторые политики генерируют события отказа.

Параметры безопасности

Параметр безопасности	Возможный итог попытки доступа
Аудит изменения политики	<p>Аудит успехов означает создание записи аудита при каждом успешном изменении политик назначения прав пользователей, политик аудита или политик доверительных отношений.</p> <p>Аудит отказов означает создание записи аудита при каждой неудачной попытке изменения политик назначения прав пользователей, политик аудита или политик доверительных отношений</p>
Аудит использования привилегий	<p>Аудит успехов означает создание записи аудита для всякого успешного применения привилегий.</p> <p>Аудит отказов означает создание записи аудита для всякого неудачного применения привилегий</p>
Аудит отслеживания процессов	<p>Аудит успехов означает создание записи аудита для всякого успешного события, связанного с отслеживаемым процессом.</p> <p>Аудит отказов означает создание записи аудита для всякого неудачного события, связанного с отслеживаемым процессом</p>
Аудит системных событий	<p>Аудит успехов означает создание записи аудита для всякого успешного системного события.</p> <p>Аудит отказов означает создание записи аудита для всякого неудачного завершения системного события</p>
Аудит событий входа в систему	<p>Аудит успехов означает создание записи аудита для всякой успешной попытки входа в систему. Аудит отказов означает создание записи аудита для всякой неудачной попытки входа в систему</p>
Аудит управления учетными записями	<p>Аудит успехов и отказов означает создание записи аудита для каждого успешного и неудачного события</p>

Новейшие средства аудита событий безопасности используют два ведущих подхода для организации хранения событий безопасности в файлах журнала аудита:

- хранение событий в упорядоченных файлах;
- хранение событий в текстовых файлах, где любая отдельная текстовая строка задает другое событие.

В таком случае, если данные благополучно получены, то дальше их нужно конвертировать в данные структур, используемых в системе для представления событий. Так как считанные данные о событиях безопасности содержатся в этом формате, в котором они хранятся в журналах аудита, то их нужно преобразовать к структуре, комфортной для дальнейшего представления в процессе обработки в системе. Для этого, как правило, нужно выделить из массива информации отдельные события и значения, задающих события.

В Windows Server 2003 определяет, какие действия нужно регистрировать, и настраивается с поддержки оснастки Security Policy (Политика безопасности). Политика аудита находится в подсистеме Local Security Authority Subsystem (LSASS – подсистема полномочий локальной безопасности), которая передает ее программе Security Service Monitor (SRM – монитор безопасности служб) при запуске и впоследствии, внесении изменений. SRM работает вместе с диспетчером объектов, они создают записи аудита и передают их LSASS. Подсистема добавляет некоторые вспомогательные сведения и записывает всю информацию в журнал регистрации событий, который в свою очередь копирует данные в журнал безопасности (Security Log).

Если для объекта определен аудит, то объект заносится в системный перечень контроля доступа (SACL – System Access Control List). Перечень определяет информацию о каких операциях и каких пользователях нужно записывать в журнал безопасности. Возможно протоколировать как удачные, так и безуспешные попытки.

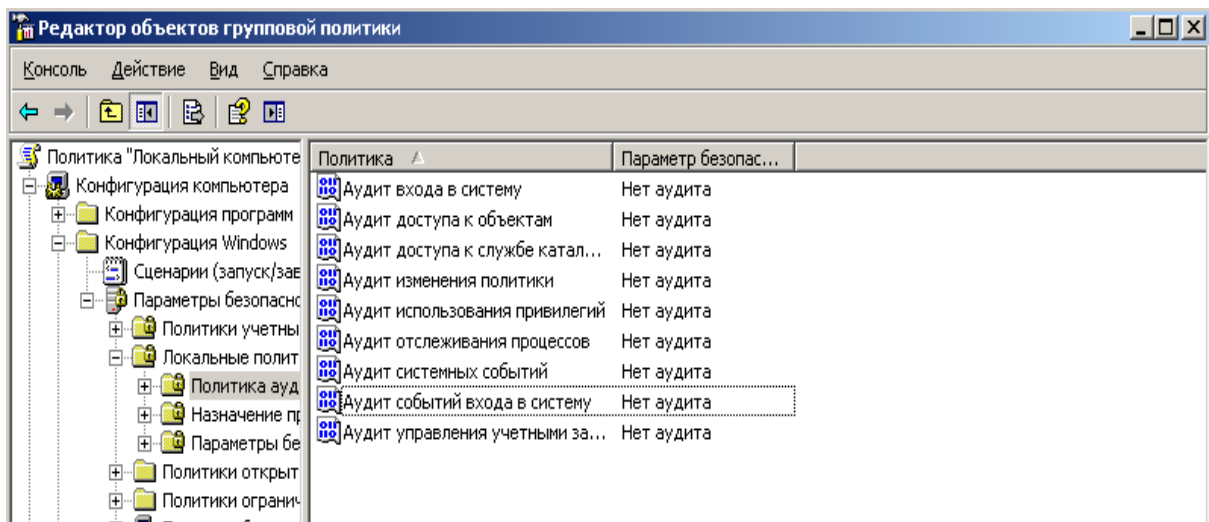


Рис. 79. Консоль редактора объектов групповой политики

Режим аудита необходимо устанавливать для каждого отдельного объекта. На контроллерах доменов, работающих под управлением Windows Server 2003, полный аудит обращений к каталогу имеет возможность привести к сильному снижению производительности. Устанавливаемые опции аудита обязаны соответствовать выполняемым функциям определенного компьютера.

При открытии политики аудита, ее можно или нельзя изменить в зависимости от того, была ли она определена в объекте групповой политики, который был применен к локальной системе. Если компьютер является членом домена AD, он автоматически применяет соответствующие объекты групповой политики из домена.

Windows всегда отображает действующие настройки в консоли MMC Local Security Policy.

Для эффективного аудита Windows Server 2003 необходимо разработать стратегию аудита, которая определяет следующее:

1. События безопасности сервера, которые следует включить в аудит.
2. Наибольшая величина и срок хранения записей для журнала Windows Security. Эти значения можно задать в Event Viewer.
3. Объекты, такие как файлы и папки, которые должны быть интегрированы в мониторинг. Администраторам необходимо включать в аудит только нужные объекты, иначе это вызовет очень быстрое заполнение журналов аудита.
4. Развертывание политики аудита с помощью средства Local Security Policy на автономной машине или в домене.

5. Систематический просмотр журналов безопасности. Значимость аудита заключается как раз в том, что есть возможность просматривать журналы безопасности для выявления вторжений в систему безопасности.

6. Обновление политики аудита по мере необходимости.

Впоследствии задания аудита, направленного на обнаружение и запись доступа сквозь систему безопасности к файлам или же к папкам, есть возможность отслеживать пользователей, которые выполняют доступ к определенным объектам, и анализировать проникновения в систему безопасности. Анализ аудита имеет возможность продемонстрировать, кто выполнил действия, и кто пробовал выполнить действия, которые запрещены.

Наиболее нужными событиями, для которых требуется мониторинг, являются события, Audit Policy Change и Audit System Events. Эти события аудита часто используются для выявления проникновений в систему.

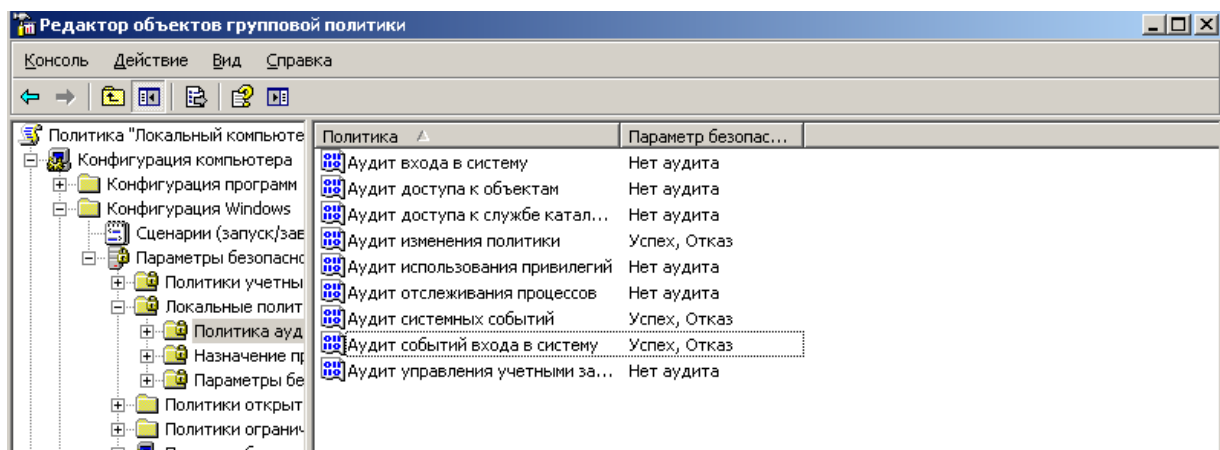


Рис. 80. Показаны изменения в политике аудита

Выполнить

Чтобы создать в домене лес Active Directory и преобразовать компьютер с Windows Server 2003 в контроллер домена в этом лесе, необходимо запустить «Мастер установки Active Directory».

1. Выбираем вариант «контроллер домена в новом домене».
2. Создаем новый домен в новом лесу.
3. Указываем полное имя DNS для нового домена.

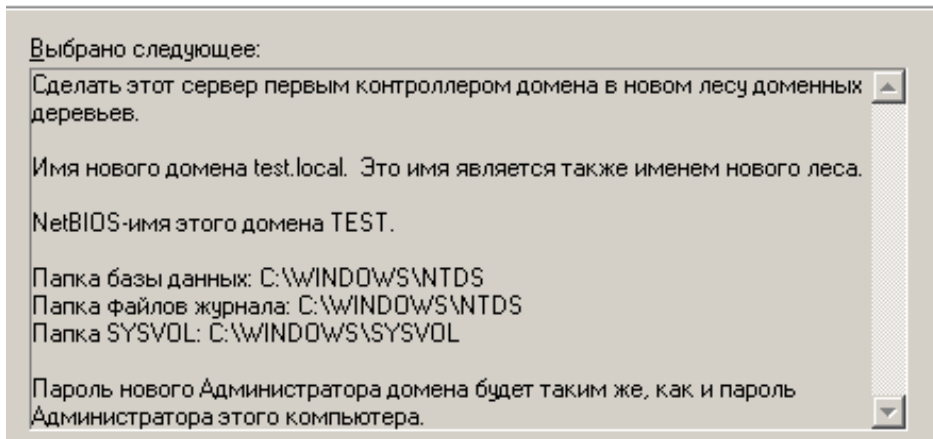


Рис. 81. Выбранные параметры

4. После перезапуска системы необходимо убедиться, что были созданы записи об адресах DNS для нового контроллера домена. Переходим в «Администрирование», затем в "DNS".

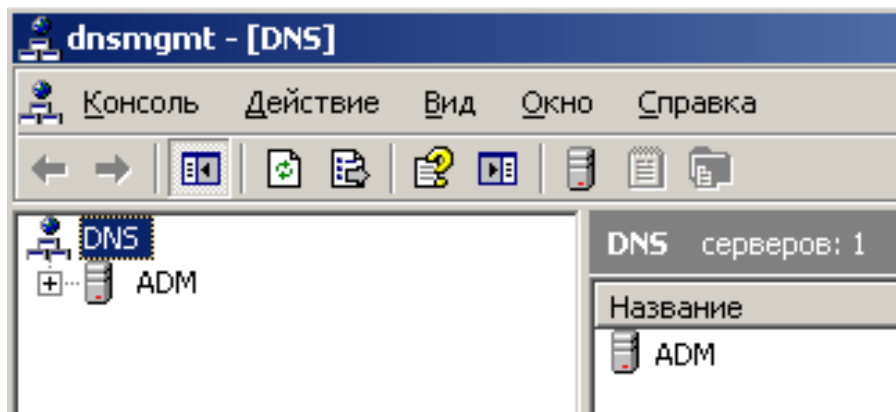


Рис. 82. Показана консоль управления DNS

5. Разворачиваем имя сервера, узел «Зоны прямого просмотра» и «домен» и проверяем наличие папок для правильной работы AD.

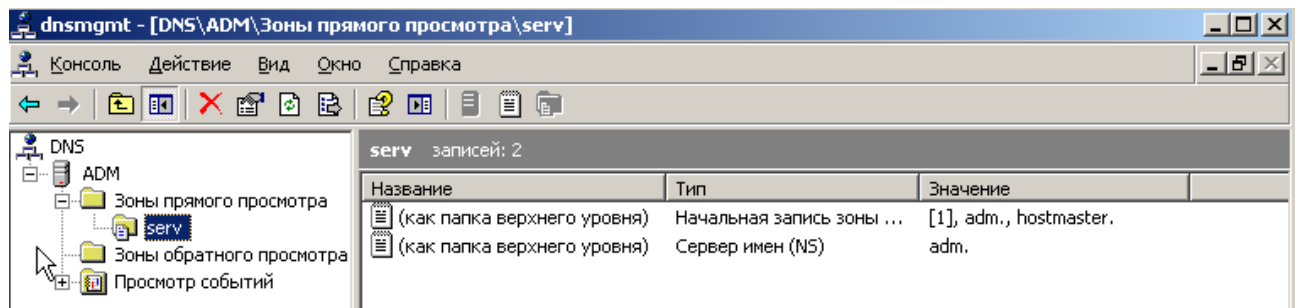


Рис. 9. Показаны необходимые для работы папки

6. Создаем пользователя. Запускаем AD – пользователи и компьютеры, чтобы запустить консоль управления пользователями и компьютерами AD. Создаем пользователя.

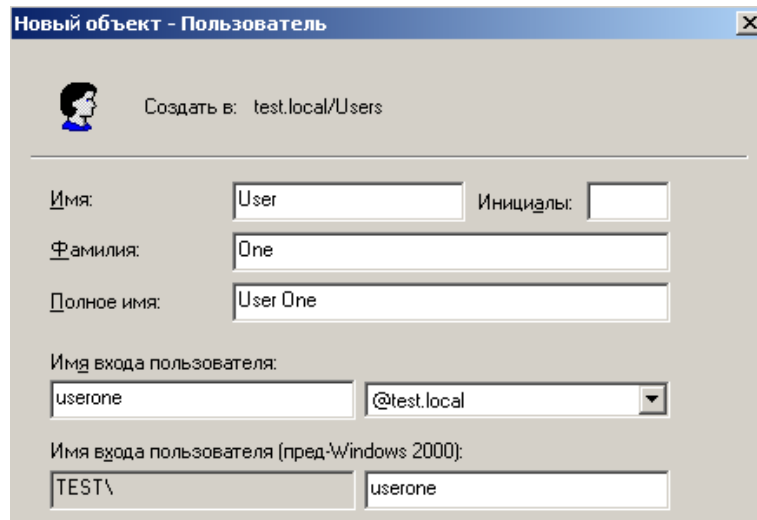


Рис. 83. Показано создание нового объекта – пользователя

Для обеспечения безопасности системы при извлечении смарт-карты, следует включить «Интерактивный вход в систему» и выбрать одно из действий.

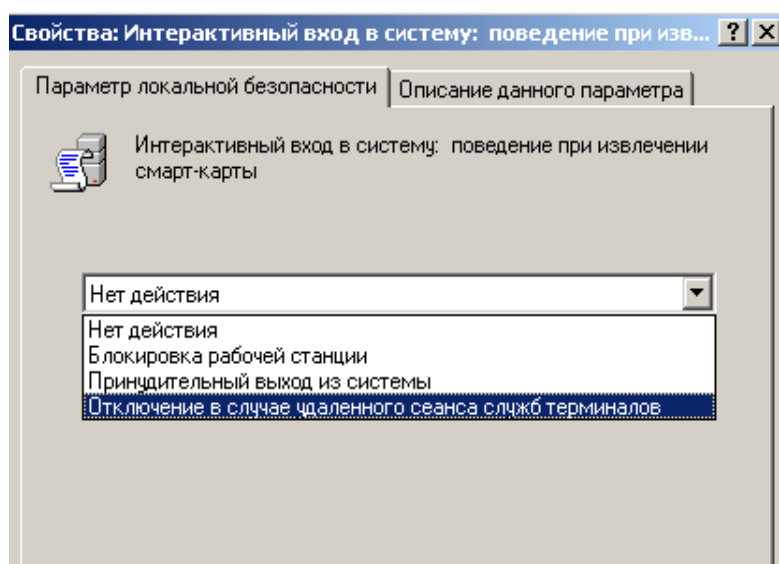


Рис. 84. Показано изменение политики интерактивного входа в систему

Теперь, когда пользователь будет уходить с рабочего места, то система будет действовать по выбранному сценарию. Например, если необходимо дать право доступа только одному пользователю, то нужно сделать следующее.

1. Щелкнуть правой кнопкой мыши на файле или папке, выбрать пункт Properties и затем щелкните на вкладке Security.
2. Щелкнуть на кнопке Advanced и затем на вкладке Auditing.
3. Определить тип изменений в аудите, которые нужно выполнить. Чтобы задать аудит для новой группы или нового пользователя, нужно нажать на Add. В поле Name ввести имя пользователя или использовать кнопку Advanced, чтобы найти конкретного пользователя.

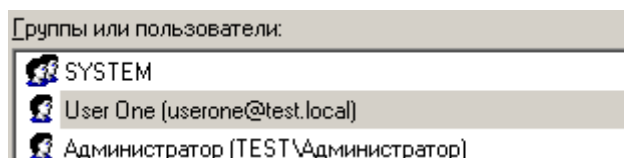


Рис. 85. Поиск конкретного пользователя

Для просмотра или изменения аудита для существующей группы или пользователя нажать на имени и затем по кнопке View/Edit. Чтобы отменить аудит для существующей группы или пользователя, нужно нажать на имени и затем щелкнуть по кнопке Remove.

4. В панели Access выбрать на флажках Successful и/или Failed в зависимости от типа доступа, аудит которого нужен.

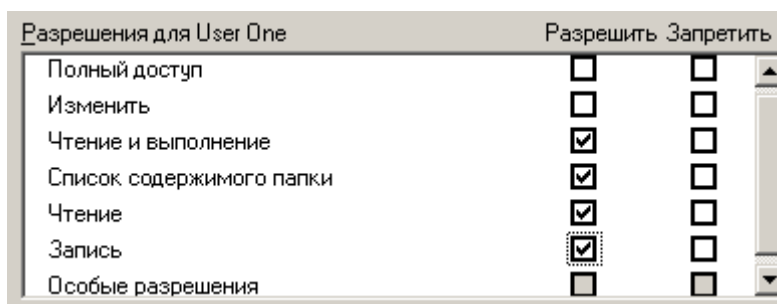


Рис. 86. Показано изменение доступа для пользователя User One

5. Если нужно, чтобы файлы и подпапки в данной папке не наследовали настройки аудита, нужно снять флажок Apply these auditing entries. Данный аудит позволяет обеспечить безопасность файлов или папок от несанкционированного доступа.

Попытки несанкционированного доступа, записанные в журнале Windows Security, можно видеть, как записи предупреждений или ошибок. Требуется:

- выбрать Start/Settings/Control Panel;
- дважды нажать на Administrative Tools и затем дважды на Computer Management;

– раскрыть System Tools и затем Event Viewer. Выбрать Security.
 6. Просмотреть журнал на предмет подозрительных событий безопасности, включая:

- а) неверные попытки входа;
- б) безуспешное использование привилегий;
- в) безуспешные попытки доступа и изменения файлов .bat или .cmd;
- г) попытки изменить привилегии безопасности или журнал аудита;
- д) попытки завершить работу сервера.

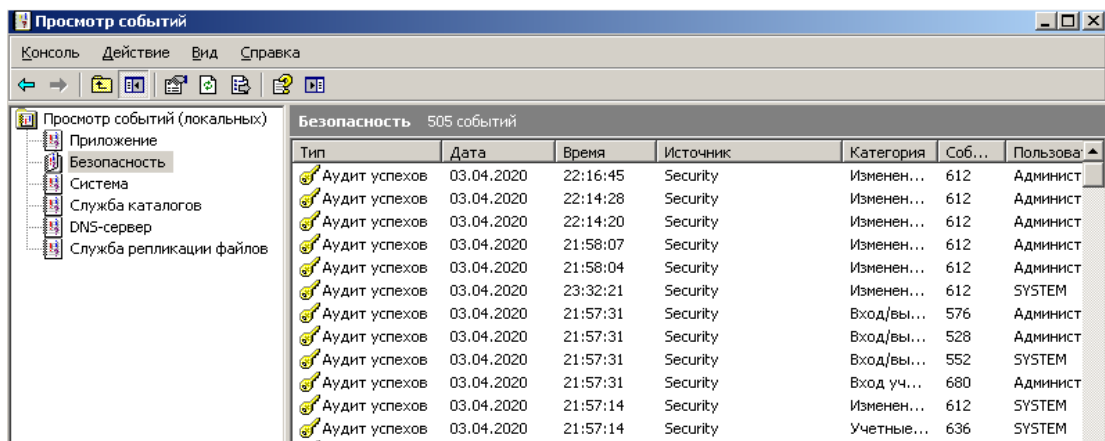


Рис. 87. Интерфейс журнала событий

Журналам безопасности можно назначить права доступа, чтобы ограничить круг пользователей, которые могут читать эти файлы. По умолчанию эти файлы могут читать члены группы Everyone. В разделе «Безопасность» просмотра событий находятся все попытки входа в систему и выхода из нее.

Если в журнале безопасности были замечены попытки несанкционированного доступа, то возникают события, приведенные на рис. 88.

Аудит успехов	04.04.2020	12:57:14	Security	Вход/вы...	528	Администратор	ADM
Аудит успехов	04.04.2020	12:57:14	Security	Вход/вы...	552	SYSTEM	ADM
Аудит успехов	04.04.2020	12:57:14	Security	Вход уч...	680	Администратор	ADM
Аудит отказов	04.04.2020	12:57:10	Security	Вход/вы...	529	SYSTEM	ADM
Аудит отказов	04.04.2020	12:57:10	Security	Вход уч...	680	SYSTEM	ADM
Аудит отказов	04.04.2020	12:57:07	Security	Вход/вы...	529	SYSTEM	ADM
Аудит отказов	04.04.2020	12:57:07	Security	Вход уч...	680	SYSTEM	ADM
Аудит успехов	04.04.2020	12:57:02	Security	Доступ ...	562	SYSTEM	ADM
Аудит успехов	04.04.2020	12:57:02	Security	Доступ ...	560	SYSTEM	ADM
Аудит успехов	04.04.2020	12:57:01	Security	Вход/вы...	551	Администратор	ADM

Рис. 88. Показаны несанкционированные попытки доступа

Выполнить

Нужно выполнить следующее:

1. Отменить права доступа группы Everyone и ограничить доступ группам Administrators и System.

2. Проследить, чтобы в политике указывалось, что делать в случае заполнения всего журнала событий, особенно журнала безопасности. В этом случае рекомендуется требовать, чтобы по возможности была завершена работа сервера.

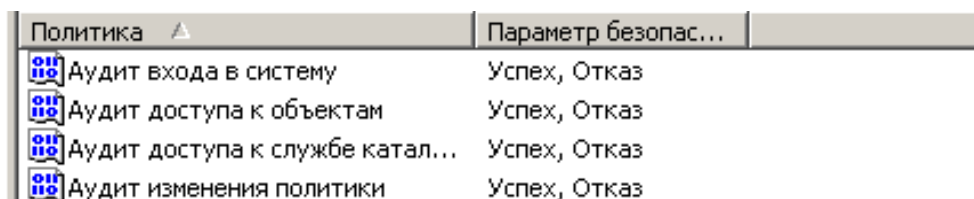
3. Задать настройки политики безопасности, чтобы локальные учетные записи Guest не могли получать доступ к системному журналу и к журналам приложений и безопасности.

4. Установить максимальный размер журнала безопасности.

5. По достижении 30 Мб самые старые события в журналах безопасности будут перезаписываться новыми. Старые события заменяются новыми.

6. Задать настройки политики безопасности, чтобы локальные учетные записи Guest не могли получать доступ к системному журналу и к журналам приложений и безопасности.

7. Проследить, чтобы для системных событий выполнялся аудит как успешных, так и безуспешных попыток, чтобы выявить возможные попытки стирания содержимого журнала безопасности.



Политика	Параметр безопас...
Аудит входа в систему	Успех, Отказ
Аудит доступа к объектам	Успех, Отказ
Аудит доступа к службе катал...	Успех, Отказ
Аудит изменения политики	Успех, Отказ

Рис. 89. Изменения в политике аудита

8. Все администраторы, которые имеют возможность просматривать или модифицировать настройки аудита, должны использовать сложные пароли или двухфакторные методы аутентификации, например, вход по смарт-карте, чтобы предотвратить атаки на эти учетные записи для получения доступа к информации аудита.

Теперь, чтобы стереть события или иным образом вмешаться в журнал безопасности или политику аудита, необходим физический доступ к целевой системе, полномочия администратора этой системы или доступ на запись в объект групповой политики, который применяется к этой системе.

Начиная с операционных систем Microsoft Windows Vista и Server 2008 компания Microsoft сделала шаг вперед в понимании аудита и управления им. Так появился расширенный аудит.

Расширенный аудит позволяет администраторам более избирательно выбирать количество и типы событий для аудита. Например, базовая политика аудита предоставляет один вариант входа в учетную запись, а расширенная политика аудита – четыре. Включение одного базового параметра эквивалентно настройке всех четырех дополнительных параметров. Для сравнения, установка одного расширенного параметра политики аудита не приводит к созданию событий аудита для действий, которые вы не хотите отслеживать.

Рассмотрим необходимые системному администратору сведения, которые должны будут отображаться в журналах событий бухгалтерской компании, после всех настроек:

- 1) информация об учетных записях (их изменение, вход в систему, блокировка и т. д.);
- 2) информация о процессах, которые запускают сотрудники;
- 3) информация о доступе к объектам и использовании разрешений пользователей;
- 4) информация об изменении политик;
- 5) информация о попытках доступа и изменения службы каталогов Active Directory;
- 6) информация о регистрации сетевых подключений.

Для обеспечения записи необходимых нам событий в журнал безопасности сначала надо настроить сам аудит. Абсолютно все возможные события включать не надо, потому что часть из них не поможет нам в дальнейшем анализе для выявления уязвимостей, а лишь затруднит прочтение журнала. Именно поэтому будем настраивать расширенный аудит, где политики аудита разбиты на подкатегории.

Затем загружаем в нашу ОС системный монитор Sysmon. Основная цель добавления Sysmon в том, что он регистрирует создание процесса с полной командной строкой для текущих и родительских процессов, обнаруживает изменения времени создания файла, чтобы понять, когда файл был создан и регистрирует сетевые подключения, включая исходный процесс каждого подключения, IP-адреса, номера портов, имена узлов и имена портов.

При помощи PowerShell и догружая модуль Posh-Sysmon изучаем возможности сортировки журналов событий, используемых нами. В завершении используем скрипт AdAudit, поскольку с его помощью появляется возможность быстрого и удобного анализа аудита доменов.

В Windows достаточно выполнить команду `gpedit.msc` в командной строке, `cmd` или окне «Выполнить» (Win+R).

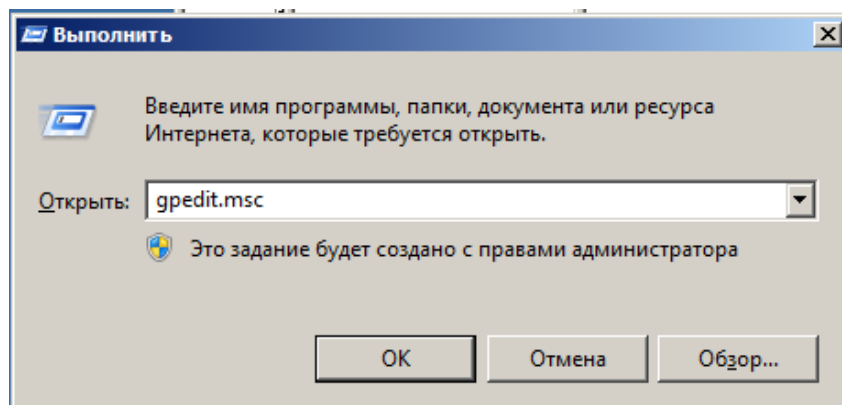


Рис. 90. Оснастка `gpedit.msc`

Полный путь к настройкам аудита выглядит следующим образом: Конфигурация компьютера / Конфигурация Windows / Параметры безопасности / Локальные политики / Политика аудита (Computer Configuration / Windows Settings / Security Settings / Local Policies / Audit Policy).

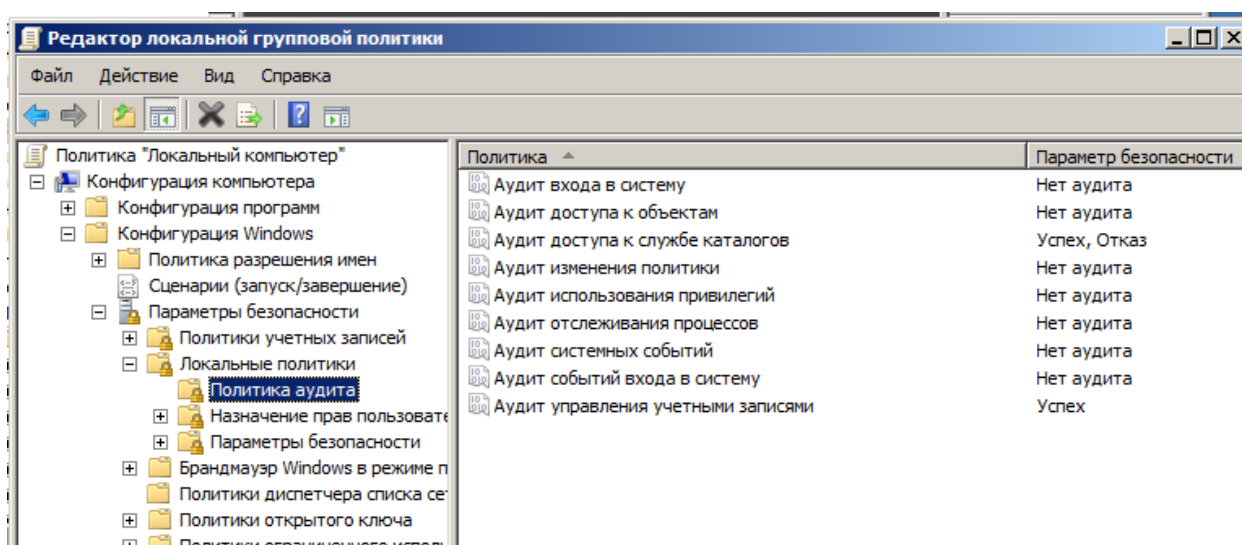


Рис. 91. Путь к политике аудита

Путь к расширенному аудиту: Конфигурация компьютера / Конфигурация Windows / Параметры безопасности / Конфигурация расширенной политики аудита (Computer Configuration / Windows Settings / Security Settings / Advanced Audit Policy Configuration). Теперь можно приступить к настройке политики аудита через подкатегории. События могут иметь значение «Успех и отказ».

Настройка параметров политики в этой категории поможет документировать попытки проверки подлинности данных учетных записей на контроллере домена или в локальном диспетчере учетных записей безопасности (SAM).

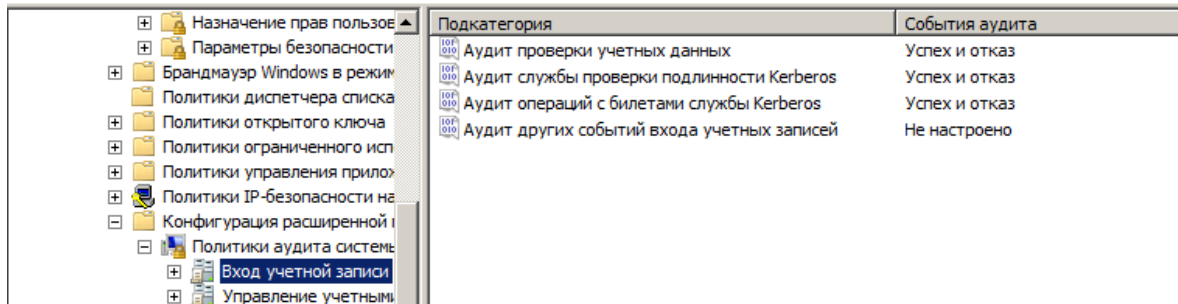


Рис. 92. Вход учетной записи

Параметры политики аудита безопасности в этой категории можно использовать для мониторинга изменений учетных записей и групп пользователей и компьютеров.

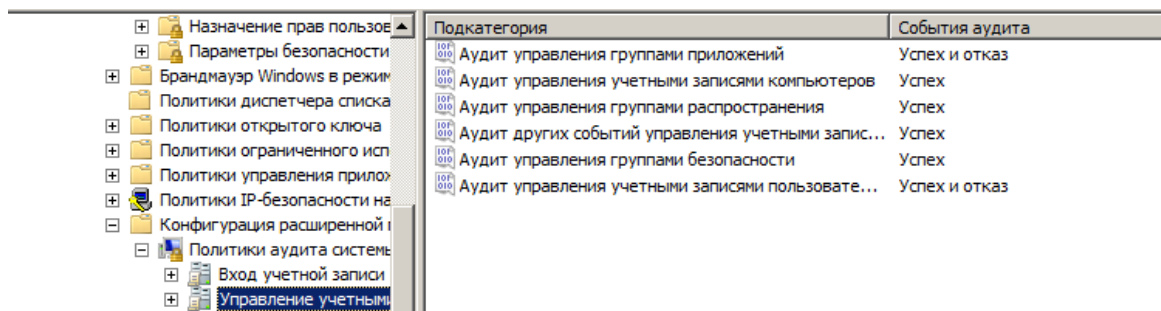


Рис. 93. Управление учетными записями

Подробные параметры политики безопасности отслеживания и события аудита можно использовать для следующих целей:

1. Отслеживание действий отдельных приложений и пользователей на компьютере.
2. Для определения специфики использования ОС.

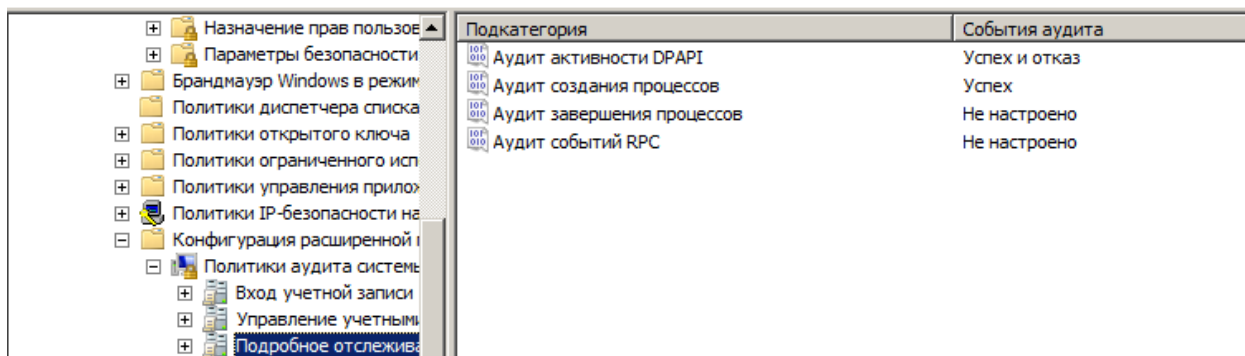


Рис. 94. Подобное отслеживание

Доступ к службе каталогов. Параметры политики аудита безопасности предоставляют подробный след аудита попыток доступа к объектам в службе доменных служб Active Directory (AD DS) и их изменения. Эти события аудита регистрируются только на контроллерах домена.

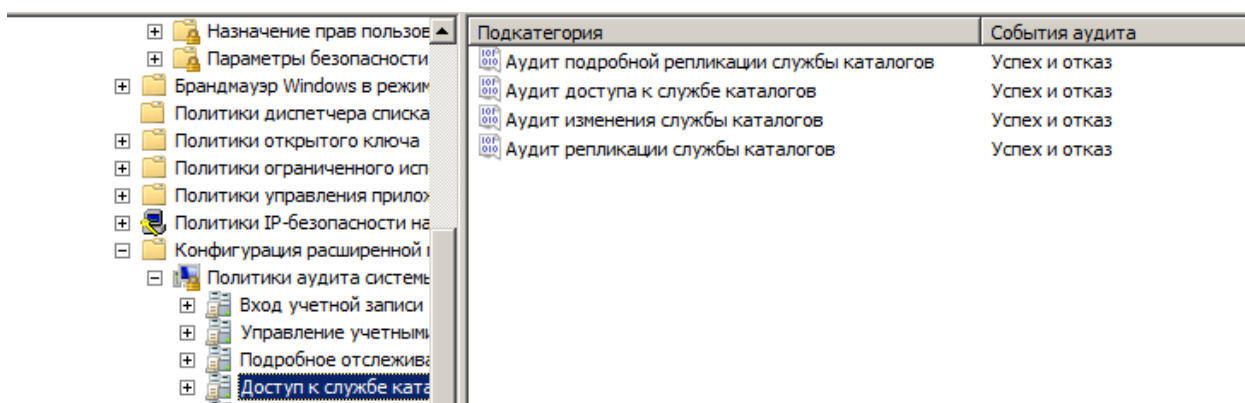


Рис. 95. Доступ к службе каталогов

Параметры политики безопасности «Вход/Выход» и события аудита позволяют отслеживать попытки входа на компьютер в интерактивном режиме или через сеть. Эти события особенно полезны для отслеживания активности пользователей и выявления потенциальных атак на сетевые ресурсы.

Параметры политики доступа к объектам и события аудита позволяют отслеживать попытки доступа к определенным объектам или типам объектов на сети или компьютере.

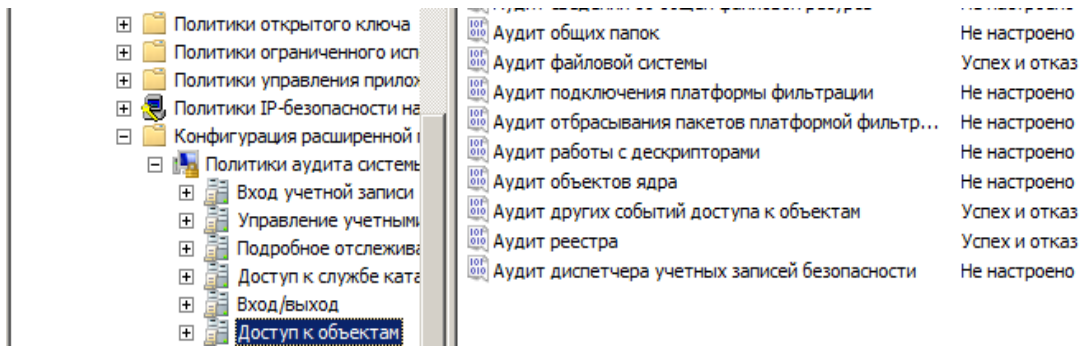


Рис. 96. Доступ к объектам

События аудита изменения политики позволяют отслеживать изменения важных политик безопасности в локальной системе или сети.

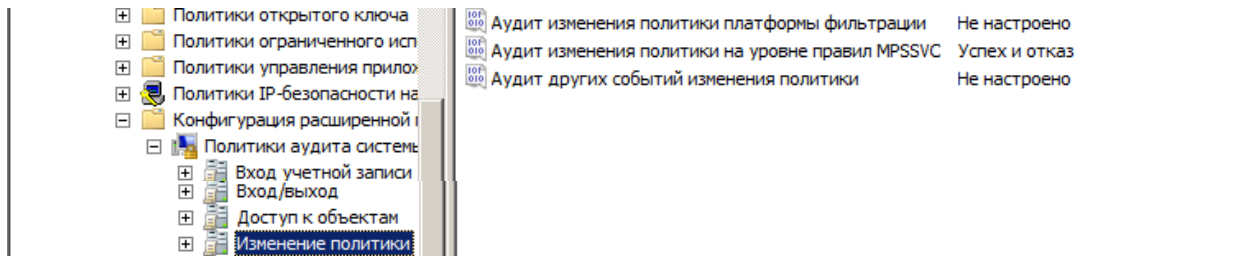


Рис. 97. Изменение политики

Использование прав. Пользователям или компьютерам предоставляется разрешение на выполнение определенных задач в сети. Параметры политики безопасности, использование прав и события аудита позволяют отслеживать использование определенных разрешений в одной или более системах.

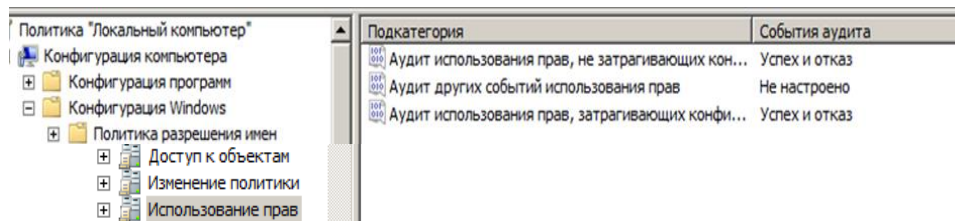


Рис. 98. Использование прав

Система. Параметры политики безопасности системы и события аудита позволяют отслеживать следующие типы изменений на уровне системы на компьютере:

- не включено в другие категории;
- возможные последствия для безопасности.

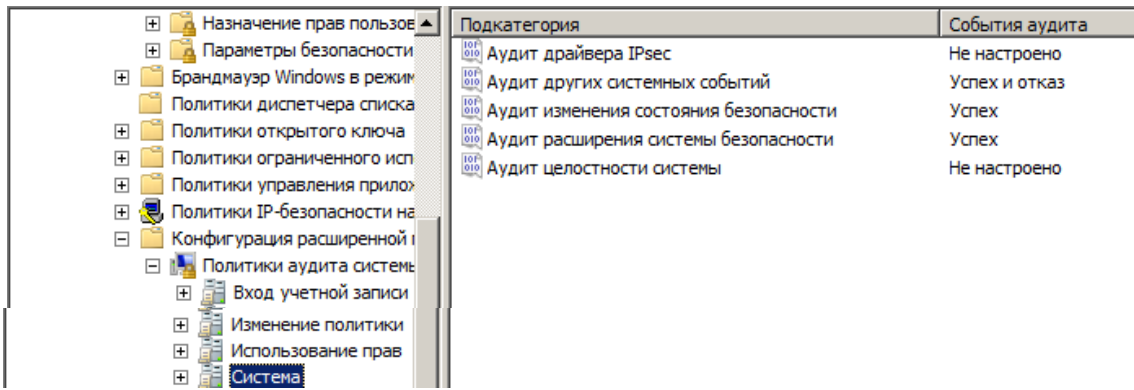


Рис. 99. Настройка «Система»

Параметры политики аудита глобального доступа к объектам позволяют администраторам определять списки управления доступом к компьютерной системе (SACLs) для типа объекта для файловой системы или реестра.

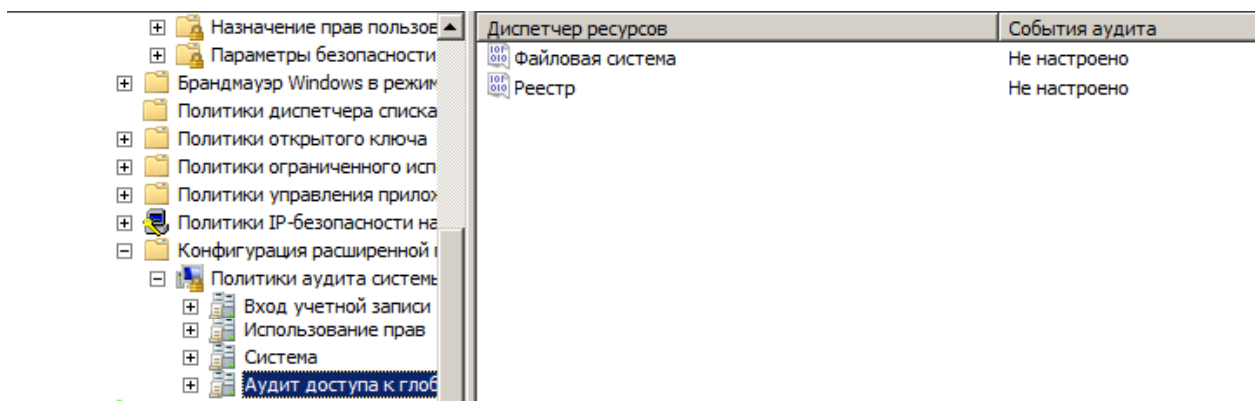


Рис. 100. Аудит доступа к глобальным объектам

Так же для просмотра и настройки политик аудита можно использовать инструмент auditpol.

```

C:\Windows\system32>auditpol /get /category:*
Политика аудита системы
Категория или подкатегория      Параметр
Система
Расширение системы безопасности  Успех
Целостность системы              Без аудита
Драйвер IPSEC                    Без аудита
Другие системные события         Успех и сбой
Изменение состояния безопасности  Успех
Вход/выход
Вход в систему                   Успех и сбой
Выход из системы                  Успех

```

Рис. 101. Работа auditpol

2.3. Реализация журнала аудита с помощью PowerShell

С помощью расширяемого средства автоматизации PowerShell можно всячески сортировать данные из журнала аудита. Для этого используются два командлета, специально предназначенные для запроса информации в журналах событий, – это Get-EventLog и Get-WinEvent.

```
PS C:\Users\Администратор> Get-WinEvent Security -MaxEvents 10
```

TimeCreated	ProviderName	Id	Message
30.05.2022 7:50:07	Microsoft-Windows-Security...	4616	Системное время изменено....
30.05.2022 7:50:07	Microsoft-Windows-Security...	4616	Системное время изменено....
03.06.2022 7:49:57	Microsoft-Windows-Security...	4688	Создан процесс....
03.06.2022 7:49:57	Microsoft-Windows-Security...	4688	Создан процесс....
03.06.2022 7:49:20	Microsoft-Windows-Security...	4634	Выполнен выход учетной зап...
03.06.2022 7:49:20	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:49:20	Microsoft-Windows-Security...	4672	Новому сеансу входа назнач...
03.06.2022 7:49:12	Microsoft-Windows-Security...	4634	Выполнен выход учетной зап...
03.06.2022 7:48:20	Microsoft-Windows-Security...	4634	Выполнен выход учетной зап...
03.06.2022 7:48:20	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...

Рис. 102. Результат работы MaxEvents

Просмотр событий, классифицируемых как отказ обработки правил фильтрации.

```
Администратор: Windows PowerShell
PS C:\Users\Администратор> Get-EventLog Security -EntryType FailureAudit -Newest 10
```

Index	Time	EntryType	Source	InstanceID	Message
7942	июн 03 00:09	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7941	июн 03 00:09	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7929	июн 03 00:07	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7928	июн 03 00:07	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7891	июн 03 00:02	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7890	июн 03 00:02	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7858	июн 03 00:00	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7857	июн 03 00:00	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7814	июн 02 23:55	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...
7813	июн 02 23:55	FailureA...	Microsoft-Windows...	4957	Брандмауэр Windows не применил следующее правил...

Рис. 103. Результат работы FailureAudit

Выполнить

Просмотр событий только с определенным значением кода события:

```
PS C:\Users\Администратор> Get-WinEvent Security -MaxEvents 10 -FilterXPath '*[System[EventID=4624]]'
```

TimeCreated	ProviderName	Id	Message
03.06.2022 7:42:39	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:42:39	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:42:38	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:42:36	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:42:36	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:42:20	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:41:20	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:40:20	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:39:20	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...
03.06.2022 7:38:20	Microsoft-Windows-Security...	4624	Вход с учетной записью вып...

Рис. 104. Результат работы EventID (PowerShell)

Системный монитор (Sysmon) – это Windows системная служба и драйвер устройства, который после установки в системе остается резидентом во время перезагрузки системы для отслеживания и регистрации действий системы в журнал событий Windows. Он предоставляет подробные сведения о создании процессов, сетевых подключениях и изменениях времен созданий файлов.

Скачать последнюю версию Sysmon можно с официального сайта Microsoft. После того как архив загрузится, его необходимо распаковать. Для дальнейшей работы необходимо зайти в командную строку и переместится в директорию куда распаковали архив с sysmon. Первым действием можно ввести команду `sysmon /?` для подробной информации об установке и удалении самого sysmon на ваш компьютер. Приступим к установке sysmon со всеми его компонентами. Для этого введем `sysmon.exe -i -h md5 -l -n`.

```
C:\Users\Администратор\Desktop\Sysmon>Sysmon.exe -i -h md5 -l -n

System Monitor v13.34 - System activity monitor
By Mark Russinovich and Thomas Garnier
Copyright (C) 2014-2022 Microsoft Corporation
Using libxml2. libxml2 is Copyright (C) 1998-2012 Daniel Veillard. All Rights Reserved.
Sysinternals - www.sysinternals.com

The driver SysmonDrv is already registered. Uninstall Sysmon before reinstalling.
```

Рис. 105. Установка Sysmon

В Windows Vista и более поздних версиях события хранятся в: Журналы приложений и служб /Microsoft /Windows /Sysmon /Operational. По своей структуре журнал sysmon похож на все остальные журналы Windows.

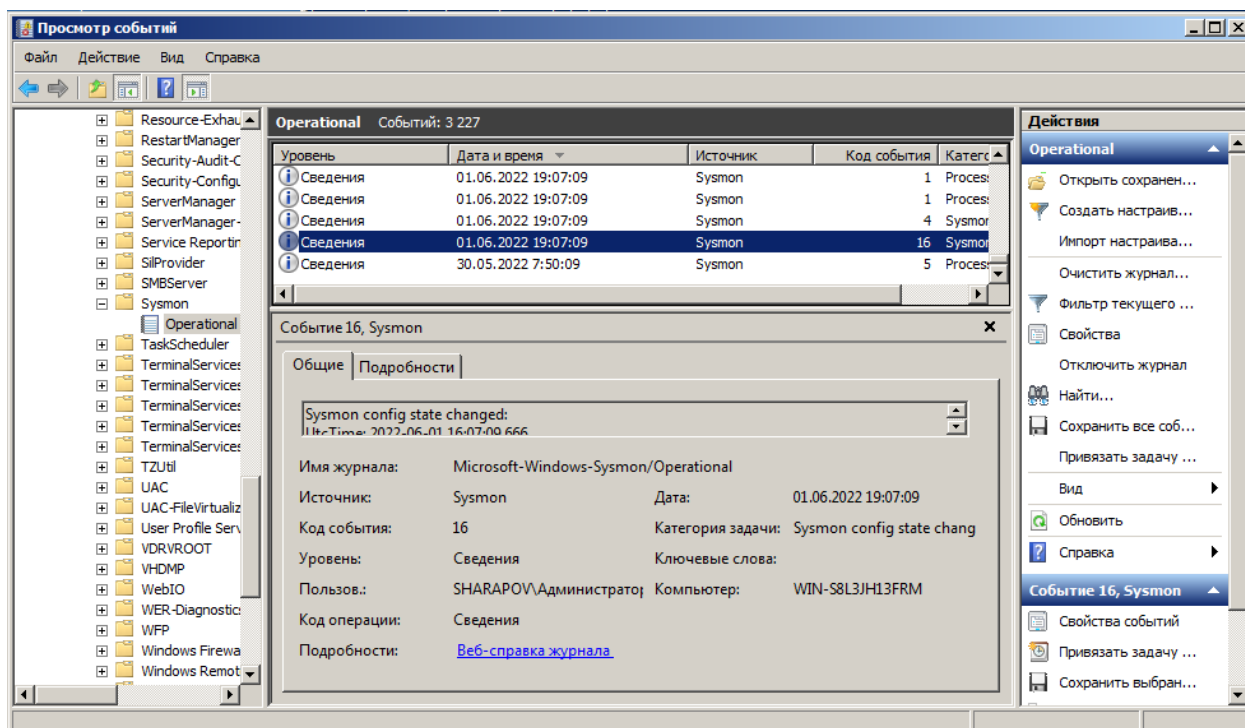


Рис. 106. Журнал Sysmon

Для примера рассмотрим пару событий:

1. Создание процесса (Process Create), код события – 1. В данном случае, смотря на рис. 107 можно понять, что в журнале отображается запуск консоли (mmc.exe).

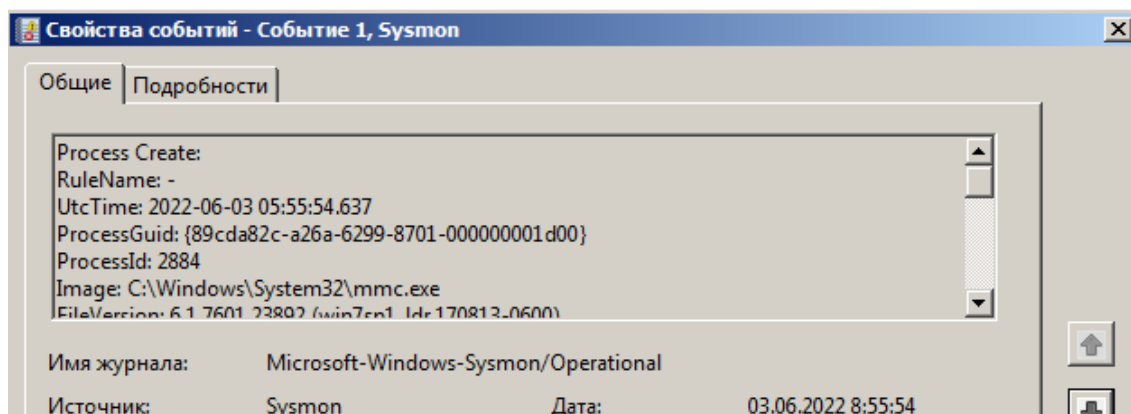


Рис. 107. Process Create

Завершение процесса (Process terminated), код события – 5. На рис. 108 показана запись о завершении работы с командной строкой (cmd.exe).

Posh-Sysmon – модуль PowerShell 3.0 или выше для создания файлов конфигурации Sysinternals Sysmon v2.0 и управления ими. Для загрузки данного модуля вводим `Install-Module -Name Posh-Sysmon`.

Выполнить

1. Введите команду на приведенном ниже рис. 108.

```
PS C:\Users\Администратор\Desktop\Sysmon> Install-Module -Name Posh-Sysmon
Ненадежный репозиторий
Идет установка модулей из ненадежного репозитория. Если вы доверяете этому репозиторию, измените его значение
InstallationPolicy, запустив командлет Set-PSRepository. Вы действительно хотите установить модули из "PSGallery"?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): Y
```

Рис. 108. Установка Posh-Sysmon

2. Чтобы посмотреть все возможные команды, которые добавляет данный модуль введём `Get-Command -Module Posh-Sysmon`

```
PS C:\Users\Администратор\Desktop\Sysmon> Get-Command -Module Posh-Sysmon
```

CommandType	Name	Version	Source
Function	ConvertFrom-SysmonBinaryConfiguration	1.2	Posh-Sysmon
Function	ConvertTo-SysmonXMLConfiguration	1.2	Posh-Sysmon
Function	Get-SysmonConfiguration	1.2	Posh-Sysmon
Function	Get-SysmonEventData	1.2	Posh-Sysmon
Function	Get-SysmonHashingAlgorithm	1.2	Posh-Sysmon

Рис. 109. Команды Posh-Sysmon

3. С модулем Posh-Sysmon так же можно просматривать определенные события, например, с определенным кодом события.

```
PS C:\Users\Администратор\Desktop\Sysmon> Get-SysMonEventData -EventId 1 -MaxEvents 10
```

EventId	: 1
EventType	: ProcessCreate
Computer	: WIN-S8L3JH13FRM.sharapov.ru
RuleName	: -
UtcTime	: 2022-06-03 07:44:36.981

Рис. 110. EventID (Sysmon)

Ещё один важный командлет – `New-SysmonConfiguration` он позволяет нам создать исходный файл конфигурации, в котором можем указать:

- а) алгоритм хеширования, поддерживаемые параметры;
- б) версию схемы (по умолчанию последняя версия 3.3);
- в) проверку отзыва сертификата для подписанных драйверов.

Командлет имеет только два обязательных параметра: путь для сохранения файла конфигурации и используемый `HashAlgorithm`. В следующем примере создадим новый файл конфигурации, включим все алгоритмы хеширования и установим правила для регистрации всех процессов создания и завершения.

```

PS C:\Users\Администратор\Desktop\Sysmon> New-SysmonConfiguration -Path .\sales_sysmon_config.xml -HashingAlgorithm
-ProcessCreate -ProcessTerminate -Verbose
VERBOSE: Enabling hashing algorithms : *
VERBOSE: Enabling logging all process creation by setting no filter and onmatch to exclude.
VERBOSE: Enabling logging all process termination by setting no filter and onmatch to exclude.
VERBOSE: Config file created as C:\Users\Администратор\Desktop\Sysmon\sales_sysmon_config.xml
VERBOSE: Configuration is for Sysmon 8.0

```

Рис. 111. Создание файла конфигурирования

Теперь можно убедиться, что sales_sysmon_config был задан правильно, с помощью команды Get-SysmonHashingAlgorithm и легко изменить алгоритмы хеширования, настроенные в файле конфигурации, с помощью командлета Set-SysmonHashingAlgorithm.

```

PS C:\Users\Администратор\Desktop\Sysmon> Get-SysmonHashingAlgorithm -Path .\sales_sysmon_config.xml
Hashing
-----
*

PS C:\Users\Администратор\Desktop\Sysmon> Set-SysmonHashingAlgorithm -Path .\sales_sysmon_config.xml -HashingAlgo
SHA1
PS C:\Users\Администратор\Desktop\Sysmon> Get-SysmonHashingAlgorithm -Path .\sales_sysmon_config.xml
Hashing
-----
SHA1

```

Рис. 112. Проверка работы хеширования и изменение алгоритмов хеширования

AdAudit – скрипт PowerShell для автоматизации аудита доменов, расположенный на крупнейшем веб-сервисе для хостинга IT-проектов – GtiHub.

Поскольку в данной работе мы используем Windows 2008, с версией PowerShell 2.0 – нам пришлось обновить PS до версии 5.1, поскольку в скрипте AdAudit присутствует модуль DSInternals, работающий на версии PowerShell, начиная с 5 версии.

На рис. 113 приведены команды установки самого модуля DSInternals.

```

PS C:\Users\Администратор\Desktop\adaudit-master> [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType]::Tls12
PS C:\Users\Администратор\Desktop\adaudit-master> Install-PackageProvider -Name NuGet -Force

Name                Version          Source            Summary
----                -
nuget                2.8.5.208       https://onege... NuGet provider for the OneGet meta-package manager

PS C:\Users\Администратор\Desktop\adaudit-master> if($null -eq (Get-PSRepository -Name PSGallery -ErrorAction SilentlyContinue)) { Register-PSRepository -Default }
PS C:\Users\Администратор\Desktop\adaudit-master> Install-Module -Name DSInternals -Force

```

Рис. 113. Установка DSInternals

Для примера введем аргумент при запуске hostdetails, который извлекает имя хоста и другую полезную информацию аудита.

```
PS C:\Users\Администратор\Desktop\adaudit-master> .\AdAudit -hostdetails
[+] Script start time 06/03/2022 12:08:51
[+] Outputting to C:\Users\Администратор\Desktop\adaudit-master\WIN-S8L3JH13FRM
[+] Lang specific variables
```

Рис. 114. Информация о hostdetails

Аргумент –All запускает все возможные проверки. В их состав входят пользователи и группы, созданные за последний месяц, политика паролей и т. д.

```
[+] Check for newly created users and groups
[!] 4 new users were created last 30 days, see C:\Users\Администратор\Desktop\adaudit-master\WIN-S8L3JH13FRM\new_users.txt
[!] 38 new groups were created last 30 days, see C:\Users\Администратор\Desktop\adaudit-master\WIN-S8L3JH13FRM\new_groups.txt
```

Рис. 115. Пользователи и группы, созданные за последний месяц

```
[+] Password Information Audit
[+] Checking default password policy
[!] Lockout threshold is less than 5, currently set to 0 (KB263)
[!] Minimum password length is less than 14, currently set to 7 (KB262)
[-] Finished checking default password policy
[+] Checking fine-grained password policies if they exist
[-] Finished checking fine-grained password policy
[!] Password quality test done, see C:\Users\Администратор\Desktop\adaudit-master\WIN-S8L3JH13FRM\password_quality.txt
```

Рис. 116. Информация о политике паролей

Данный скрипт весьма быстро предоставляет необходимую нам информацию. Отдельные команды выводятся на экран за пару секунд, а для аргумента -all нужно примерно полминуты.

С помощью политики аудита можно настроить параметры безопасности на сервере, чтобы предотвратить несанкционированный доступ к системе, защищенным файлам или папкам, журналу безопасности, а так же попытки изменить настройку сервера, удалить личные файлы или записи в журнале безопасности.

Хотя Windows обладает некоторыми собственными возможностями аудита, они не соответствуют потребностям большинства организаций. Собственные возможности аудита Windows недостаточно эффективны для анализа журналов и отката изменений. С другой стороны, решения для корпоративного аудита Windows могут обеспечивать аудит

изменений в реальном времени, мониторинг целостности файлов, упрощенный анализ журналов, точное восстановление данных, расширенные предупреждения и централизованные отчеты, а также многие другие функции, которые значительно упрощают администрирование, особенно, когда дело доходит до отмены изменений и предоставления быстрой, чистой информации аудиторам.

Правильно организованный аудит Windows помогает организациям соблюдать требования защиты данных, выявлять потенциальные угрозы (например, нежелательные изменения) на ранних этапах и помогает снизить риск взлома данных. Часто инструменты аудита и безопасности Windows также позволяют откатить изменения в более раннюю, более желательную конфигурацию.

Администратор обязан проводить регулярный аудит безопасности ОС, осуществлять мониторинг всех систем ОС на каждом компьютере. Аудит безопасности в операционных системах – это постоянный мониторинг событий, связанных с нарушениями безопасности, контроль, проверка их с целью своевременного выявления нарушений политики безопасности, а также попыток взлома.

2.4. Аудит событий безопасности FreeBSD

Для обеспечения безопасности данных и сети в целом, а также для регистрации событий системы, обнаружения вторжений и анализа событий существует аудит событий безопасности. Аудит позволяет выполнять гибко настраиваемое протоколирование различных событий. Данный аудит является встроенным в систему [11].

Таблица 5

Классы событий системы аудита

Имя класса	Расшифровка	Действие
all	all	Соответствует всем классам событий
aa	authentication and authorization	Аудит аутентификации и авторизации
ad	administrative	Аудит административных действий, произошедших в системе
ap	application	События, определяемые каким-либо приложением
cl	file close	Аудит вызовов системной функции close

Имя класса	Расшифровка	Действие
ex	exec	Аудит запуска приложения. Аудит аргументов командной строки и переменных окружения контролируется через audit_control, используя параметры argv и envv в опции policy
fa	file attribute access	Аудит доступа к атрибутам объектов
fc	file create	Аудит событий, в результате которых создаются файлы
fd	file delete	Аудит событий, в результате которых удаляются файлы
fm	file attribute modify	Аудит событий, в результате которых изменяются атрибуты файлов
fr	file read	Аудит событий, в результате которых происходит чтение данных или открываются файлы на чтение
fw	file write	Аудит событий, в результате которых происходит запись данных, запись или изменение файлов
io	ioctl	Аудит вызовов системной функции ioctl.
ip	ipc	Аудит различных видов взаимодействия процессов, включая создание неименованных каналов (POSIX pipe) и взаимодействие процессов в стиле System V IPC
lo	login_logout	Аудит событий login и logout
na	non attributable	Аудит неприписываемых событий
no	invalid class	Не соответствует никаким событиям аудита
nt	network	Аудит событий, связанных с сетевыми подключениями
ot	other	Аудит различных событий
pc	process	Аудит действий процессов

К каждому классу аудита можно добавить префикс, который обозначает операцию (она будет учитываться), а также то, включает ли данная запись аудит для данного класса и типа, либо отключает его. Если аудит используется без префикса, то по умолчанию аудиту подлежат и успешные, и ошибочные события.

Префиксы классов аудита событий

Префикс	Действие
+	Аудит успешных событий в данном классе.
-	Аудит ошибочных событий в данном классе.
^	Отключение аудита как успешных, так и ошибочных событий в данном классе.
^+	Отключение аудита успешных событий в данном классе.
^-	Отключение аудита ошибочных событий в данном классе.

Параметры и признаки, относящиеся к аудиту событий безопасности [12]:

а) событие: событие, которое может быть занесено в журнал (для user) и не приписываемое (до аутентификации субъекта доступа);

б) класс: именованные наборы одноподобных событий, которые используются в выражениях выбора;

в) запись: единичная запись в журнале, которая описывает то или иное событие;

г) журнал: файл, который содержит последовательность записей аудита, описывающих события безопасности;

д) выражение выбора: строка, содержащая список префиксов и имен классов, используемая для выбора группы событий;

е) предварительный выбор: процесс, с помощью которого система определяет, какие события имеют важность для администратора (классы событий, глобальные настройки);

ё) фильтрация: процесс, в результате которого записи из существующего журнала выделяются для хранения, анализа или распечатки.

Рассмотрим файлы системы аудита. При подключении аудита в каталоге `/etc/security` будут храниться конфигурационные файлы системы аудита, такие как:

– `audit_class` – содержит определения классов аудита;

– `audit_control` – контролирует некоторые аспекты системы аудита: классы по умолчанию, минимальное дисковое пространство, которое должно оставаться на разделе журнала аудита, максимальный размер журнала аудита;

– `audit_event` – связывает идентификаторы событий с их текстовыми именами, описаниями и классами событий;

– `audit_user` – уточняет настройки аудита для конкретных пользователей;

– `audit_warn` – настраиваемый скрипт командного интерпретатора, который вызывается `auditd` для генерации предупреждений в исключительных ситуациях.

Журналы аудита находятся в каталоге `/var/audit/` и хранятся в бинарном формате `BSM`. Для перевода в текстовый формат, либо для изменения необходимо использовать утилиту `praudit`, которая является встроенной. Помимо `praudit` существует утилита `auditreduce`, которая фильтрует журнальные записи для анализа, распечатки или же архивации. Журнал управляется с помощью демона аудита `auditd` и пишется ядром.

Для обеспечения информационной безопасности при использовании операционной системы `FreeBSD` нужно использовать аудит безопасности.

Для этого необходимо следующее:

1. Установить аудит безопасности, а для этого нужно:

– отредактировать файл `/etc/rc.conf`;

– запустить демон аудита.

2. Установить необходимые дополнительные требования к аудиту в файле `audit_control`.

3. Установить требования к аудиту для конкретных пользователей путём добавления их в файл `audit_user`.

После этих действий в системе будет происходить журналирование определённых действий пользователей, которое зависит от выбранных классов событий для конкретного пользователя и системы в целом. Для работы с аудитом безопасности сначала необходимо было выбрать конкретную версию `FreeBSD`, нами была выбрана `FreeBSD 12.0`. Так как данный аудит входит в базовую систему, то его необходимо только активировать. Активация аудита происходит путём добавления в `/etc/rc.conf` строки `auditd_enable="YES"`. Редактировать `/etc/rc.conf` можно с помощью встроенного редактора `ee`.

После того, как была введена команда `ee /etc/rc.conf`, откроется текстовый редактор, в котором необходимо дописать строку `auditd_enable="YES"` для активации аудита безопасности. Перед тем, как работать с `/etc/rc.conf` желательно создать резервную копию данного файла.

Теперь можно переходить к редактированию файла. Для того, чтобы вернуть исходные данные в `rc.conf` из резервного файла необходимо использовать команду `mv: mv /etc/rc.conf.orig /etc/rc/ conf`.

Выполнить

1. Приступим к редактированию файла с целью активации аудита безопасности. Введите команду на приведенном ниже рис. 117.

```
====line 1 col 0 lines from top 1 =====  
sendmail_enable="NONE"  
hostname="dallas"  
keymap="ru.kbd"  
ifconfig_em0="DHCP"  
sshd_enable="YES"  
ntpd_enable="YES"  
powerd_enable="YES"  
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable  
dumpdev="AUTO"  
auditd_enable="YES"
```

Рис. 117. Активация аудита в `/etc/rc.conf` с помощью `ee`

Альтернативой редактору `ee` может послужить `vi`.

```
clear_tmp_enable="YES"  
syslogd_flags="-ss"  
sendmail_enable="NONE"  
hostname="dallas.loc"  
keymap="ru.kbd"  
ifconfig_em0="inet 192.168.1.100 netmask 255.255.255.0"  
defaultrouter="192.168.1.1"  
sshd_enable="YES"  
ntpd_enable="YES"  
powerd_enable="YES"  
# Set dumpdev to "AUTO" to enable crash dumps, "NO" to disable  
dumpdev="NO"  
auditd_enable="YES"
```

Рис. 118. Активация аудита в `/etc/rc.conf` с помощью `vi`

2. После активации аудита необходимо запустить демон аудита `auditd` путём ввода команды `# service auditd start`.

Если используется специализированное ядро системы FreeBSD, то необходимо включить запись `options AUDIT` в конфигурацию ядра.

Конфигурационные файлы системы аудита хранятся в каталоге `/etc/security`.

Файл `audit_class` содержит определения классов аудита.


```

#
# $FreeBSD: releng/12.0/contrib/openbsm/etc/audit_class 292432 2015-12-18 09:48:
01Z brueffer $
#
0x00000000:no:invalid class
0x00000001:fr:file read
0x00000002:fw:file write
0x00000004:fa:file attribute access
0x00000008:fm:file attribute modify
0x00000010:fc:file create
0x00000020:fd:file delete
0x00000040:cl:file close
0x00000080:pc:process
0x00000100:nt:network
0x00000200:ip:ipc
0x00000400:na:non attributable
0x00000800:ad:administrative
0x00001000:lo:login_logout
0x00002000:aa:authentication and authorization
0x00004000:ap:application
0x20000000:io:ioctl
0x40000000:ex:exec
0x80000000:ot:miscellaneous
0xffffffff:all:all flags set
/etc/security/audit_class (END)

```

Рис. 119. Содержимое audit_class

Файл audit_control контролирует некоторые аспекты системы аудита. Содержит общие настройки системы аудита.

```

#
# $FreeBSD: releng/12.0/contrib/openbsm/etc/audit_control 292432 2015-12-18 09:4
8:01Z brueffer $
#
dir:/var/audit
dist:off
flags:lo,aa
minfree:5
naflags:lo,aa
policy:cnt,argv
filesz:2M
expire-after:10M
/etc/security/audit_control (END)

```

Рис. 120. Содержимое audit_control

Записи файла audit_control:

- dir – устанавливает каталог хранения журнала.

Примечание: если указано больше 1 каталога, то каталоги будут использоваться по мере заполнения;

- dist – включение жестких ссылок для всех журналов аудита;

- flags – установка глобальной маски предварительного выбора для приписываемых событий;

- minfree – определение минимального количества свободного дискового пространства на разделе сохранения файлов журнала аудита;

- naflags – определение классов аудита для неприписываемых событий;
- policy – определение разделяемых запятыми списка флагов политики, определяющей различные аспекты поведения аудита;
- filesz – определение максимального размера журнала событий аудита, после достижения которого журнал будет закончен и подвергнут ротации;
- expire-after – определение момента времени, после которого журнальные файлы удаляются.

Файл `audit_event` связывает идентификаторы событий с их текстовыми именами, описаниями и классами событий.

```
#
# $FreeBSD: releng/12.0/contrib/openbsm/etc/audit_event 316006 2017-03-26 21:14:
49Z rwatson $
#
# The mapping between event identifiers and values is also hard-coded in
# audit_kevents.h and audit_uevents.h, so changes must occur in both places,
# and programs, such as the kernel, may need to be recompiled to recognize
# those changes. It is advisable not to change the numbering or naming of
# kernel audit events.
#
# Allocation of BSM event identifier ranges:
#
# 0                Reserved and invalid
# 1 - 2047         Reserved for Solaris kernel events
# 2048 - 5999     Reserved and unallocated
# 6000 - 9999     Reserved for Solaris user events
# 10000 - 32767   Reserved and unallocated
# 32768 - 65535   Available for third party applications
#
# Of the third party range, OpenBSM allocates from the following ranges:
#
# 43000 - 44999   Reserved for OpenBSM kernel events
# 45000 - 46999   Reserved for OpenBSM application events
#
/etc/security/audit_event
```

Рис. 121. Содержимое `audit_event`

Файл `audit_user` уточняет настройки аудита для конкретных пользователей. Файл `audit_warn` является настраиваемым скриптом командного интерпретатора, который вызывается `auditd` для генерации предупреждений в исключительных ситуациях.

Для изменения и просмотра в текстовом формате журнала аудита необходимо использовать утилиту `praudit`, так как он хранится в бинарном формате BSM.

```
root@dallas:~ # praudit /var/audit/  
20200520222804.20200520222841. current@  
20200522002032.crash_recovery. dist/  
20200522002724.20200522021850. remote/  
20200522021850.not_terminated.
```

Рис. 122. Список журналов аудита

Для проверки содержимого журнала необходимо использовать:
praudit /var/audit/*, где * – имя необходимого журнала.

```
root@dallas:~ # praudit /var/audit/20200520222804.20200520222841.  
header,56,11,audit startup,0,Wed May 20 22:28:04 2020, + 11 msec  
text,auditd::Audit startup  
return,success,0  
trailer,56  
header,68,11,logout - local,0,Wed May 20 22:28:41 2020, + 339 msec  
subject,root,root,wheel,root,0,763,763,0,0.0.0.0  
return,success,0  
trailer,68  
header,57,11,audit shutdown,0,Wed May 20 22:28:41 2020, + 548 msec  
text,auditd::Audit shutdown  
return,success,0  
trailer,57
```

Рис. 123. Содержание журнала аудита

Помимо общей настройки аудита для всех пользователей, можно точно настраивать записи определённых событий для какого-либо конкретного пользователя. Для этого необходимо изменять `audit_user`.

Так как в системе имеется только один пользователь – `root`, то необходимо создать ещё одного пользователя и настроить аудит для него.

Выполнить

Введите команду на приведенном ниже рис. 124. Создадим нового пользователя утилитой `adduser`.

```
root@dallas:~ # adduser  
Username: dallas  
Full name: dallas  
Uid (Leave empty for default):  
Login group [dallas]:  
Login group is dallas. Invite dallas into other groups? []:  
Login class [default]:  
Shell (sh csh tcsh nologin) [sh]:  
Home directory [/home/dallas]:  
Home directory permissions (Leave empty for default):  
Use password-based authentication? [yes]: no
```

Рис. 124. Создание нового пользователя

- При создании пользователя необходимо указать:
- Username – имя пользователя;
 - Full name – полное имя пользователя;
 - Uid – идентификатор пользователя (используется для однозначной идентификации в системе);
 - Login group – группа пользователя;
 - Invite to other groups – приглашение пользователя к другим группам (“yes” / “no”);
 - Login class – класс доступ (более гибкое приспособление системы для различных пользователей);
 - Shell – оболочка (по умолчанию выбирается sh);
 - Home directory – домашний каталог;
 - Home directory permissions – права доступа к домашнему каталогу;
 - Use password-based authentication – использование аутентификации, основанной на пароле (если прописать “yes”, то необходимо будет ввести пароль);
 - Use an empty password – использование пустого пароля (“yes” / “no”);
 - Use a random password – использование случайно сгенерированного пароля (“yes” / “no”);
 - Enter password – ввод пароля;
 - Enter password again – повторный ввод пароля;
 - Lock out the account after creation – блокировка пользователя после создания (“yes” / “no”).

После заполнения всех полей система выведет все параметры для проверки и спросит верно ли введены данные.

Подтвердив верность введённых данных, система предложит создание ещё одного пользователя. Помимо утилиты создания пользователя существует утилита удаления пользователя из системы – `rmuser`.

```
root@dallas:~ # rmuser
Please enter one or more usernames: dallas
Matching password entry:
dallas::1001:1001::0:0:dallas:/home/dallas:/bin/sh
Is this the entry you wish to remove? █
```

Рис. 125. Пример использования утилиты `rmuser`

Если необходимо изменить какую-либо информацию о пользователе, то необходимо воспользоваться утилитой `chpass`.

```

#Changing user information for dallas.
Login: dallas
Password:
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/dallas
Shell: /bin/sh
Full Name: dallas
Office Location:
Office Phone:
Home Phone:
Other information:
~

```

Рис. 126. Пример использования утилиты `chpass`

Пользователь может менять только свою информацию. Исключением является суперпользователь, который может менять информацию других пользователей.

Перейдём к изменению аудита для созданного пользователя.

Выполнить

Установим для пользователя `dallas` несколько аудитов, таких как:

- а) `lo` (аудит событий `login` и `logout`);
- б) `+ex` (аудит успешных событий запуска приложения);
- в) `+fc` (аудит успешных событий в результате которых создаются файлы);
- г) `+fd` (аудит успешных событий в результате которых удаляются файлы);
- д) `+fw` (аудит успешных событий в результате которых происходит запись данных, запись или изменение файлов);
- е) `-fr` (аудит ошибочных событий, в результате которых файлы не открылись для чтения).

Для этого необходимо добавить строку в файл `audit_user` с именем пользователя и аудитами.

```

#
# $FreeBSD: releng/12.0/contrib/openbsm/etc/audit_user 292432 2015-12-18 09:48:0
#
root:lo:no
dallas:lo,+ex,+fc,+fd,+fw,-fr:no

```

Рис. 127. Добавление дополнительных требований к аудиту для пользователя `dallas`

Лог-файлы системы могут иметь довольно большой размер и найти какую-либо необходимую информацию может быть затруднительно. В таких ситуациях необходимо использовать утилиту `auditreduce`.

Выполнить

Введите команду на приведенном ниже рис. 128.

```
root@dallas:~ # auditreduce -u dallas /var/audit/202005222002032.crash_recovery. | praudit
```

Рис. 128. Использование команды `auditreduce`

Так как пользователь `dallas` был недавно создан и не проявлял активности, то журнал аудита пустой.

```
root@dallas:~ # auditreduce -u root /var/audit/20200522221903.20200522222009. | praudit
header,68,11,login - local,0,Fri May 22 22:19:06 2020, + 689 msec
subject,root,root,wheel,root,0,661,661,0,0.0.0.0
return,success,0
trailer,68
header,68,11,logout - local,0,Fri May 22 22:20:09 2020, + 849 msec
subject,root,root,0,root,0,661,661,0,0.0.0.0
return,success,0
trailer,68
```

Рис. 129. Пример использования утилиты `auditreduce`

Чтобы убедиться в том, что утилита `auditreduce` работает правильно, требуется заменить пользователя `dallas` на `root`. В итоге на экран вывелась информация из журнала аудита, связанная только с пользователем `root`.

Из-за того, что файлы журнала аудита могут быть очень больших размеров, то в некоторых случаях появляется необходимость сжатия сразу после закрытия их демоном аудита.

Для выполнения сжатия журнала используется скрипт `audit_warn`, чтобы выполнять сжатие журнала сразу после закрытия необходимо добавить несколько строк в файл `audit_warn`.

Также возможен мониторинг системы в реальном времени с использованием потоков аудита. Для этого необходимо выполнить `# praudit /dev/auditpipe`

Выполнить

Введите команду: `praudit /dev/auditpipe`.

По умолчанию, в системе потоки доступны только root-пользователю. Чтобы предоставить доступ для группы audit необходимо добавить правило devfs в `/etc/devfs.rules`.

Выполнить

Введите команду предоставления доступа к потокам членам группы audit: `add path `auditpipe*' mode 0440 group audit`.

Теперь пользователи, которые находятся в группе audit имеют доступ к потокам. Прделанная работа поможет контролировать не только вход и выход в систему, но и получать данные об успешных или ошибочных событиях. Примером событий может быть открытие файла, его редактирование, удаление и так далее.

Для управления аудитом существует утилита для выборочной проверки журнала, с помощью которой можно выделить действия определённого пользователя, для этого необходимо было установить дополнительные требования к аудиту.

Аудит безопасности во FreeBSD крайне необходим для предотвращения утечки данных, а также для регистрации событий системы.

2.5. Изучение работы инструмента hping3, организация стресс-теста сети с его помощью. Настройка брандмауэра с помощью ufw и iptables

Hping3 – это инструмент для создания и анализа пакетов ICMP или TCP. В дополнение к обнаружению пробелов в безопасности вы также можете использовать команду для проверки сетей и хостов на их функциональность. Анализатор пакетов TCP/IP адаптирован к командной строке на основе команды Ping Unix и даже имеет режим traceroute. Его также можно использовать для отправки файлов между фактически скрытыми соединениями. Другие возможные применения включают в себя тесты межсетевого экрана, расширенное сканирование портов, сетевые тесты с различными протоколами, фрагментацию или проверку стеков TCP/IP. С помощью hping начинающие и профессионалы получают полное представление о мире TCP/IP. Инструмент работает во всех Unix-подобных системах, таких как Linux, FreeBSD, NetBSD, OpenBSD и Solaris.

В то время как `hping` использовался в прошлом как инструмент безопасности, он может использоваться многими способами людьми, которые не заботятся о безопасности для тестирования сетей и хостов.

`Hping3` по умолчанию (без параметров) отправляет нулевой пакет с заголовком TCP на порт 0.

Вы можете выбрать использование другого протокола с помощью числовой опции, доступной для каждого из них:

- 0 (режим Raw IP);
- 1 (режим ICMP);
- 2 (режим UDP);
- 8 (режим сканирования);
- 9 (режим прослушивания).

Поскольку `hping3` использует TCP по умолчанию, отсутствие указанных ниже параметров отправит сегмент TCP.

При использовании TCP можем решить либо опустить флаги (по умолчанию), либо установить флаг, используя один из следующих параметров:

- а) -S (SYN);
- б) -A (ACK);
- в) -R (RST);
- г) -F (FIN);
- д) -P (PUSH);
- е) -U (URG);
- ё) -X (XMAS);
- ж) -Y (YMAS).

В Linux-системах для настройки брандмауэра чаще всего используются две утилиты: `Ufw` и `Iptables`.

`UFW` (Uncomplicated Firewall) – является самым простым и довольно популярным инструментарием командной строки для настройки и управления брандмауэром в дистрибутивах `Ubuntu` и `Debian`. Правильно функционирующий брандмауэр является наиболее важной частью обеспечения полной безопасности системы `Linux`.

`Iptables` – это брандмауэр для `Linux`, используемый для мониторинга входящего и исходящего трафика и его фильтрации в соответствии с заданными пользователем правилами для исключения несанкционированного доступа к системе. При помощи `Iptables` можно разрешить на вашем сервере движение только определенного трафика. В данном руководстве рассмотрим использование `Iptables` для обеспечения безопасности веб-приложения.

Все данные передаются по сети в виде пакетов. Ядро Linux предоставляет интерфейс для фильтрации пакетов входящего и исходящего трафика при помощи специальных таблиц. Iptables – это приложение командной строки и межсетевой экран для Linux, которым можно пользоваться для создания, поддержания работоспособности и проверки этих таблиц.

Можно создать несколько таблиц. В каждой таблице может содержаться несколько цепочек. Цепочка – это набор правил. Каждое правило определяет, что делать с пакетом, если он соответствует условиям. При соответствии пакета над ним выполняется целевое действие (TARGET). Это может быть проверка следующей цепочкой или один из следующих вариантов:

1. ACCEPT: разрешить передачу пакета.
2. DROP: запретить передачу пакета.
3. RETURN: пропустить текущую цепочку и перейти к следующему правилу в цепочке, которая ее вызвала.

4. В данном руководстве будем работать с одной из таблиц по умолчанию, которая называется фильтром (filter). В таблице фильтра есть три цепочки (набора правил):

5. INPUT – используется для контроля входящих пакетов. Можно разрешать или блокировать подключения по порту, протоколу или IP-адресу источника.

6. FORWARD – используется для фильтрации пакетов, приходящих на сервер, но перенаправляемых куда-либо еще.

7. OUTPUT – используется для фильтрации исходящих пакетов.

Помимо Nping существуют и другие инструменты, связанные с формированием пакетов, эхо-запросов и сканированием портов:

1) ping: проверяет end-to-end соединение (задержку времени приёма передачи (RTT), флуктуации, потерю пакетов) удалённого хоста запросами ICMP echo/reply. Полезна для проверки статуса и доступности;

2) traceroute: определяет путь третьего уровня переадресации из локального хоста к удалённому конечному хосту посредством пакетов зондирования ICMP/UDP/TCP с ограниченным временем жизни (TTL). Полезна для решения таких сетевых проблем как доступность и роутинг;

3) mtr: вариант traceroute который комбинирует функциональность traceroute и ping и отображает статистику. Полезен для характеристики задержек роутинга по всему пути;

4) netcat/socat: это швейцарский нож для TCP/IP сети, позволяет читать/записывать поток байтов на TCP/UDP. Полезен для решения проблем с политиками файервола и доступностью служб;

5) dig: инструмент по решению DNS-проблем, который может генерировать прямые запросы, обратные запросы, находить авторитетные сервера имён, проверять CNAME, MX и другие DNS-записи. Можно указать для запросов конкретный DNS-сервер по вашему выбору;

6) nslookup: другой инструмент проверки/решения проблем с DNS. Работает со всеми DNS запросами и записями. Может делать запросы к конкретному DNS-серверу;

7) dnsyo: инструмент тестирования DNS, который проверяет пространство DNS, выполняя поиск DNS из ряда открытых резолверов, расположенных в 1500 различных сетях по всему миру;

8) lsof: показывает информацию о файлах (например, обычные файлы, трубы или сокеты), которые открыты процессами. Полезно для монитора процессов или пользователей с точки зрения их открытых сетевых соединений или открытых файлов;

9) iftop: основанная на ncurses утилита с основанным на тексте интерфейсом (TUI), которая может быть использована для монитора в реальном времени использования полосы пропускания и сетевых соединений для отдельных сетевых интерфейсов. Полезна для отслеживания пропускной способности загруженных приложений, пользователей, мест назначений и портов;

10) netstat: утилита статистики сети, которая может показать информацию о статусе и статистику об открытых сетевых соединениях (TCP/UDP портах, IP-адресах), таблицах роутинга, переданного/полученного трафика и протоколах. Полезна для связанных с сетью диагностик и тонкой настройки производительности;

11) tcpdump: сниффер пакетов, основанный на библиотеке по захвату пакетов libpcap. Может определять фильтры по захвату пакетов в формате Berkeley Packet Filters;

12) tshark: ещё одна программа-сниффер пакетов командной строки с полной совместимостью со своей коллегой с графическим интерфейсом Wireshark. Поддерживает 1000 протоколов и этот список продолжает расти. Полезна для решения проблем, анализа и сохранения информации по живым пакетам;

13) ip: разносторонняя утилита командной строки, которая является частью пакета iproute2. Полезна для проверки и модификации таблиц маршрутизации, состояний сетевых устройств и настроек IP туннелирования. Полезна для просмотра таблиц маршрутизации, добавления/удаления статичных маршрутов, настройке сетевых интерфейсов и других функций по решению проблем с маршрутизацией;

14) ifup/ifdown: используется для поднятия или отключения конкретного сетевого интерфейса. Часто предпочтительней альтернативе по перезапуску всего сетевого устройства;

15) autossh: это программа, которая создаёт SSH сессии и автоматически перезапускает сессии, которые должны быть разъединены. Часто полезна для создания постоянного реверсного SSH туннеля между ограниченными корпоративными сетями.

Для начала необходимо ознакомиться с основными командами и функциями в hping3. Воспользуемся справкой по данному инструменту (в качестве целевого компьютера использовалась виртуальная машина на основе Windows: #hping3 -help).

1. Простой запрос пинг(ICMP):

#hping3 -l -c 10 целевой IP-адрес.

```
nastya@kali:~$ sudo hping3 -l -c 10 192.168.1.101
[sudo] пароль для nastya:
HPING 192.168.1.101 (eth1 192.168.1.101): icmp mode set, 28 headers + 0 data bytes
len=46 ip=192.168.1.101 ttl=128 id=165 icmp_seq=0 rtt=21.7 ms
len=46 ip=192.168.1.101 ttl=128 id=166 icmp_seq=1 rtt=5.8 ms
len=46 ip=192.168.1.101 ttl=128 id=167 icmp_seq=2 rtt=3.9 ms
len=46 ip=192.168.1.101 ttl=128 id=168 icmp_seq=3 rtt=3.9 ms
```

Рис. 130. Запрос пинг

В данной команде -l – означает, что работаем в режиме ICMP.

В -c 10 говорится, что хотим отправить 10 пакетов, а 192.168.1.101 – наша цель.

Выполнить

1. Введите команду:

#hping3 -S -c 3 -s 5151 целевой IP-адрес.

Здесь -S отмечает флаг SYN в нашем заголовке TCP. также видим здесь новый вариант: -s 5151, который выбирает порт источника для использования. Без этой опции hping3 просто выбирает случайный порт источника. Поскольку порт 0 не открыт, видим ответ RST-ACK (отмеченный на выходе).

2. Отправьте идентичный пакет, за исключением этого времени с установленным флагом FIN:

#hping3 -F -c 1 -c 5151 целевой IP-адрес.

```
nastya@kali:~$ sudo hping3 -F -c 1 -c 5151 192.168.1.101
HPING 192.168.1.101 (eth1 192.168.1.101): F set, 40 headers + 0 data bytes
len=46 ip=192.168.1.101 ttl=128 id=198 sport=0 flags=RA seq=0 win=0 rtt=13.9 ms
len=46 ip=192.168.1.101 ttl=128 id=199 sport=0 flags=RA seq=1 win=0 rtt=3.8 ms
len=46 ip=192.168.1.101 ttl=128 id=200 sport=0 flags=RA seq=2 win=0 rtt=3.0 ms
len=46 ip=192.168.1.101 ttl=128 id=201 sport=0 flags=RA seq=3 win=0 rtt=2.9 ms
len=46 ip=192.168.1.101 ttl=128 id=202 sport=0 flags=RA seq=4 win=0 rtt=1.0 ms
^C
— 192.168.1.101 hping statistic —
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.0/4.9/13.9 ms
nastya@kali:~$
```

Рис. 131. FIN-запрос

В предыдущих примерах отправляли пакеты в порт 0. Теперь проверим известный порт, порт 80 (http). Поскольку SYN является первым шагом в трехстороннем рукопожатии TCP-соединения (SYN, SYN-ACK, ACK), если порт открыт, получим правильный ответ SYN-ACK из-за того, что цель пытается завершить соединение, это популярный метод, используемый при сканировании портов, известный как «полуоткрытое соединение»

3. Отправьте пакет: `#hping3 -S -c 1 -s 5151 -p 80` целевой IP-адрес.

```
nastya@kali:~$ sudo hping3 -S -c 3 -s 5151 -p 80 192.168.1.101
HPING 192.168.1.101 (eth1 192.168.1.101): S set, 40 headers + 0 data bytes
len=46 ip=192.168.1.101 ttl=128 id=205 sport=80 flags=RA seq=0 win=0 rtt=20.3 ms
len=46 ip=192.168.1.101 ttl=128 id=206 sport=80 flags=RA seq=1 win=0 rtt=3.8 ms
len=46 ip=192.168.1.101 ttl=128 id=207 sport=80 flags=RA seq=2 win=0 rtt=2.1 ms
```

Рис. 132. Запрос SYN на порт 80

Здесь `-p 80` указывает порт назначения, который будет установлен в нашем заголовке TCP.

4. Отправьте пакет: `#hping3 -A -c 3 -s 5151 -p 80` целевой IP-адрес.

```
nastya@kali:~$ sudo hping3 -A -c 3 -s 5151 -p 80 192.168.1.101
HPING 192.168.1.101 (eth1 192.168.1.101): A set, 40 headers + 0 data bytes
len=46 ip=192.168.1.101 ttl=128 id=208 sport=80 flags=R seq=0 win=0 rtt=45.1 ms
len=46 ip=192.168.1.101 ttl=128 id=209 sport=80 flags=R seq=1 win=0 rtt=4.0 ms
len=46 ip=192.168.1.101 ttl=128 id=210 sport=80 flags=R seq=2 win=0 rtt=2.8 ms

--- 192.168.1.101 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 2.8/17.3/45.1 ms
```

Рис. 133. Запрос ACK на порт 80

Хост-цель ответил, но на этот раз с установленным флагом RST.

5. Отправьте пакет (UDP, порт 80): `#hping3 -2 -c 1 -s 5151 -p 80` целевой IP-адрес.

```
nastya@kali:~$ sudo hping3 -2 -c 3 -s 5151 -p 80 192.168.1.101
HPING 192.168.1.101 (eth1 192.168.1.101): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.1.101 name=UNKNOWN
status=0 port=5151 seq=0
ICMP Port Unreachable from ip=192.168.1.101 name=UNKNOWN
status=0 port=5152 seq=1
ICMP Port Unreachable from ip=192.168.1.101 name=UNKNOWN
status=0 port=5153 seq=2

--- 192.168.1.101 hping statistic ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 2.9/9.0/17.0 ms
```

Рис. 134. Запрос UDP на порт 80

Используя команду «-2» в этой команде, указываем использование UDP в качестве нашего протокола транспортного уровня. На выходе видим, что получили сообщение ICMP Port Unreachable из-за того, что этот порт не открыт для трафика UDP.

6. Отправьте пакет, чтобы реализовать сканирование портов:

#hping3 -8 50-81 -S целевой IP-адрес -v.

```
nastya@kali:~$ sudo hping3 -8 50-81 192.168.1.101 -V
using eth1, addr: 192.168.1.103, MTU: 1500
Scanning 192.168.1.101 (192.168.1.101), port 50-81
32 ports to scan, use -V to see all the replies
+-----+-----+-----+-----+
|port| serv name | flags |ttl| id | win | len |
+-----+-----+-----+-----+
 50      : ..R.A... 128  513  0  46
 51      : ..R.A... 128  769  0  46
 52      : ..R.A... 128  1025 0  46
```

Рис. 135. Сканирование 50-81 портов

В нашем случае произведенное сканирование показывает, что все порты открыты, так как имеют значение флага flags=RA. Удаленное сканирование портов позволяет определять, какие порты хоста открыты для передачи трафика.

Здесь -v – опция, позволяющая выводить таблицу с данными по портам.

Hping позволяет реализовать несколько способов сбора ISN (число, используемое при нумерации дейтаграмм при установлении IP/TCP-соединения) и определения их приращений (ОС Фингерпринтинг). Самый простой – использовать -Q.

Выполнить

Введите команду:

#hping3 целевой IP-адрес -p 139 -Q -S.

```
nastya@kali:~$ sudo hping3 192.168.1.101 -p 139 -Q -S
HPING 192.168.1.101 (eth1 192.168.1.101): S set, 40 headers + 0 data bytes
 592762815 +592762815
2025745520 +1432982705
2162178635 +136433115
1678786371 +3811575031
4161830885 +2483044514
 88178274 +221314684
 294321076 +206142802
1083451665 +789130589
2664818076 +1581366411
2070616854 +3700766073
3155280480 +1084663626
1604088550 +2743775365
3487510166 +1883421616
^C
— 192.168.1.101 hping statistic —
13 packets transmitted, 13 packets received, 0% packet loss
round-trip min/avg/max = 1.8/5.7/16.3 ms
```

Рис. 136. Сбор и определение их приращений

Фаззинг (англ. fuzzing) – это способ тестирования, в основе которого лежит передача некорректных, случайных или непредвиденных логикой программы данных.

Базовую трассировку UDP можно эмулировать с помощью следующей команды:

```
#hping3 -2 целевой IP-адрес -p 4444 -T -n
```

```
nastya@kali:~$ sudo hping3 -2 192.168.1.101 -p 4444 -T -n
HPING 192.168.1.101 (eth1 192.168.1.101): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.1.101
status=0 port=1921 seq=0
ICMP Port Unreachable from ip=192.168.1.101
status=0 port=1922 seq=1
```

Рис. 137. Трассировка UDP

Чтобы наблюдать, как меняется маршрут при разных прыжках, можем зафиксировать значение TTL. Например, при пинге в Google:

```
#hping3 -S 64.233.167.99 -p 80 -T -ttl 15 -tr-keep-ttl -n
```

```
nastya@kali:~$ sudo hping3 -S 64.233.167.99 -p 80 -T -ttl 15 -tr-keep-ttl -n
HPING 64.233.167.99 (eth0 64.233.167.99): S set, 40 headers + 0 data bytes
len=46 ip=64.233.167.99 ttl=64 id=15465 sport=80 flags=SA seq=0 win=65535 rtt=157.6 ms
len=46 ip=64.233.167.99 ttl=64 id=15466 sport=80 flags=SA seq=1 win=65535 rtt=125.0 ms
len=46 ip=64.233.167.99 ttl=64 id=15467 sport=80 flags=SA seq=2 win=65535 rtt=116.7 ms
len=46 ip=64.233.167.99 ttl=64 id=15471 sport=80 flags=SA seq=3 win=65535 rtt=123.1 ms
len=46 ip=64.233.167.99 ttl=64 id=15472 sport=80 flags=SA seq=4 win=65535 rtt=110.2 ms
len=46 ip=64.233.167.99 ttl=64 id=15473 sport=80 flags=SA seq=5 win=65535 rtt=109.3 ms
len=46 ip=64.233.167.99 ttl=64 id=15474 sport=80 flags=SA seq=6 win=65535 rtt=153.0 ms
len=46 ip=64.233.167.99 ttl=64 id=15476 sport=80 flags=SA seq=7 win=65535 rtt=115.8 ms
^C
--- 64.233.167.99 hping statistic ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 109.3/126.3/157.6 ms
```

Рис. 138. Трассировка UDP при пинге в Google

Параметры следующие: -ttl – время жизни, 15 -tr-keep-ttl – фиксация трафика каждые 15 прыжков.

Стресс-тест указанный ниже производится с учетом параметров сети и адресации целевого сервера:

```
#hping3 -c 10000 -d 120 -S -w 64 -p 21 -flood --rand-source целевой IP-адрес.
```

```
nastya@kali:~$ sudo hping3 -c 10000 -d 120 -S -w 64 -p 21 --flood --rand-source 192.168.1.101
HPING 192.168.1.101 (eth1 192.168.1.101): S set, 40 headers + 120 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.1.101 hping statistic ---
622063 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Рис. 139. Реализация DoS-атаки посредством hping3

Синтаксис команды:

-c 1001 – количество пакетов для отправки.

-d 120 – размер каждого пакета, который будет отправлен на целевую машину.

-S – отправляются только пакеты SYN.

-w 64 – размер окна TCP.

-p 21 – порт назначения (используется 21 порт FTP).

-flood – отправка пакетов так быстро, как возможно, не заботясь об отображении входящих пакетов. Режим флуда.

-rand-source – использование рандомных IP-адресов источника. Вы также можете использовать -a или -spoof чтобы спрятать имя хоста.

192.168.1.37 – целевой IP-адрес или IP-адрес целевой машины.

Также вы можете использовать здесь сайт.

Поскольку с помощью hping3 нельзя отследить действительно ли работает данная команда, нужно обратиться к команде top.

```
top - 18:23:04 up 1 min, 1 user, load average: 2,07, 0,75, 0,27
Tasks: 139 total, 1 running, 138 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,0 us, 12,5 sy, 0,0 ni, 87,5 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 1992,1 total, 1169,6 free, 462,8 used, 359,8 buff/cache
MiB Swap: 2046,0 total, 2046,0 free, 0,0 used. 1375,0 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1284 nastya   20   0   9016   3512 3044  R   5,9   0,2    0:00.02 top
     1 root     20   0 166368 10708 8148  S   0,0   0,5    0:03.12 systemd
     2 root     20   0      0      0      0  S   0,0   0,0    0:00.00 kthreadd
     3 root      0  -20      0      0      0  I   0,0   0,0    0:00.00 rcu_gp
     4 root      0  -20      0      0      0  I   0,0   0,0    0:00.00 rcu_par_gp
     5 root     20   0      0      0      0  I   0,0   0,0    0:00.00 kworker/0:0-events
     6 root      0  -20      0      0      0  I   0,0   0,0    0:00.00 kworker/0:0H-kblockd
     7 root     20   0      0      0      0  I   0,0   0,0    0:00.15 kworker/0:1-events
     8 root     20   0      0      0      0  I   0,0   0,0    0:00.00 kworker/u2:0-events_unbound
     9 root      0  -20      0      0      0  I   0,0   0,0    0:00.00 mm_percpu_wq
    10 root     20   0      0      0      0  S   0,0   0,0    0:00.14 ksoftirqd/0
    11 root     20   0      0      0      0  I   0,0   0,0    0:00.26 xen_xcbad
```

Рис. 140. Выполнение команды top в обычном состоянии

Выполнить

Сравните данные, выводимые командой в нормальном состоянии хоста, с данными хоста при реализации атаки.


```

top - 18:40:51 up 1:11, 1 user, load average: 0,12, 0,45, 1,09
Tasks: 142 total, 1 running, 141 sleeping, 0 stopped, 0 zombie
%Cpu(s): 3,4 us, 2,4 sy, 0,0 ni, 94,2 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
MiB Mem : 1992,1 total, 424,5 free, 604,1 used, 963,6 buff/cache
MiB Swap: 2046,0 total, 2044,7 free, 1,3 used. 1188,3 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR  S  %CPU  %MEM    TIME+  COMMAND
 595 root        20   0  408140 127916 55380 S   3,3   6,3   1:01.85 Xorg
1121 nastya     20   0  400004  81244  64060 S   3,0   4,0   0:07.73 qterminal
 477 root        20   0   8096   4728   1664 S   0,3   0,2   0:05.95 haveged
 835 nastya     20   0  163108  2804   2328 S   0,3   0,1   0:20.27 VBoxClient
 882 nastya     20   0  391528  87488  61428 S   0,3   4,3   0:21.39 xfwm4
1096 nastya     20   0  515372  37980  30728 S   0,3   1,9   0:20.36 panel-16-pulsea
3027 root        20   0   9016   3492   3020 R   0,3   0,2   0:00.10 top
  1 root        20   0 100876  10708  8080 S   0,0   0,5   0:05.07 systemd
  2 root        20   0     0     0     0 S   0,0   0,0   0:00.00 kthreadd
  3 root        20   0     0     0     0 I   0,0   0,0   0:00.00 rcu_gp
  4 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 rcu_par_gp
  6 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 kworker/0:0H-kblockd
  9 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 mm_percpu_wq
 10 root        20   0     0     0     0 S   0,0   0,0   1:20.78 ksoftirqd/0
 11 root        20   0     0     0     0 I   0,0   0,0   0:02.68 rcu_sched
 12 root        rt   0     0     0     0 S   0,0   0,0   0:00.05 migration/0
 13 root        20   0     0     0     0 S   0,0   0,0   0:00.00 cpuhp/0
 14 root        20   0     0     0     0 S   0,0   0,0   0:00.00 kdevtmpfs
 15 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 netns
 16 root        20   0     0     0     0 S   0,0   0,0   0:00.00 kauditd
 17 root        20   0     0     0     0 S   0,0   0,0   0:00.00 khungtaskd
 18 root        20   0     0     0     0 S   0,0   0,0   0:00.00 oom_reaper
 19 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 writeback
 20 root        20   0     0     0     0 S   0,0   0,0   0:00.00 kcompactd0
 21 root        25   5     0     0     0 S   0,0   0,0   0:00.00 ksm
 22 root        39  19     0     0     0 S   0,0   0,0   0:00.39 khugepaged
 66 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 kintegrityd
 67 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 kblockd
 68 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 blkcg_punt_bio
 69 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 edac-poller
 70 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 devfreq_wq
 73 root        20   0     0     0     0 S   0,0   0,0   0:00.27 kswapd0
 74 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 kthrotld
 75 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 acpi_thermal_pm
 76 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 ipv6_addrconf
 87 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 kstrp
 91 root        0 -20     0     0     0 I   0,0   0,0   0:00.00 kworker/u3:0

```

Рис. 141. Выполнение команды top во время совершения атаки

Эта атака не оказала влияния на оперативную память, но полностью поглотила ресурсы процессора.

Пояснение к параметрам команды:

а) us – (User CPU time) время, затраченное на работу программ пользователей sy – (System CPU time) время, затраченное на работу процессов ядра;

б) ni – (Nice CPU time) время, затраченное на работу программ с измененным приоритетом;

в) id – простой процессора;

г) wa – (iowait) время, затраченное на завершение ввода-вывода;

д) hi – (Hardware IRQ) время, затраченное на обработку hardware-прерываний;

е) isi – (Software Interrupts) время, затраченное на работу обработки software-прерываний (network);

ё) st – (Steal Time) время, «украденное» гипервизором у этой виртуальной машины для других задач (например, работа другой виртуальной машины).

Выполнить

Произведите настройку брандмауэра с помощью пакета UFW. В этом разделе ознакомимся со стандартной настройкой брандмауэра в рамках локальной сети с помощью пакета Ufw. Сначала необходимо убедиться установлен ли данный пакет на нашу ОС, если нет выполнить полную установку:

1. Введите команду на приведенном ниже рис. 142.

```
nastya@kali:~$ sudo apt install ufw
[sudo] пароль для nastya:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет ufw самой новой версии (0.36-6).
Следующие пакеты устанавливались автоматически и больше не требуются:
 libpython3.7-dev python3.7-dev
Для их удаления используйте «sudo apt autoremove».
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и
 362 пакетов не обновлено.
```

Рис. 142. Установка пакета UFW

При установке программного пакета ufw с использованием диспетчера пакетов он будет включать в себя профили приложений, находящиеся в каталоге /etc/ufw/applications.d, который определяет приложение или службу и соответствующие им настройки безопасности, например, открытые или закрытые порты.

Все профили создаются вручную. Посмотреть созданные профили можно следующим образом:

```
nastya@kali:~$ sudo ufw app list
Available applications:
AIM
Bonjour
CIFS
DNS
Deluge
IMAP
IMAPS
IPP
KTorrent
Kerberos Admin
Kerberos Full
Kerberos KDC
Kerberos Password
LDAP
LDAPS
LPD
MSN
```

Рис. 143. Вывод установленных профилей приложений

Если ваш сервер настроен на IPv6, убедитесь, что ваш брандмауэр UFW настроен с поддержкой IPv6 и IPv4. Чтобы проверить его настройки, откройте файл конфигурации UFW, используя редактор:
#vi /etc/default/ufw

```
# the changes to take affect.
IPV6=yes

# Set the default input policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_INPUT_POLICY="DROP"

# Set the default output policy to ACCEPT, DROP, or REJECT. Please note that if
# you change this you will most likely want to adjust your rules.
DEFAULT_OUTPUT_POLICY="ACCEPT"

# Set the default forward policy to ACCEPT, DROP or REJECT. Please note that
# if you change this you will most likely want to adjust your rules
DEFAULT_FORWARD_POLICY="DROP"

# Set the default application policy to ACCEPT, DROP, REJECT or SKIP. Please
# note that setting this to ACCEPT may be a security risk. See 'man ufw' for
# details
DEFAULT_APPLICATION_POLICY="SKIP"

# By default, ufw only touches its own chains. Set this to 'yes' to have ufw
# manage the built-in chains too. Warning: setting this to 'yes' will break
# non-ufw managed firewall rules
MANAGE_BUILTINS=no

#
# IPT backend
#
# only enable if using iptables backend
IPT_SYSCTL=/etc/ufw/sysctl.conf
```

Рис. 144. Включение UFW под IPv6

Сохраним изменения файла, а затем перезапустим брандмауэр:
ufw disable – выключение брандмауэра;
ufw enable – включение брандмауэра.

Выполнить

Введите команду на приведенном ниже рис. 145.

```
nastya@kali:~$ sudo ufw disable
[sudo] пароль для nastya:
Firewall stopped and disabled on system startup
nastya@kali:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Рис. 145. Перезагрузка брандмауэра

Чтобы к серверу можно было «достучаться» по определенному порту, его нужно сначала открыть. UFW хорош тем, что вам даже не нужно помнить номер порта – нужно знать только название сервиса.

Например, вот как можно разрешить подключение по SSH:

```
nastya@kali:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
nastya@kali:~$ sudo ufw deny ssh/tcp
Rule updated
Rule updated (v6)
nastya@kali:~$ sudo ufw allow 2222/tcp
Rule added
Rule added (v6)
```

Рис. 146. Разрешение SSH-соединения

После разрешения ssh можно включить ufw командой:
#ufw enable

```
nastya@kali:~$ sudo ufw enable
Firewall is active and enabled on system startup
```

Рис. 147. Включение ufw

Так же можно открыть определенный порт в брандмауэре, чтобы разрешить подключение через него к определенному сервису. Например, если вам необходимо настроить веб-сервер, который по умолчанию прослушивает порт **80** (HTTP) и **443** (HTTPS).

Выполнить

1. Введите команду: #ufw allow http

```
nastya@kali:~$ sudo ufw allow http
Rule added
Rule added (v6)
```

Рис. 148. Открытие порта 80 HTTP на UFW

2. То же самое делаем и с https:
Введите команду: #ufw allow https

```
nastya@kali:~$ sudo ufw allow https
Rule added
Rule added (v6)
```

Рис. 149. Открытие порта 443 HTTPS на UFW

Ufw позволяет разрешить определенному IP-адресу доступ ко всем портам сервера: #ufw allow from целевой IP-адрес.

```
nastya@kali:~$ sudo ufw allow from 192.168.1.101
Rule added
nastya@kali:~$
```

Рис. 150. Разрешение доступа определенного IP-адреса

Если нужно разрешить доступ конкретному IP-адресу только к определенному порту, то делается это так:

#ufw allow from целевой IP-адрес to any port 22

```
nastya@kali:~$ sudo ufw allow from 192.168.1.101 to any port 22
Rule added
nastya@kali:~$
```

Рис. 151. Разрешение доступа определенного IP-адреса по нужному порту

Выполнить

1. Разрешим доступ подсети для конкретного порта:

Введите команду: #ufw allow from диапазон IP-адресов to any port 22

```
nastya@kali:~$ sudo ufw allow from 192.168.1.0/24 to any port 22
Rule added
```

Рис. 152. Разрешение доступа подсети для конкретного порта

2. Разрешим доступ через определенный сетевой интерфейс:

Введите команду: #ufw allow in on eth1 to any port 22

```
nastya@kali:~$ sudo ufw allow in on eth1 to any port 22
Rule added
Rule added (v6)
```

Рис. 153. Организация доступа через определенный интерфейс

Проверим запись всех наших ранее прописанных правил.

```
nastya@kali:~$ sudo ufw status
Status: active

To Action From
--
80 ALLOW 127.0.0.1
22/tcp DENY Anywhere
2222/tcp ALLOW Anywhere
443/tcp ALLOW Anywhere
80/tcp ALLOW Anywhere
5000:5003/tcp ALLOW Anywhere
5000:5003/udp ALLOW Anywhere
Anywhere ALLOW 192.168.1.101
22 ALLOW 192.168.1.101
22 ALLOW 192.168.1.0/24
22 on eth1 ALLOW Anywhere
22/tcp (v6) DENY Anywhere (v6)
2222/tcp (v6) ALLOW Anywhere (v6)
443/tcp (v6) ALLOW Anywhere (v6)
80/tcp (v6) ALLOW Anywhere (v6)
5000:5003/tcp (v6) ALLOW Anywhere (v6)
5000:5003/udp (v6) ALLOW Anywhere (v6)
22 (v6) on eth1 ALLOW Anywhere (v6)
```

Рис. 154. Вывод списка правил

Далее с помощью сканера портов Nmap удаленно проверим работу наших правил.

```
C:\Documents and Settings\nastya>nmap 192.168.1.103
Starting Nmap 6.01 ( http://nmap.org ) at 2020-05-15 16:01 Էնэм эфш (чшыр)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is abled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.103
Host is up (0.0039s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
21/tcp    opened  ftp
22/tcp    opened  ssh
80/tcp    opened  http
443/tcp   opened  https
MAC Address: 08:00:27:7C:33:41 (Cadmus Computer Systems)
Nmap done: 1 IP address (1 host up) scanned in 5.57 seconds
```

Рис. 155. Сканирование портов Nmap

Как можно заметить из вывода Nmap с данного удаленного компьютера имеем доступ только к прописанным в правилах портам.

Настройка брандмауэра с помощью пакета Iptables

Теперь ознакомимся со стандартной настройкой брандмауэра в рамках локальной сети с помощью пакета Iptables.

Iptables предустановлен практически во всех дистрибутивах Linux, но если его нет, в системах Ubuntu/Debian воспользуйтесь командами:

```
# apt-get update
# apt-get install iptables
```

```
nastya@kali:~$ sudo apt-get update
Пол:1 http://dl.google.com/linux/chrome/deb stable InRelease [1811 B]
Пол:2 http://dl.google.com/linux/chrome/deb stable/main amd64 Packages [1128 B]
Пол:3 http://mirror-1.truenetwork.ru/kali kali-rolling InRelease [30,5 kB]
Пол:4 http://mirror-1.truenetwork.ru/kali kali-rolling/main amd64 Packages [16,5 MB]
Пол:5 http://mirror-1.truenetwork.ru/kali kali-rolling/non-free amd64 Packages [196 kB]
Пол:6 http://mirror-1.truenetwork.ru/kali kali-rolling/contrib amd64 Packages [98,5 kB]
Получено 16,8 MB за 21с (811 kB/s)
Чтение списков пакетов... Готово
nastya@kali:~$ sudo apt-get install iptables
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Уже установлен пакет iptables самой новой версии (1.8.4-3).
Следующие пакеты устанавливались автоматически и больше не требуются:
 libpython3.7-dev python3.7-dev
Для их удаления используйте «sudo apt autoremove».
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 683 пакетов не обновлено.
```

Рис. 156. Установка/обновление Iptables

Перед настройкой следует посмотреть какие правила уже есть в Iptables: #iptables-save

```
nastya@kali:~$ sudo iptables-save
# Generated by iptables-save v1.8.4 on Wed Apr 29 22:31:31 2020
*filter
:INPUT ACCEPT [3016:136910]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1016:41804]
:ufw-before-logging-input - [0:0]
:ufw-before-logging-output - [0:0]
:ufw-before-logging-forward - [0:0]
:ufw-before-input - [0:0]
:ufw-before-output - [0:0]
:ufw-before-forward - [0:0]
:ufw-after-input - [0:0]
:ufw-after-output - [0:0]
:ufw-after-forward - [0:0]
:ufw-after-logging-input - [0:0]
:ufw-after-logging-output - [0:0]
:ufw-after-logging-forward - [0:0]
:ufw-reject-input - [0:0]
:ufw-reject-output - [0:0]
:ufw-reject-forward - [0:0]
:ufw-track-input - [0:0]
:ufw-track-output - [0:0]
:ufw-track-forward - [0:0]
COMMIT
# Completed on Wed Apr 29 22:31:31 2020
```

Рис. 157. Вывод текущих правил Iptables

Создадим пару правил, которые желательно иметь в цепочке INPUT практически для любой серверной конфигурации.

Правило разрешающее любой входящий трафик на служебный внутренний loopback device. Это может потребоваться для корректной работы разного рода служб и приложений:

```
#iptables -I INPUT 1 -i -I 10 -j ACCEPT
```

Здесь `-I` означает, что вставляем правило в определенную часть списка:

`-i` – сетевой адаптер, через который приходят пакеты;

`-j` – действия, которые будут выполняться над пакетом.

Создадим ряд однотипных правил для запущенных на нашем сервере служб. Например, правила разрешающие входящие TCP-пакеты для подключения к службам SSH-сервера OpenSSH, веб-сервера Apache, прокси-сервера Squid, сервера времени NTPD на сетевом интерфейсе, направленном в локальную сеть (eth1):

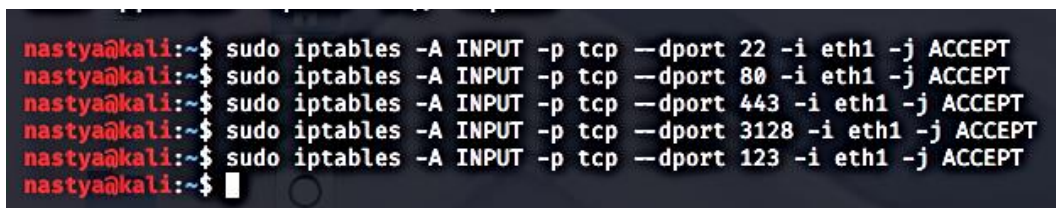
а) #iptables -A INPUT -p tcp --dport 22 -i eth1 -j ACCEPT

б) #iptables -A INPUT -p tcp --dport 80 -i eth1 -j ACCEPT

в) #iptables -A INPUT -p tcp --dport 443 -i eth1 -j ACCEPT

г) #iptables -A INPUT -p tcp --dport 3128 -i eth1 -j ACCEPT

д) #iptables -A INPUT -p tcp --dport 123 -i eth1 -j ACCEPT



```
nastya@kali:~$ sudo iptables -A INPUT -p tcp --dport 22 -i eth1 -j ACCEPT
nastya@kali:~$ sudo iptables -A INPUT -p tcp --dport 80 -i eth1 -j ACCEPT
nastya@kali:~$ sudo iptables -A INPUT -p tcp --dport 443 -i eth1 -j ACCEPT
nastya@kali:~$ sudo iptables -A INPUT -p tcp --dport 3128 -i eth1 -j ACCEPT
nastya@kali:~$ sudo iptables -A INPUT -p tcp --dport 123 -i eth1 -j ACCEPT
nastya@kali:~$ █
```

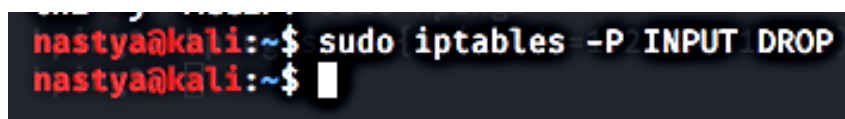
Рис. 158. Настройка входящего трафика с портов

Теперь все TCP-подключения с этими номерами портов будут разрешены. Дополнительно создадим правило разрешающее ответы на echo-запросы по протоколу ICMP поступающие на интерфейс внутренней локальной сети нашего Linux-сервера:

```
#iptables -A INPUT -p icmp --icmp-type echo-request -i eth1 -j ACCEPT
```

Меняем политику для всего входящего трафика не попавшего ни под одно правило в цепочке INPUT на запрещающую:

```
#iptables -P INPUT DROP
```



```
nastya@kali:~$ sudo iptables -P INPUT DROP
nastya@kali:~$ █
```

Рис. 159. Ограничение входящего трафика

Проверим что получилось в результате для протокола IPv4:
#iptables -S

```
nastya@kali:~$ sudo iptables -S
-P INPUT DROP
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -i eth1 -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -i eth1 -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -i eth1 -p tcp -m tcp --dport 443 -j ACCEPT
```

Рис. 160. Вывод настроенных правил

Настроенные нами правила вступают в силу сразу после их создания. Значит мы, как и в предыдущем варианте, можем проверить доступность портов с удаленного компьютера локальной сети.

Для этого воспользуемся сканированием открытых портов Nmap.

```
C:\Documents and Settings\nastya>nmap 192.168.1.103
Starting Nmap 6.01 ( http://nmap.org ) at 2020-05-15 16:23 4323 эш (чшьд)
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is abled.
Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.1.103
Host is up (0.0045s latency).
Not shown: 996 filtered ports
PORT      STATE SERVICE
22/tcp    opened  ssh
80/tcp    opened  http
443/tcp   opened  https
3128/tcp  opened  squid-http
MAC Address: 08:00:27:7C:33:41 (Cadmus Computer Systems)

Nmap done: 1 IP address (1 host up) scanned in 5.54 seconds
```

Рис. 161. Вывод сканирования Nmap

Исходя из полученных результатов, можно отметить, что настройка параметров фаервола была правильной и весь трафик, неподходящий под эти правила, отбрасывается.

Многие брандмауэры включают правило для отбрасывания пакетов TCP, для которых не установлена опция TCP Timestamp, что является обычным явлением в популярных сканерах портов. Просто добавьте опцию *tcp-timestamp*, чтобы добавить информацию о метках времени: #hping3 -S целевой IP-адрес -p 80 -tcp-timestamp.


```
nastya@kali:~$ sudo hping3 -S 127.0.0.1 -p 80 --tcp-timestamp
HPING 127.0.0.1 (lo 127.0.0.1): S set, 40 headers + 0 data bytes
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=RA seq=0 win=0 rtt=3.9 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=RA seq=1 win=0 rtt=8.6 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=RA seq=2 win=0 rtt=3.1 ms
len=40 ip=127.0.0.1 ttl=64 DF id=0 sport=80 flags=RA seq=3 win=0 rtt=7.7 ms
^C
— 127.0.0.1 hping statistic —
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 3.1/5.8/8.6 ms
nastya@kali:~$
```

Рис. 162. Установка опции TCP Timestamp

Исходя из проведенных испытаний можно сказать, что hping3 вполне может входить в арсенал системных администраторов и специалистов, занимающихся безопасностью сетей, так как с помощью этой утилиты можно анализировать сетевой трафик, собирать необходимые пакеты, а также выполнять тестирование локальных сетей.

Также, ознакомившись с работой пакетов Iptables и ufw, можно сделать вывод, что оба этих инструмента имеют право являться основными в обеспечении безопасности сети и отдельного компьютера, а именно использоваться в настройке брандмауэра. Стоит обратить внимание, что ufw не предназначен для обеспечения полной функциональности брандмауэра через свой командный интерфейс, но он предоставляет легкий способ добавления или удаления простых правил. Сейчас в большинстве случаев он используется для централизованных брандмауэров. В то время как настройка брандмауэра с помощью Iptables довольно трудная задача для любого специалиста, но именно с помощью этого инструмента можно максимально защитить любую сеть.

2.6. Особенности аудита ОС Linux

Система аудита Linux позволяет отслеживать информацию относящуюся к безопасности в системе. Основываясь на предварительно настроенных правилах, аудит создает записи журнала, чтобы записывать как можно больше информации о событиях, происходящих в вашей системе. Эта информация имеет решающее значение для критически важных сред для определения нарушителя политики безопасности и действий, которые он выполнил. Аудит не обеспечивает дополнительной безопасности вашей системы; скорее, его можно использовать для обнаружения нарушений политик безопасности, используемых в вашей системе. Эти нарушения можно предотвратить с помощью дополнительных мер безопасности, таких как SELinux.

В следующем списке представлена некоторая информация, которую аудит может записывать в свои файлы журналов.

1. Дата и время, тип и результат события.
2. Метки чувствительности субъектов и объектов.
3. Связь события с личностью пользователя, инициировавшего событие.
4. Все изменения конфигурации аудита и попытки доступа к файлам журнала аудита.
5. Все виды использования механизмов аутентификации, таких как SSH, Kerberos и другие.
6. Изменения в любой доверенной базе данных, такой как /etc/passwd.
7. Попытки импортировать или экспортировать информацию в систему или из нее.
8. Включение или исключение события на основе идентификатора пользователя, меток темы, объекта и других атрибутов.

Использование системы аудита также является требованием для ряда сертификатов, связанных с безопасностью. Аудит предназначен для удовлетворения или превышения требований следующих сертификатов или руководств по соответствию:

1. Профиль защиты контролируемого доступа (CAPP).
2. Маркированный профиль защиты безопасности (LSPP).
3. Базовый контроль доступа к набору правил (RSBAC).
4. Руководство по эксплуатации Национальной программы промышленной безопасности (NISPOM).
5. Федеральный закон об управлении информационной безопасностью (FISMA).
6. Индустрия платежных карт – стандарт безопасности данных (PCI-DSS).
7. Руководство по технической реализации безопасности (STIG).

1. Просмотр доступа к файлам. Аудит может отслеживать доступ к файлу или каталогу, их изменение, выполнение или изменение атрибутов файла. Это полезно, например, для обнаружения доступа к важным файлам и наличия журнала аудита в случае повреждения одного из этих файлов.

2. Мониторинг системных вызовов. Аудит можно настроить для создания записи в журнале каждый раз, когда используется определенный системный вызов. Это можно использовать, например, для отслеживания изменений системного времени путем мониторинга `settimeofday`, `clock_adjtime` и других системных вызовов, связанных со временем.

3. Запись команд, выполняемых пользователем. Аудит может отслеживать, выполнялся ли файл, поэтому можно определить правила для регистрации каждого выполнения конкретной команды. Например, правило может быть определено для каждого исполняемого файла в /bin каталоге. Полученные записи журнала затем можно искать по идентификатору пользователя, чтобы создать контрольный журнал выполненных команд для каждого пользователя.

4. Запись выполнения системных путей. Помимо наблюдения за доступом к файлу, который преобразует путь в индексный дескриптор при вызове правила, аудит теперь может отслеживать выполнение пути, даже если он не существует при вызове правила, или если файл заменяется после вызова правила. Это позволяет правилам продолжать работать после обновления исполняемого файла программы или еще до его установки.

5. Запись событий безопасности. Модуль *pat_faillock* аутентификации способен записывать неудачные попытки входа в систему. Аудит также может быть настроен для записи неудачных попыток входа в систему и предоставления дополнительной информации о пользователе, который пытался войти в систему.

6. Поиск событий. Audit предоставляет утилиту *ausearch*, которую можно использовать для фильтрации записей журнала и предоставления полного контрольного журнала на основе ряда условий.

7. Запуск сводных отчетов. Утилита *aureport* может использоваться, помимо прочего, для создания ежедневных отчетов о зарегистрированных событиях. Затем системный администратор может проанализировать эти отчеты и продолжить расследование подозрительной активности.

8. Мониторинг доступа к сети. Утилиты *iptables* и *ebtables* можно настроить для запуска событий аудита, что позволяет системным администраторам отслеживать доступ к сети.

Архитектура системы аудита специфичная. Система аудита состоит из двух основных частей: приложений и утилит пользовательского пространства и обработки системных вызовов на стороне ядра. Компонент ядра получает системные вызовы от приложений пользовательского пространства и фильтрует их с помощью одного из следующих фильтров: *user*, *task*, *fstype* или *exit*.

Как только системный вызов проходит фильтр *исключения*, он отправляется через один из вышеупомянутых фильтров, который на основе конфигурации правил аудита отправляет его демону аудита для дальнейшей обработки.

Демон аудита пользовательского пространства собирает информацию из ядра и создает записи в файле журнала. Другие утилиты пользовательского пространства аудита взаимодействуют с демоном аудита, компонентом аудита ядра или файлами журнала аудита:

– **audisp** – демон диспетчера аудита взаимодействует с демоном аудита и отправляет события другим приложениям для дальнейшей обработки. Цель этого демона – предоставить подключаемый механизм, позволяющий аналитическим программам в реальном времени взаимодействовать с событиями аудита;

– **auditctl** – утилита управления аудитом взаимодействует с компонентом аудита ядра для управления правилами и управления рядом настроек и параметров процесса генерации событий.

Остальные утилиты аудита принимают содержимое файлов журнала аудита в качестве входных данных и генерируют выходные данные в соответствии с требованиями пользователя. Например, утилита **aureport** формирует отчет обо всех записанных событиях.

Установка пакетов аудита. Чтобы использовать систему аудита, в вашей системе должны быть установлены пакеты аудита. Пакеты аудита (аудит и аудит-библиотеки) устанавливаются по умолчанию в Red Hat Enterprise Linux 7. Если у вас не установлены эти пакеты, выполните следующую команду от имени пользователя root, чтобы установить аудит и зависимости.

Выполнить

Введите команду на приведенном ниже рис. 163.

```
[root@localhost grochowskia]# yum install audit
```

Рис. 163. Установка пакетов аудита

На рис. 163 видим команду, с помощью которой скачиваем пакеты аудита в нашу операционную систему. После установки пакетов аудита перейдем к настройке служб.

Настройка службы аудита. Демон аудита можно настроить в `/etc/audit/auditd.conf` файле. Этот файл состоит из параметров конфигурации, которые изменяют поведение демона аудита. Пустые строки и текст после знака решетки (`#`) игнорируются.

Настройка аудита для безопасной среды. Конфигурация по умолчанию `auditd` должна подходить для большинства сред. Однако, если ваша среда должна соответствовать строгим политикам безопасно-

сти, для конфигурации демона аудита в `/etc/audit/auditd.conf` файле предлагаются следующие параметры:

а) ***log_file***. Каталог, содержащий файлы журнала аудита (обычно `/var/log/audit/`), должен находиться в отдельной точке монтирования. Это предотвращает использование пространства в этом каталоге другими процессами и обеспечивает точное определение оставшегося места для демона аудита.

б) ***max_log_file***. Указывает максимальный размер одного файла журнала аудита, который необходимо установить для полного использования доступного пространства в разделе, содержащем файлы журнала аудита.

в) ***max_log_file_action***. Определяет, какое действие следует предпринять после достижения установленного предела *max_log_file*, *keep_logs* чтобы предотвратить перезапись файлов журнала аудита.

г) ***space_left***. Указывает объем свободного места, оставшегося на диске, для которого *space_left_action* запускается действие, заданное в параметре. Должно быть установлено число, которое дает администратору достаточно времени, чтобы ответить и освободить место на диске. Значение *space_left* зависит от скорости создания файлов журнала аудита.

д) ***space_left_action***. Рекомендуется установить для *space_left_action* параметра значение `email` или `exec` соответствующим методом оповещения.

е) ***admin_space_left***. Указывает абсолютный минимум свободного места, для которого *admin_space_left_action* запускается действие, заданное в параметре, должно быть установлено значение, оставляющее достаточно места для регистрации действий, выполняемых администратором.

ж) ***admin_space_left_action***. Следует установить, *single* чтобы перевести систему в однопользовательский режим и позволить администратору освободить место на диске.

и) ***disk_full_action***. Указывает действие, которое запускается, когда в разделе, содержащем файлы журнала аудита, нет свободного места, должно быть задано значение *halt* или *single*. Это гарантирует, что система либо остановится, либо будет работать в однопользовательском режиме, когда аудит больше не сможет регистрировать события.

к) ***disk_error_action***. Указывает действие, которое запускается в случае обнаружения ошибки в разделе, содержащем файлы журнала аудита, должно быть установлено на *syslog*, *single* или *halt*, в зависимости от локальной политики безопасности, касающейся обработки сбоев оборудования.

л) *Flush*. Должно быть установлено значение *incremental_async*. Работает в сочетании с *freq*-параметром, определяющим, сколько записей можно отправить на диск, прежде чем принудительно выполнить жесткую синхронизацию с винчестером. Параметр *freq* должен быть установлен в 100. Эти параметры обеспечивают синхронизацию данных о событиях аудита с файлами журналов на диске, сохраняя при этом хорошую производительность при всплесках активности.

Остальные параметры конфигурации должны быть установлены в соответствии с локальной политикой безопасности.

После *auditd* настройки запустите службу для сбора информации об аудите и сохранения ее в файлах журнала. Используйте следующую команду в качестве пользователя *root* для запуска *auditd* (рис. 164):

Выполнить

1. Введите команду на приведенном ниже рис. 164.

```
[root@localhost grochowskia]# service auditd start
Redirecting to /bin/systemctl start auditd.service
```

Рис. 164. Запуск службы аудита

После того, как выполнили команду, система отправляет нам ответное сообщение о том, что аудит был запущен.

2. Чтобы настроить *auditd* запуск во время загрузки, выполним:

```
[root@localhost grochowskia]# systemctl enable auditd
```

Рис. 165. Запуск демона *auditd* во время автозагрузки

С помощью данной команды ставим аудит в автозапуск.

С помощью команды *service auditd <действие>* можно выполнить ряд других действий, где действие может быть одним из следующих:

- а) *service auditd stop* – остановка *auditd*.
- б) *service auditd restart* – перезагрузка *auditd*.
- в) *service auditd Reload* или *service auditd force-reload* – перезагружает конфигурацию *auditd* из */etc/audit/auditd.conf* файла.
- г) *service auditd rotate* – ротация файлов журнала в */var/log/audit/* каталоге.
- д) *service auditd resume* – возобновляет регистрацию событий аудита после того, как она была ранее приостановлена, например, когда недостаточно свободного места на разделе диска, содержащем файлы журнала аудита.

е) `service auditd Condrestart` или `service auditd try-restart` – перезапускает **auditd** только в том случае, если он уже запущен.

ж) `service auditd status` – отображает текущее состояние аудита.

Проверим статус нашего аудита.

```
[root@localhost grochowski]# service auditd status
Redirecting to /bin/systemctl status auditd.service
auditd.service - Security Auditing Service
   Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled)
   Active: active (running) since Вт 2022-05-31 08:28:31 MSK; 2h 39min ago
     Docs: man:auditd(8)
           https://github.com/linux-audit/audit-documentation
   Main PID: 2459 (auditd)
   CGroup: /system.slice/auditd.service
           └─2459 /sbin/auditd -n

май 31 08:28:31 localhost.localdomain augenrules[2460]: enabled 1
май 31 08:28:31 localhost.localdomain augenrules[2460]: failure 1
май 31 08:28:31 localhost.localdomain augenrules[2460]: pid 2459
```

Рис. 166. Проверка статуса аудита

Перейдем к определению правил аудита. Система аудита работает на основе набора правил, определяющих, что должно фиксироваться в файлах журналов. Можно указать следующие типы правил аудита:

1. Правила контроля

Разрешить изменение поведения системы аудита и некоторых ее настроек.

2. Правила файловой системы

Также известные как файловые часы, позволяют контролировать доступ к определенному файлу или каталогу.

3. Правила системного вызова

Разрешить регистрацию системных вызовов, которые делает любая указанная программа.

Правила аудита могут быть установлены:

1. В командной строке с помощью утилиты **auditctl**. Важно обратить внимание на то, что эти правила не сохраняются после перезагрузки.

2. В `/etc/audit/audit.rules` файле.

Определение правил аудита с помощью auditctl. Здесь **auditctl** команда позволяет управлять основными функциями системы аудита и определять правила, определяющие, какие события аудита регистрируются.

Все команды, взаимодействующие со службой аудита и файлами журнала аудита, требуют привилегий root. Убедитесь, что вы выполняете эти команды как пользователь root. Кроме того, `CAP_AUDIT_CONTROL`

(позволяет включать или выключать аудит ядра; изменять фильтрующие правила аудита; получать состояние аудита и фильтрующие правила) требуется для настройки служб аудита, а настройка `CAP_AUDIT_WRITE` (позволяет записывать данные в журнал аудита ядра) требуется для регистрации сообщений пользователей.

Ниже приведены некоторые из правил управления, которые позволяют изменять поведение системы аудита.

Ключ `-b` устанавливает максимальное количество существующих буферов аудита в ядре.

```
[root@localhost grochowskia]# auditctl -b 2048
enabled 1
failure 1
pid 2459
rate_limit 0
backlog_limit 2048
lost 0
backlog 1
```

Рис. 167. Установка количества буферов в ядре

Ключ `-f` задает действие, которое выполняется при обнаружении критической ошибки (рис. 168).

```
[root@localhost grochowskia]# auditctl -f 2
enabled 1
failure 2
pid 2459
rate_limit 0
backlog_limit 2048
lost 0
backlog 1
```

Рис. 168. Действие, которое выполняется при обнаружении критической ошибки

Приведенная выше конфигурация вызывает панику ядра в случае критической ошибки.

Ключ `-r` устанавливает скорость генерируемых сообщений в секунду.


```
[root@localhost grochowskia]# auditctl -r 0
enabled 1
failure 1
pid 2459
rate_limit 0
backlog_limit 2048
lost 0
backlog 1
```

Рис. 169. Установка скорости генерируемых сообщений в секунду

Приведенная выше конфигурация не устанавливает ограничения скорости для генерируемых сообщений.

Ключ `-s` сообщает о состоянии системы аудита (рис. 170).

```
[root@localhost grochowskia]# auditctl -s
enabled 1
failure 1
pid 2459
rate_limit 0
backlog_limit 2048
lost 0
backlog 0
loginuid_immutable 0 unlocked
```

Рис. 170. Состояние аудита

Ключ `-l` перечисляет все загруженные в данный момент правила аудита (рис. 171).

```
[root@localhost grochowskia]# auditctl -l
No rules
```

Рис. 171. Просмотр действующих правил аудита

Ключ `-D` удаляет все загруженные в данный момент правила аудита.

```
[grochowskia@localhost ~]$ sudo auditctl -D
No rules
```

Рис. 172. Удаление действующих правил аудита

Рассмотрим определение правил файловой системы. Чтобы определить правило файловой системы, следует использовать следующий синтаксис: `auditctl -w путь_к_файлу -p разрешения -k имя_ключа`.

Ключ `-w` указывает на то, что это правило файловой системы. Далее следует путь к файлу или каталогу.

Ключ `-p` может содержать любые комбинации прав доступа `r` (чтение), `w` (запись), `x` (выполнение) и `a` (изменение атрибута).

Ключ `-k` задает имя правила, по которому впоследствии можно фильтровать логи.

Выполнить

Создадим несколько правил:

1. Создадим правило аудита для регистрации изменения файла `/etc/passwd`.

Введите команду на приведенном ниже рис. 173.

```
[root@localhost grochowskia]# auditctl -w /etc/passwd -p wa -k passwd
```

Рис. 173. Создание правила аудита для регистрации изменений в файле `/etc/passwd`

2. Проверим, записалось ли наше правило на рис. 174

```
[root@localhost grochowskia]# auditctl -l  
-w /etc/passwd -p wa -k passwd
```

Рис. 174. Просмотр действующих правил аудита

3. Теперь посмотрим, как наше правило работает. Для этого создадим и сразу же удалим пользователя. Введите команду на приведенном ниже рис. 175.

```
[root@localhost grochowskia]# useradd user  
[root@localhost grochowskia]# userdel user
```

Рис. 175. Создание и удаление пользователя

4. Убедимся, что по нашему правилу сгенерировались события. Для этого построим отчет. Введите команду на приведенном ниже рис. 176.

```
[root@localhost grochowskia]# sudo aureport --summary -k
Key Summary Report
=====
total key
=====
47 passwd
[root@localhost grochowskia]# _
```

Рис. 176. Построение отчета по правилу

5. Теперь посмотрим эти события.

```
-----
type=PATH msg=audit(31.05.2022 11:47:45.662:585) : item=1 name=/etc/passwd+ inode=9052076 dev=fd:00
mode=file,000 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:passwd_file_t:s0 objtype=CREATE
type=PATH msg=audit(31.05.2022 11:47:45.662:585) : item=0 name=/etc/ inode=8388737 dev=fd:00 mode=di
r,755 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:etc_t:s0 objtype=PARENT
type=CMD msg=audit(31.05.2022 11:47:45.662:585) : cwd=/home/grochowskia
type=SYSCALL msg=audit(31.05.2022 11:47:45.662:585) : arch=x86_64 syscall=open success=yes exit=5 a0
=0x7ffff4017e80 a1=0_WRONLY+O_CREAT+O_TRUNC a2=0666 a3=0x0 items=2 ppid=2363 pid=9331 auid=grochowsk
ia uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=tty1 ses=1 co
mm=userdel exe=/usr/sbin/userdel subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key=pass
wd
-----
type=CONFIG_CHANGE msg=audit(31.05.2022 11:47:45.665:586) : auid=grochowskia ses=1 op="updated rules
" path=/etc/passwd key=passwd list=exit res=yes
```

Рис. 177. Просмотр событий аудита

На приведенных выше скриншотах можем увидеть, что наша система аудита записала процесс создания и удаления нашего пользователя user.

Рассмотрим определение правил системных вызовов. Для определения правил системного вызова будем использовать следующий синтаксис:

sudo auditctl -a список,действие -S имя_системного_вызова -F фильтр

Список – это список событий, в который нужно добавить правило. Для упрощения можно воспринимать список как фильтр, позволяющий сделать правило точнее.

Существует пять списков:

- 1) *task* – события, связанные с созданием процессов;
- 2) *entry* – события, происходящие при входе в системный вызов;
- 3) *exit* – события, происходящие во время выхода из системного вызова;
- 4) *user* – события, использующие параметры пользовательского пространства (uid, pid и gid);

5) *exclude* – используется для исключения событий.

На практике в основном используются *entry* и *exit*.

Действие – определяет, что нужно выполнить после события: записать его в журнал (*always*) или не записывать (*never*).

Имя системного вызова – при обращении к какому вызову должен срабатывать триггер и перехватываться событие (например, *open*, *close*, *exit*, и т. д.)

Фильтр – необязательная опция, которая используется для указания дополнительных параметров.

Рассмотрим определение правил для исполняемого файла. Чтобы определить правило исполняемого файла, используйте следующий синтаксис:

```
auditctl -a список,действие [-F arch=cpi -S имя_системного_вызова] -F exe=путь_к_исполняемому_файлу -k имя_ключа
```

Список – это список событий, в который нужно добавить правило. Для упрощения можно воспринимать список как фильтр, позволяющий сделать правило точнее.

Действие – определяет, что нужно выполнить после события: записать его в журнал (*always*) или не записывать (*never*).

Имя системного вызова – при обращении к какому вызову должен срабатывать триггер и перехватываться событие (например, *open*, *close*, *exit*, и т. д.)

Ключ *-k* задает имя правила, по которому впоследствии можно фильтровать логи.

Поиск файлов журнала аудита. Утилита *ausearch* позволяет искать определенные события в файлах журнала аудита. По умолчанию *ausearch* ищет */var/log/audit/audit.log* файл. Использование нескольких параметров в одной команде эквивалентно использованию оператора И между типами полей и оператора ИЛИ между несколькими элементами одного и того же типа поля (*ausearch options –if file_name ausearch*).

Создание аудиторских отчетов. Утилита *aureport* позволяет создавать сводные и столбчатые отчеты о событиях, записанных в файлах журнала аудита. По умолчанию для создания отчета запрашиваются все *audit.log* файлы в каталоге */var/log/audit/*. Вы можете указать другой файл для запуска отчета с помощью команды: *aureport options –if file_name*.

Выполнить

Создадим два пользователя: *user1* и *user2*. Каждому пользователю создадим личный каталог и создадим в этом каталоге несколько файлов.

Далее создадим «секретный каталог», в котором будут храниться некие файлы.

Система аудита должна записать создание пользователей, их вход в систему, создание каталогов и файлов, работу в общей директории.

1. После запуска системы и входа в терминал проверим статус аудита. Введите команду на приведенном ниже рис. 178.

```
[root@localhost grochowskia]# service auditd status
Redirecting to /bin/systemctl status auditd.service
auditd.service - Security Auditing Service
  Loaded: loaded (/usr/lib/systemd/system/auditd.service; enabled)
  Active: active (running) since Вт 2022-05-31 12:41:36 MSK; 4min 57s ago
  Docs: man:auditd(8)
        https://github.com/linux-audit/audit-documentation
  Process: 563 ExecStartPost=/sbin/augenrules --load (code=exited, status=0/SUCCESS)
  Process: 558 ExecStart=/sbin/auditd (code=exited, status=0/SUCCESS)
  Main PID: 559 (auditd)
  CGroup: /system.slice/auditd.service
          └─559 /sbin/auditd

май 31 12:41:36 localhost.localdomain augenrules[563]: /sbin/augenrules: No change
май 31 12:41:36 localhost.localdomain augenrules[563]: No rules
```

Рис. 178. Проверка статуса аудита

На рис. 178 видим, что статус аудита активен. Об этом нам говорит строка “Active: active (running)”.

2. Далее, проверим, установлены ли у нас какие-либо правила, и при наличии удалим их. Введите команду на приведенном ниже рис. 179.

```
[root@localhost grochowskia]# auditctl -l
No rules
```

Рис. 179. Просмотр действующих правил системы аудита

3. Затем установим набор правил, определяющих базовые настройки. Введите команду на приведенном ниже рис. 180.

```
[root@localhost grochowskia]# auditctl -D
No rules
[root@localhost grochowskia]# auditctl -b 320
enabled 1
failure 1
pid 559
rate_limit 0
backlog_limit 320
```

Рис. 180. Правила, определяющие базовые настройки

- а) `auditctl -D` – удаление действующих правил;
- б) `auditctl -b 320` – задали количество буферов, в которых будут храниться сообщения;
- в) `auditctl -f 1` – действие при переполнении буферов: 0 – ничего не делать; 1 – отправить сообщение в `dmesg`; 2 – отправить ядро в режим условной работы.

Далее укажем пользовательские правила наблюдения за конфигурационными файлами системы аудита. Введите команду на приведенном ниже рис. 181.

```
[root@localhost grochowskia]# auditctl -w /etc/audit/auditd.conf -p wa
[root@localhost grochowskia]# auditctl -w /etc/audit/audit.rules -p wa
```

Рис. 181. Создание правил наблюдения за конфигурационными файлами системы аудита

Данное правило позволяет наблюдать за любым воздействием с конфигурационными файлами системы аудита.

Установим наблюдение за журнальными файлами. Введите команду на приведенном ниже рис. 182.

```
[root@localhost grochowskia]# auditctl -w /var/log/audit/
[root@localhost grochowskia]# auditctl -w /var/log/audit/audit.log
```

Рис. 182. Создание правил наблюдения за журнальными файлами

Данное правило наблюдает за любым воздействием с журнальными файлами.

Реализуем создание правил отслеживания файлов паролей и групп. Введите команды на приведенном ниже рис. 183.

```
[root@localhost grochowskia]# auditctl -w /etc/group -p wa -k group
[root@localhost grochowskia]# auditctl -w /etc/passwd -p wa -k passwd
[root@localhost grochowskia]# auditctl -w /etc/shadow -k shadow
```

Рис. 183. Создание правил отслеживания файлов паролей и групп

Данное правило наблюдает за любым воздействием с файлами паролей и групп.

Реализуем создание правил отслеживания конфигурационных и журнальных файлов входа в систему. Введите команды на приведенном ниже рис. 184.


```
[root@localhost grochowski]# auditctl -w /etc/login.defs -p wa -k login
[root@localhost grochowski]# auditctl -w /etc/securetty -k securetty
[root@localhost grochowski]# auditctl -w /var/log/faillog -k faillog
[root@localhost grochowski]# auditctl -w /var/log/lastlog -k lastlog
```

Рис. 184. Создание правил отслеживания конфигурационных и журнальных файлов входа в систему

Данное правило наблюдает за любым воздействием с конфигурационными и журнальными файлами входа в систему.

Реализуем создание правил отслеживания создания и удаления каталогов. Введите команды на приведенном ниже рис. 185.

```
[root@localhost grochowski]# auditctl -a entry,always -S mkdir -S rmdir -k mkdir_rmdir
```

Рис. 185. Создание правил отслеживания создания и удаления каталогов

Данное правило наблюдает за любым воздействием с созданием и удалением каталогов.

Реализуем создание правил отслеживания удаления или создания ссылок. Введите команды на приведенном ниже рис. 186.

```
[root@localhost grochowski]# auditctl -a entry,always -S unlink -S rename -S link -S symlink -k links
```

Рис. 186. Создание правила отслеживания удаления или создания ссылок

Данное правило наблюдает за любым воздействием с созданием и удалением ссылок.

Реализуем создание правил отслеживания изменения прав доступа к файлам и каталогам. Введите команды на приведенном ниже рис. 187.

```
[root@localhost grochowski]# auditctl -a entry,always -S chmod -S chown -k own
```

Рис. 187. Создание правила отслеживания изменения прав доступа к файлам и каталогам

Данное правило наблюдает за любым воздействием с изменением прав доступа к файлам и каталогам.

Реализуем создание правил отслеживания создания, открытия или изменения размеров файлов. Введите команды на приведенном ниже рис. 188.

```
[root@localhost grochowska]# auditctl -a entry,always -S creat -S open -S truncate -k open
```

Рис. 188. Создание правил отслеживания создания, открытия или изменения размеров файлов

Данное правило наблюдает за любым воздействием с созданием, открытием и изменением размеров файлов.

4. Создадим пользователей user1 и user2. Введите команду на приведенном ниже рис. 189.

```
[root@localhost grochowska]# useradd user2
```

Рис. 189. Создание пользователей user1 и user2

Посмотрим записи в журнале аудита по этим событиям.

```
type=PATH msg=audit(01.06.2022 20:48:23.138:2158) : item=4 name=/etc/passwd inode=9052057 dev=fd:00 mode=file,644 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:passwd_file_t:s0 objtype=CREATE
type=PATH msg=audit(01.06.2022 20:48:23.138:2158) : item=3 name=/etc/passwd inode=8852349 dev=fd:00 mode=file,644 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:passwd_file_t:s0 objtype=DELETE
type=PATH msg=audit(01.06.2022 20:48:23.138:2158) : item=2 name=/etc/passwd+ inode=9052057 dev=fd:00 mode=file,644 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:passwd_file_t:s0 objtype=DELETE
type=PATH msg=audit(01.06.2022 20:48:23.138:2158) : item=1 name=/etc/ inode=8388737 dev=fd:00 mode=dir,755 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:etc_t:s0 objtype=PARENT
type=PATH msg=audit(01.06.2022 20:48:23.138:2158) : item=0 name=/etc/ inode=8388737 dev=fd:00 mode=dir,755 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:etc_t:s0 objtype=PARENT
type=CWD msg=audit(01.06.2022 20:48:23.138:2158) : cwd=/home/grochowska
type=SYSCALL msg=audit(01.06.2022 20:48:23.138:2158) : arch=x86_64 syscall=rename success=yes exit=0 a0=0x7fff846eb030 a1=0x7ff216021ce0 a2=0x7fff846eafa0 a3=0x7ff2159327b8 items=5 ppid=2344 pid=2661 auid=grochowska uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=tty1 ses=1 comm=useradd exe=/usr/sbin/useradd subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key=passwd
```

Рис. 190. Записи из журнала аудита о создании пользователя user1

На рис. 190 можем видеть, что в журнал аудита были добавлены записи о создании пользователя user1.

```
type=PATH msg=audit(01.06.2022 20:48:24.789:2297) : item=1 name=/etc/shadow+ inode=9052056 dev=fd:00 mode=file,000 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:shadow_t:s0 objtype=CREATE
type=PATH msg=audit(01.06.2022 20:48:24.789:2297) : item=0 name=/etc/ inode=8388737 dev=fd:00 mode=dir,755 ouid=root ogid=root rdev=00:00 obj=system_u:object_r:etc_t:s0 objtype=PARENT
type=CWD msg=audit(01.06.2022 20:48:24.789:2297) : cwd=/home/grochowska
type=SYSCALL msg=audit(01.06.2022 20:48:24.789:2297) : arch=x86_64 syscall=open success=yes exit=5 a0=0x7fff2c63fac0 a1=0_WRONLY:O_CREAT:O_TRUNC a2=0666 a3=0x6165726373662f72 items=2 ppid=2344 pid=2665 auid=grochowska uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=tty1 ses=1 comm=useradd exe=/usr/sbin/useradd subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key=passwd
```

Рис. 191. Записи из журнала аудита о создании пользователя user2

На рис. 191 можем видеть, что в журнал аудита были добавлены записи о создании пользователя user2.

При создании пользователя автоматически создается каталог этого пользователя в папке /home. Проверим, записал ли аудит эти события (рис. 192).

```
time->Wed Jun 1 20:48:23 2022
type=PATH msg=audit(1654105703.179:2195): item=1 name="/home/user1" inode=401634 dev=fd:00 mode=0400
000 ouid=0 ogid=0 rdev=00:00 obj=unconfined_u:object_r:user_home_dir_t:s0 objtype=CREATE
type=PATH msg=audit(1654105703.179:2195): item=0 name="/home/" inode=8388794 dev=fd:00 mode=040755 o
uid=0 ogid=0 rdev=00:00 obj=system_u:object_r:home_root_t:s0 objtype=PARENT
type=CWD msg=audit(1654105703.179:2195): cwd="/home/grochowskia"
type=SYSCALL msg=audit(1654105703.179:2195): arch=c000003e syscall=83 success=yes exit=0 a0=7ff2172b
f050 a1=0 a2=7ff215932778 a3=5f656d6f685f7265 items=2 ppid=2344 pid=2661 auid=1000 uid=0 gid=0 euid=
0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=tty1 ses=1 comm="useradd" exe="/usr/sbin/useradd" subj=un
confined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="mkdir_rmdir"
----
time->Wed Jun 1 20:48:24 2022
type=PATH msg=audit(1654105704.949:2326): item=1 name="/home/user2" inode=8388789 dev=fd:00 mode=040
000 ouid=0 ogid=0 rdev=00:00 obj=unconfined_u:object_r:user_home_dir_t:s0 objtype=CREATE
type=PATH msg=audit(1654105704.949:2326): item=0 name="/home/" inode=8388794 dev=fd:00 mode=040755 o
uid=0 ogid=0 rdev=00:00 obj=system_u:object_r:home_root_t:s0 objtype=PARENT
type=CWD msg=audit(1654105704.949:2326): cwd="/home/grochowskia"
type=SYSCALL msg=audit(1654105704.949:2326): arch=c000003e syscall=83 success=yes exit=0 a0=7f451cc7
b050 a1=0 a2=7f451a022778 a3=5f656d6f685f7265 items=2 ppid=2344 pid=2666 auid=1000 uid=0 gid=0 euid=
0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=tty1 ses=1 comm="useradd" exe="/usr/sbin/useradd" subj=un
confined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="mkdir_rmdir"
```

Рис. 192. Записи из журнала аудита о создании именных каталогов пользователей user1 и user2

На рис. 192 видно, что в журнал аудита были добавлены записи о создании каталогов пользователей user1 и user2.

5. Создадим в каталогах пользователей user1 и user2 несколько файлов от имени самих пользователей. Посмотрим, записал ли аудит эти события.

```
time->Wed Jun 1 21:17:47 2022
type=PATH msg=audit(1654107467.889:7412): item=1 name="file1.txt" inode=401679 dev=fd:00 mode=010066
4 ouid=1001 ogid=1001 rdev=00:00 obj=unconfined_u:object_r:user_home_t:s0 objtype=CREATE
type=PATH msg=audit(1654107467.889:7412): item=0 name="/home/user1" inode=401634 dev=fd:00 mode=0407
000 ouid=1001 ogid=1001 rdev=00:00 obj=unconfined_u:object_r:user_home_dir_t:s0 objtype=PARENT
type=CWD msg=audit(1654107467.889:7412): cwd="/home/user1"
type=SYSCALL msg=audit(1654107467.889:7412): arch=c000003e syscall=2 success=yes exit=3 a0=7fffb350
87e a1=941 a2=1b6 a3=7fffb34e580 items=2 ppid=9363 pid=9386 auid=1000 uid=1001 gid=1001 euid=1001 s
uid=1001 fsuid=1001 egid=1001 sgid=1001 fsgid=1001 tty=tty1 ses=1 comm="touch" exe="/usr/bin/touch"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="open"
```

Рис. 193. Создание file1.txt пользователем user1

В первой строке видим запись о том, что файл file1.txt был создан пользователем user1. Посмотрим, создался ли файл file2.txt.

```

time->Wed Jun 1 21:17:51 2022
type=PATH msg=audit(1654107471.309:7416): item=1 name="file2.txt" inode=401680 dev=fd:00 mode=0100664
oid=1001 ogid=1001 rdev=00:00 obj=unconfined_u:object_r:user_home_t:s0 objtype=CREATE
type=PATH msg=audit(1654107471.309:7416): item=0 name="/home/user1" inode=401634 dev=fd:00 mode=040700
oid=1001 ogid=1001 rdev=00:00 obj=unconfined_u:object_r:user_home_dir_t:s0 objtype=PARENT
type=CWD msg=audit(1654107471.309:7416): cwd="/home/user1"
type=SYSCALL msg=audit(1654107471.309:7416): arch=c000003e syscall=2 success=yes exit=3 a0=7fff0ad4187e
a1=941 a2=1b6 a3=7fff0ad40e50 items=2 ppid=9363 pid=9387 auid=1000 uid=1001 gid=1001 euid=1001 s
uid=1001 fsuid=1001 egid=1001 sgid=1001 fsgid=1001 tty=tty1 ses=1 comm="touch" exe="/usr/bin/touch"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="open"

```

Рис. 194. Создание file2.txt пользователем user1

В первой строке видим запись о том, что файл file2.txt был создан пользователем user1. Посмотрим, создался ли файл file3.txt.

```

time->Wed Jun 1 21:17:55 2022
type=PATH msg=audit(1654107475.610:7420): item=1 name="file3.txt" inode=401681 dev=fd:00 mode=0100664
oid=1001 ogid=1001 rdev=00:00 obj=unconfined_u:object_r:user_home_t:s0 objtype=CREATE
type=PATH msg=audit(1654107475.610:7420): item=0 name="/home/user1" inode=401634 dev=fd:00 mode=040700
oid=1001 ogid=1001 rdev=00:00 obj=unconfined_u:object_r:user_home_dir_t:s0 objtype=PARENT
type=CWD msg=audit(1654107475.610:7420): cwd="/home/user1"
type=SYSCALL msg=audit(1654107475.610:7420): arch=c000003e syscall=2 success=yes exit=3 a0=7fff06fa087e
a1=941 a2=1b6 a3=7fff06f9fa20 items=2 ppid=9363 pid=9388 auid=1000 uid=1001 gid=1001 euid=1001 s
uid=1001 fsuid=1001 egid=1001 sgid=1001 fsgid=1001 tty=tty1 ses=1 comm="touch" exe="/usr/bin/touch"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="open"

```

Рис. 195. Создание file3.txt пользователем user1

В первой строке видим запись о том, что файл file3.txt был создан пользователем user1. Теперь рассмотрим, создалось ли файлы у пользователя user2.

```

time->Wed Jun 1 21:19:34 2022
type=PATH msg=audit(1654107574.955:7713): item=1 name="file1.txt" inode=9052087 dev=fd:00 mode=0100664
oid=1002 ogid=1002 rdev=00:00 obj=unconfined_u:object_r:user_home_t:s0 objtype=CREATE
type=PATH msg=audit(1654107574.955:7713): item=0 name="/home/user2" inode=8388789 dev=fd:00 mode=040700
oid=1002 ogid=1002 rdev=00:00 obj=unconfined_u:object_r:user_home_dir_t:s0 objtype=PARENT
type=CWD msg=audit(1654107574.955:7713): cwd="/home/user2"
type=SYSCALL msg=audit(1654107574.955:7713): arch=c000003e syscall=2 success=yes exit=3 a0=7fffa2c6d878
a1=941 a2=1b6 a3=7fffa2c6c0c0 items=2 ppid=9393 pid=9410 auid=1000 uid=1002 gid=1002 euid=1002 s
uid=1002 fsuid=1002 egid=1002 sgid=1002 fsgid=1002 tty=tty1 ses=1 comm="touch" exe="/usr/bin/touch"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="open"

```

Рис. 196. Создание file1.txt пользователем user2

В первой строке видим запись о том, что файл file1.txt был создан пользователем user2. Посмотрим, создался ли файл file2.txt.

```

time->Wed Jun 1 21:19:37 2022
type=PATH msg=audit(1654107577.722:7717): item=1 name="file2.txt" inode=9052088 dev=fd:00 mode=0100664
oid=1002 ogid=1002 rdev=00:00 obj=unconfined_u:object_r:user_home_t:s0 objtype=CREATE
type=PATH msg=audit(1654107577.722:7717): item=0 name="/home/user2" inode=8388789 dev=fd:00 mode=040700
oid=1002 ogid=1002 rdev=00:00 obj=unconfined_u:object_r:user_home_dir_t:s0 objtype=PARENT
type=CWD msg=audit(1654107577.722:7717): cwd="/home/user2"
type=SYSCALL msg=audit(1654107577.722:7717): arch=c000003e syscall=2 success=yes exit=3 a0=7fff1eb7e878
a1=941 a2=1b6 a3=7fff1eb7db40 items=2 ppid=9393 pid=9411 auid=1000 uid=1002 gid=1002 euid=1002 s
uid=1002 fsuid=1002 egid=1002 sgid=1002 fsgid=1002 tty=tty1 ses=1 comm="touch" exe="/usr/bin/touch"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="open"

```

Рис. 197. Создание file2.txt пользователем user2

В первой строке видим запись о том, что файл file2.txt был создан пользователем user2. Посмотрим, создался ли файл file3.txt

```
time->Wed Jun 1 21:19:40 2022
type=PATH msg=audit(1654107580.570:7721): item=1 name="file3.txt" inode=9052089 dev=fd:00 mode=01006
64 ouid=1002 ogid=1002 rdev=00:00 obj=unconfined_u:object_r:user_home_t:s0 objtype=CREATE
type=PATH msg=audit(1654107580.570:7721): item=0 name="/home/user2" inode=8388789 dev=fd:00 mode=040
700 ouid=1002 ogid=1002 rdev=00:00 obj=unconfined_u:object_r:user_home_dir_t:s0 objtype=PARENT
type=CWD msg=audit(1654107580.570:7721): cwd="/home/user2"
type=SYSCALL msg=audit(1654107580.570:7721): arch=c000003e syscall=2 success=yes exit=3 a0=7fffeb5a1
878 a1=941 a2=1b6 a3=7fffeb59fd90 items=2 ppid=9393 pid=9412 auid=1000 uid=1002 gid=1002 euid=1002 s
uid=1002 fsuid=1002 egid=1002 sgid=1002 fsgid=1002 tty=tty1 ses=1 comm="touch" exe="/usr/bin/touch"
subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key="open"
```

Рис. 198. Создание file3.txt пользователем user2

В первой строке видим запись о том, что файл file3.txt был создан пользователем user2. На этом перейдем к следующему пункту.

6. Создадим «секретный каталог», в котором хранятся «важные» файлы и попробуем зайти и прочитать эти самые файлы с пользователя user1.

```
[root@localhost /]# chmod 755 /secret
[root@localhost /]# ls -l
итого 28
lrwxrwxrwx. 1 root root 7 апр 16 12:31 bin -> usr/bin
dr-xr-xr-x. 4 root root 4096 апр 16 12:38 boot
drwxr-xr-x. 20 root root 3000 июн 1 19:46 dev
drwxr-xr-x. 75 root root 8192 июн 1 23:49 etc
drwxr-xr-x. 7 root root 80 июн 1 23:00 home
lrwxrwxrwx. 1 root root 7 апр 16 12:31 lib -> usr/lib
lrwxrwxrwx. 1 root root 9 апр 16 12:31 lib64 -> usr/lib64
drwxr-xr-x. 2 root root 6 июн 10 2014 media
drwxr-xr-x. 2 root root 6 июн 10 2014 mnt
drwxr-xr-x. 2 root root 6 июн 10 2014 opt
dr-xr-xr-x. 114 root root 0 июн 1 19:46 proc
dr-xr-xr-x. 2 root root 4096 апр 16 12:53 root
drwxr-xr-x. 21 root root 620 июн 1 20:37 run
lrwxrwxrwx. 1 root root 8 апр 16 12:31 sbin -> usr/sbin
drwxr-xr-x. 2 root root 6 июн 1 23:53 secret
drwxr-xr-x. 2 root root 6 июн 10 2014 srv
dr-xr-xr-x. 13 root root 0 июн 1 19:46 sys
drwxrwxrwt. 7 root root 88 июн 1 22:31 tmp
drwxr-xr-x. 13 root root 4096 апр 16 12:31 usr
drwxr-xr-x. 20 root root 4096 июн 1 19:46 var
```

Рис. 199. Настройка прав доступа в «секретном каталоге»

Меняются права доступа к каталогу /secret. Пользователь может записывать, читать и выполнять, группа может читать и выполнять, остальные могут читать и выполнять. Находясь под управлением пользователем user1, переходим в каталог /secret. Появится запись о том, что содержимое файла недоступно "warning" (невозможно писать и сохранять).

Теперь посмотрим, что записал аудит о действиях пользователя user1.

```
-----
type=PATH msg=audit(02.06.2022 00:15:58.942:13330) : item=0 name=file1 inode=401683 dev=fd:00 mode=file,644 ouid=root ogid=root rdev=00:00 obj=unconfined_u:object_r:default_t:s0 objtype=NORMAL
type=CWD msg=audit(02.06.2022 00:15:58.942:13330) : cwd=/secret
type=SYSCALL msg=audit(02.06.2022 00:15:58.942:13330) : arch=x86_64 syscall=getxattr success=no exit=-61(Нет доступных данных) a0=0x7ffffd801fba0 a1=0x7ff61c3b15db0 a2=0x0 a3=0x0 items=1 ppid=9825 pid=9843 auid=grochowska uid=user1 gid=user1 euid=user1 suid=user1 fsuid=user1 egid=user1 sgid=user1 fsgid=user1 tty=tty1 ses=1 comm=ls exe=/usr/bin/ls subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key=secret
-----
type=PATH msg=audit(02.06.2022 00:15:58.943:13336) : item=0 name=file2 inode=401684 dev=fd:00 mode=file,644 ouid=root ogid=root rdev=00:00 obj=unconfined_u:object_r:default_t:s0 objtype=NORMAL
type=CWD msg=audit(02.06.2022 00:15:58.943:13336) : cwd=/secret
type=SYSCALL msg=audit(02.06.2022 00:15:58.943:13336) : arch=x86_64 syscall=getxattr success=no exit=-61(Нет доступных данных) a0=0x7ffffd801fba0 a1=0x7ff61c3d23114 a2=0x7ffffd801fba0 a3=0x14 items=1 ppid=9825 pid=9843 auid=grochowska uid=user1 gid=user1 euid=user1 suid=user1 fsuid=user1 egid=user1 sgid=user1 fsgid=user1 tty=tty1 ses=1 comm=ls exe=/usr/bin/ls subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 key=secret
-----
```

Рис. 200. Отчет о действиях пользователя user в «секретном» каталоге secret

На данном рис. 200 видим, что система аудита записала вход пользователя user1 в файлы file1 и file2.

Таким образом, успешно была настроена и использована система аудита ОС Linux. С помощью демона auditd и утилит auditctl, ausearch, aureport был проведен аудит файлов и пользователей системы, введены правила, регистрирующие события, которые происходят в системе и записываются в журнал аудита.

ЗАКЛЮЧЕНИЕ

Аудит современной сетевой инфраструктуры предприятия может быть представлен двумя общими моделями: модель аудита, которая предполагает непосредственный доступ к компонентам КИС в границах локальной инфраструктуры, и модель аудита КИС с сегментами, включающими удаленные АРМ. Вторая общая модель технического аудита должна отражать ту специфику удаленной сети вне ДМЗ, к которой подключён АРМ, а также же учитывать стабильность сетевой среды АРМ, поскольку она частично контролируется или полностью неподконтрольна политике безопасности ДМЗ. Также в соответствии с типом исследования сетевой инфраструктуры выделяются две уникальные разновидности второй модели: модель аудита в инфраструктуре защищенного трафика и модель аудита вне инфраструктуры защищенного трафика. Учитывая своеобразие обследования всех видов сетевых узлов, при реализации модели уникального аудита можно выделить два основных этапа: первичный сетевой этап и сетевой этап с реализацией внутреннего локального доступа. На основе приведенных этапов определяется последовательность применения механизмов сбора данных, а также факторы, ограничивающие их функциональность. При этом следует учитывать фактор гарантии достоверности сведений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Аверченков, В.И. Аудит информационной безопасности: учеб. пособие для вузов / В.И. Аверченков. – 2-е изд., стер. – Брянск: БГТУ, 2010. – 268 с.

2. Галатенко А. Активный аудит// JetInfo [Электронный ресурс]. 1999. – 8(75) – URL: https://www.jetinfo.ru/wp-content/uploads/2021/04/1999_8.pdf (дата обращения: 29.01.2018).

3. Воробейкина И.В., Подтопельный В.В. Особенности использования фреймов для решения задач аудита информационной безопасности автоматизированных систем // В сборнике: Балтийский морской форум. Материалы IX Международного Балтийского морского форума: в 6 т.. Калининград, 2021. С. 430-434. – [Электронный ресурс]. – 2020. – URL: https://klgtu.ru/upload/science/bmf/bmf_2021/tom_2.pdf (дата обращения 17.05.2022)

4. Подтопельный В.В. Особенности формирования базы признаков для инструментов интеллектуального аудита информационной безопасности инфраструктуры АСУТП // В сборнике: Балтийский морской форум. Материалы IX Международного Балтийского морского форума: в 6 т. Калининград, 2021. – С. 440–446. – [Электронный ресурс]. – 2020. – URL: https://klgtu.ru/upload/science/bmf/bmf_2021/tom_2.pdf (дата обращения 17.05.2022).

5. Хватов Д.А., Ковтун А.И., Подтопельный В.В. Проблемы аудита информационной безопасности АСУТП // Вестник Балтийского федерального университета им. И. Канта. Серия: Физико-математические и технические науки. 2019. – № 4. – С. 67–75. – [Электронный ресурс]. – 2020. – URL: <http://elibrary.ru/item.asp?id=42686801>. (дата обращения 17.05.2022).

6. Подтопельный В.В. Проблемы аудита безопасности информационных систем с применением актуальной методики ФСТЭК // Известия Балтийской государственной академии рыбопромыслового флота: психолого-педагогические науки. ФГБОУ ВО «КГТУ». 2022 г. № 1 (59). – [Электронный ресурс]. – 2020. – URL: <http://elibrary.ru/item.asp?id=48218854> (дата обращения 17.05.2022).

7. Аверченков В.И., Рытов М.Ю., Гайнулин Т.Р. Оптимизация выбора состава средств инженерно-технической защиты информации на основе модели Клементса-Хоффмана // Вестник Брянского государственного технического университета. – Брянск, 2008. – № 1(17)

8. Большев А., Чербов Г., Черкасова С. Компоненты DTM: тайные ключи к королевству АСУТП // Исследовательский центр DigitalSecurity, 2014. – 36 с.

9. Горбачев И.Е., Глухов А.П. Моделирование процессов нарушения информационной безопасности критической инфраструктуры // Труды СПИИРАН. – Москва, 2015. – Вып. 1(38). – С. 112 – 135.

10. Котенко И.В. Многоагентные технологии анализа уязвимостей и обнаружения вторжений в компьютерных сетях // Новости искусственного интеллекта. – Москва, 2004. – № 1. – С. 56–72.

11. Макаренко С.И. Аудит информационной безопасности: основные этапы, концептуальные основы, классификация мероприятий // Системы управления, связи и безопасности, 2018. – №1.

12. Основы Windows PowerShell // Заметки IT специалиста URL: <https://info-comp.ru/sisadminst/546-windows-powershell-basics.html> (дата обращения: 25.05.21).

13. Введение в Windows PowerShell. – Спб.: БХВ–Петербург, 2009. – 464 с.

14. Документация по PowerShell // Microsoft URL: <https://docs.microsoft.com/ru-ru/powershell/> (дата обращения: 27.05.21).

Подтопельный Владислав Владимирович

**АУДИТ
ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ**

Учебное пособие
для студентов специальности 10.05.03
«Информационная безопасность
автоматизированных систем»
всех форм обучения

Часть 1

Редактор Н.В. Желтухина

Лицензия № 021350 от 28.06.99.

Редактор Г.В. Деркач

Печать офсетная.

*Специалист по компьютерной
правке И.В. Леонова*

Формат 60 x 90 1/16.

*Подписано в печать 12.04.2023 г.
Усл. печ. л. 10,8. Уч.-изд. л. 11,6.*

Заказ № 1825. Тираж 40 экз.

Доступ к архиву публикации и условия доступа к нему:
<https://lib.bgarf.ru/?LNG=&C21COM=F&I21DBN=IBIS&P21DBN=IBIS>

БГАРФ ФГБОУ ВО «КГТУ»

*Издательство БГАРФ,
член Издательско-полиграфической ассоциации высших учебных заведений
236029, Калининград, ул. Молодежная, 6.*