Всем добрый вечер, сегодня я хочу рассказать вам о моделях безопасности: их видах, концепциях, преимуществах и недостатках.

сразу хочу ввести немного терминов:   
объект – пассивная сущность, к которой обращаются  
субъект – активная сущность, которая обращается к объекту

Также перед началом доклада хочу предупредить о том, что я в процессе буду упоминать организации, в которых применяется та или иная модель безопасности. Когда я так говорю, я имею в виду, что они реализуются не в чистом виде – это просто неэффективно, а используют их как часть системы для выполнения определенного требования)

Начнем с модели ДАК – это дискреционная модель доступа. Ее суть заключается в том, что создаются списки объектов с указанием их пользователей. Права для объекта определяет его владелец: можно быстро выдавать и отзывать доступ – здесь все решают люди.   
вы с этой системой уже знакомы, работает она на базовой выдаче прав в формате rwx, о которой нам рассказывали на ОПД. Модели ДАК считаются самыми гибкими и удобными для пользователя, но за счет этого страдает безопасность.

Преимущества такой модели:  
1. Удобство для пользователя

2. гибкость

3. не требуется много усилий для поддержки и обслуживания

Недостатки:  
1. Не особо безопасны (проблема троянского коня)

2. сложно контролировать поток данных и права доступа

3. в больших системах сложно отследить кто кому что разрешил

Используется в операционках (юниксоподобных, винде), в google-drive, dropbox, MySQL, PostreSQL

MAC – мандатная модель доступа. Используется в основном в госструктурах, где уровни секретности объектов очень важны.

Каждому объекту и субъекту присваивается своя мандатная метка или мандатный уровень и возможны следующие исходы событий:

1. Мандатная метка субъекта равна мандатной метке объекта. В этом случае субъекту разрешено читать и изменять объект.
2. Мандатная метка субъекта выше мандатной метки объекта. Субъекту разрешено только читать объект: он его видит, но не может изменить.
3. Мандатная метка субъекта ниже мандатной метки объекта. Субъекту формально разрешено создать объект с более высокой мандатной меткой (так называемое «повышение уровня секретности объекта»). На практике у субъекта нет технической возможности для выполнения данной операции (он просто «не видит» изменяемый объект, например файл или каталог с файлами).

Важно: допускается повышение уровня доступа, но не понижение.

Также существуют категории для горизонтального разграничения прав доступа (например, между отделами)

Ограничения и особенности:  
1. Пользователи не могут определять доступ субъектов к объектам, этим занимается только администратор

2. если пользователь хочет изменить метку объекта, и он является автором, то он должен передать метку по наследству (от своего уровня секретности или от папки, где находится объект), просто так он ее придумать не может

3. тк используется с другими моделями безопасности то могут быть коллизии и будет сложно понять в каком именно слое произошел отказ доступа

4. сложности настройки и реализации,

Преимущества:  
1. Жесткий контроль

2. защита от внутренних угроз

3. многоуровневая безопасность(уровни конфиденциальности)

4. защита от троянских коней посредством сегментации на уровни

Недостатки:  
1. Отсутствие гибкости ВООБЩЕ

2. требует очень тонкой настройки

3. проблемы с масштабированием

4. неудобство для пользователя

5. не используется в коммерции

6. очень дорого

Используется госструктурами, банками, IT гигантами с ооооочень высокими требованиями к безопасности (гугл, амазон веб сервисы), росатом и другие серьезные ребята

Белл-лападула – модель, основанная на МАК.

Основной концепт: субъектам с низким уровнем безопасности запрещается читать объекты из высокого уровня безопасности, субъектам из более высокого уровня безопасности нельзя записывать что-либо в объекты более низкого уровня безопасности. Простыми словами: нет чтения вверх, нет записи вниз.

Преимущества модели:

1. защита от утечки данных

2. простота для понимания

Недостатки:  
1. Сложность реализации

2. неудобство

3. нельзя менять права доступа (особенно неудобно в случаях, если объект стал рассекреченным)

Основная цель модели, как и в МАК – конфиденциальность, так что считайте, что это инфа для общего развития, вряд ли вы будете ее применять в своих проектах, однако это, что называется, база.

Используется, например, минобороны США

Модель Биба – была введена для улучшения модели белл-лападула. Основное отличие от нее - это введение уровней целостности вместо уровней безопасности, и, соответственно, новых правил: нет чтения вниз и нет записи вверх.

Преимущества:   
1. Защита от изменения данных

2. также простота и строгость

Недостатки:  
1. Утечка данных (люди могут читать то что выше по уровню)

2. нет изменения прав доступа

Используют различные СУБД, например Oracle, Microsoft SQL server, а также старые версии SAP (система по управлению бизнес процессами).

Кларк-Вилсон:   
модель существенно отличается от вышеперечисленных. Ее созданию способствовал анализ методов управления ресурсов в организациях. В итоге мы имеем больше руководство разработчикам, чем конкретную модель, но моделью ее все-таки называют.

КЦ на рисунке – контроль целостности

Основная суть: модель строится на транзакциях, которые состоят из последовательности операций, переводящих объекты из состояния в состояние.

По факту у нас есть два непересекающихся подмножества CDI – ограниченные элементы данных и UDI – неограниченные элементы данных.

Также есть 9 правил, которые и определяют всю модель:  
1. В системе есть процедуры утверждения целостности, которые очевидно утверждают целостность любого CDI

2. применение любой процедуры преобразования к любому CDI должно

сохранять целостность этого CDI

3. только ПП может вносить изменение в cdi (Это обеспечивает замкнутость в пределах набора CDI.)

4. Субъекты могут инициировать только определенные ПП над определенными CDI(правило создает квм тройки, Например, если тройка является элементом отношения, то субъекту разрешается применить ПП к CDI. Если же эта тройка не является элементом отношения, то такой тип применения ПП будет запрещен. Это правило гарантирует, что всегда можно определить, кто может изменить CDI и как это изменение может произойти.)

5. Тройки должны проводить соответствующую политику разделения обязанностей субъектов (не позволять субъектам изменять cdi без соответствующего вовлечения других субъектов, это предотвращает субъектов от возможности нанести ущерб целостности cdi)

6. некоторые специальные ПП могут превращать UDI в CDI

7. каждое применение ПП должно регистрироваться в специальном CDI, в который может производиться только добавление данных для создания общей картины изменений

8. система должна распознавать субъекты, пытающиеся инициировать ПП(предотвращение механизма атак, где один CDI пытается выдать за другого)

9. изменения в списках авторизации доступны только специальным субъектам (ну стандарт безопасность чтоб непонятно кто не менял)

Преимущества модели:  
1. Контролируется не только целостность данных, но и целостность транзакций

2. разделение обязанностей (правило 5)

3. гибкость

4. защита от внутренних угроз (разделение обязанностей, контроль транзакций, верификация данных которая постоянно проводится, ограничение прямого доступа)

Недостатки модели:

1.сложность реализации

2. зависимость от правил целостности, которые нужно задавать заранее

3. неудобна в больших системах

Используется Amazon, Ebay, SAP

Модель РБАК – управление доступом на основе ролей.

Идея заключается в том, чтобы пользователь был ограничен необходимым минимумом (принцип наименьших привилегий), у каждой роли есть набор того, что она может делать, у человека может быть несколько ролей, при этом часто ставят ограничения на то, чтобы у человека не было противоречащих ролей.

Преимущества:  
1. Простота управления ролями

2. масштабируемость

3. поддержка иерархии ролей

4. уменьшение риска внутренних угроз (принцип минимальных привилегий, разделение обязанностей, динамическое управление ролями, аудит)

Недостатки:  
1. Не самая гибкая (если нужны уникальные права, то придется усложнить систему из этого вытекает еще и проблема с большим количеством ролей в системе)

2. нет контекстно зависимого доступа (не учитываются внешние условия, например время, местоположение, устройство)

3. сложно отслеживать действия

Используется Microsoft Active Directory, SELinux, СУБД Oracle, PostgreSQL 8.1, SAP R/3

Песочница

Существует два вида песочницы: традиционная и облачная

Суть работы песочницы:   
1. проводится сигнатурный анализ

2. источник документа проверяется по антиспам и антибот базам

3. ПО запускается в песочнице: там создается временная виртуальная машина, которая проверяет программу на подозрительную активность

4. если во время эмуляции замечена подозрительная активность, то песочница запускает ПО еще несколько раз с более расширенным функционалом проверки, чтобы в этом убедиться

5. отдельным пунктом хочется выделить ситуации, в которых вирус проявляется не сразу. Для таких моментов песочницы умеют ускорять время на ВМ.

Преимущества песочницы:  
1. Высокий уровень защиты от неизвестных угроз

2. изоляция опасных процессов

3. совместимость с другими моделями безопасности

Недостатки песочницы:  
1. Современные вирусы умеют определять, что они в песочнице, так что работает не всегда(

2. ресурсоемкость, в тч и финансовая

3. ложные срабатывания

4. не защищает от всех типов атак

5. сложность настройки и управления

Используется да почти во всех технологических гигантах и просто в крупных компаниях. У каждого из вас стоит автоматическая песочница на устройстве, будь то смартфон или ноутбук, поэтому вы наверное и слышали о песочнице до того, как я вам рассказала, в отличие от других моделей безопасности

АБАК – разграничение доступа на основе атрибутов.

У каждого объекта, субъекта, действия и среды(среда всм совокупность внешних условий) есть свои атрибуты, на этом и строится подход: у каждой ситуации есть свои атрибуты, сравнивая которые система решает давать ли доступ к объекту.

Преимущества:

1. динамическое управление доступом

2. масштабируемость

3. контекстно-зависимые правила (в отличие от рбак)

4. снижение администрирования (нет многомерного роста условий)

Недостатки:   
1. Требует доп инфраструктуры для управления атрибутами(системы аудита, сложные модули принятия решений)

2. медленная производительность (особенно если атрибутов много)

3. из за динамичности правил сложнее отслеживать аудиты

4. возможность создания конфликтов при переусложнении

5. требует обновления атрибутов при устаревании (постоянный контроль)

Используется платежными системами, облачными провайдерами (гугл клауд платформ, амазон веб сервисы), сап (для динамического контроля доступа к фин.данным), чутка юзается гос системами где доступ зависит от атрибута

Zero-Trust – никогда не доверяй, всегда проверяй

является наиболее популярной концепцией на данный момент.

В концепции фигурирует пять основных «частей»: данные, сети, нагрузка, пользователи, устройства

Приоритеты модели:

1. Защита данных
2. Необходимо сделать перемещение по сети наиболее сложным
3. Люди являются самым слабым звеном – нужно ограничивать, отслеживать и навязывать принципы получения доступа к данным
4. Под нагрузкой воспринимается весь стек используемых технологий – проверка безопасности должна быть на каждом этапе
5. Устройства также являются слабым звеном, поэтому их нужно отслеживать и сегментировать
6. Используется по большей части автоматизация, тк человек не способен уследить за таким количеством событий

Основные принципы:  
1. Требуется подтверждение для доступа к любым ресурсам «Вы должны рассматривать каждую попытку доступа к вашей сети как угрозу до тех пор, пока не подтверждено обратное»

2. использовать модель наименьших привилегий и контролировать доступ: необходимо ограничивать права доступа каждого до уровня, который необходим для выполнения служебных обязанностей, для этого рекомендуют использовать рбак, потому что ее хватает и для ограничения доступа, и для того, чтобы владельцы бизнеса могли управлять разрешениями

3. отслеживать все: принципы модели нулевого доверия в том, чтобы проверить вооооооообще все перед тем, как дать доступ. Также проводится логирование и аналитика безопасности данных для отслеживания.

Доп инфа: в процессе работы пользователя его данные постоянно перепроверяются.

Для модели зиро траст даже двухфакторная аутентификация считается слабой и рекомендуется использовать многофакторную аутентификацию

Преимущества:  
1. Повышенная защита от внутренних и внешних угроз

2. гибкая безопасность (адаптивна в зависимости от контекста, есть точечный контроль каждого запроса

3. минимизация ущерба при взломах (спасибо микросегментации)

4. соответствие гос стандартам ес, сша, рф

5. адаптивная аутентификация

6. экономия ресурсов (траты на ликвидацию последствий атак огромны у больших компаний)

7. упрощение управления доступом (можно выдать временный доступ, отозвать права в один клик)

8. ориентирована на развитие в будущем, планируется внедрение ИИ

Недостатки:

1. Высокая сложность внедрения
2. Дороговизна (решение вопроса обеспечитвают майкрософт зиро траст и гугл бейонд корп и это требует подписки)
3. Проблемы с пользовательским опытом, тк постоянно нужно подтверждать подлинность
4. Ложные срабатывания защиты от взлома при аномальном поведении
5. Требуется постоянное подключение к интернету для проверки данных
6. Старые приложения не поддерживают современные протоколы аутентификации (например сап r/3, 1С)
7. Чем сложнее правила, тем выше шанс ошибки
8. Юридические конфликты (например в россии есть закон, обязывающих хранить данные локально, но большинство зиро траст решений используют облака в сша (хотя если есть желание то варианты российских компаний тоже есть))
9. Человеческий фактор (происходят утечки данных при использовании личных почт, делятся токенами при аутентификации(смс коды например) и тп)

Используется в гугл, майкрософт, убер, минобороны сша, нетфликс, кока кола, тесла, спотик

Ну и будущие модели безопасности AI-Driven Control, Self-Healing Model

Оба подхода включают в себя использование ИИ для повышения контроля над системой и ее восстановления.

Необходимость в такой модели объясняется тем, что нападающие используют ИИ для создания вирусов и обхода защит. ИИ позволяет вирусам мутировать, адаптироваться, запускать координированные атаки, взаимодействовать с людьми как человек для повышенного успеха.

Также сейчас идут активные внедрения самовосстанавливающихся систем на основе ИИ. В целом, идея не новая, первые разработки появились еще в 00х в компании IBM, однако наиболее близко к грамотной реализации мы подобрались только сейчас и разработки в этой сфере идут крайне активно.

Работа с ИИ позволяет предупреждать атаки, когда работа с человеком основана на ликвидации последствий, что обходится намного дороже. Так еще и ликвидация с помощью ии обходится дешевле.

Как правило, ии вплетают в зиро траст модели, так проще + можно подключить модель на основе поведения в дополнение.

Суть стратегии при интеграции ии в модель безопасности состоит в том, чтобы

1. Ии использовался для выявления уязвимостей еще на этапе разработки, Ии изначально вплетен в систему, а не добавляется после всего
2. По мере развития системы, ии отслеживает и адаптирует протоколы безопасности

Преимущества такой модели:  
1. Обнаружение сложных и неизвестных угроз (0-day атаки)

2. автоматическая блокировка подозрительных действий

3. адаптивность (динамическая оценка рисков, самообучение, автоматическое реагирование на угрозы)

4. масштабируемость

5. снижение ложных срабатываний за счет контекстного анализа(смотрит условия в которых происходит действие)

6. быстрое восстановление в сравнении с человеком

Недостатки:   
1. Зависимость от качества данных

2. высокие вычислительные затраты

3. нейросети тупят и часто придумывают (ну вы заметили), поэтому не всегда могут пояснить, почему заблокировали доступ

4. что будет если ИИ взломают и дадут не те инструкции?

5. очень дорого

6. сложность интеграции в устаревшие системы

Вывод: ии не замена человеку (по крайней мере сейчас), модель нужно дорабатывать и проверять постоянно, потому что есть и плюсы, и минусы. + не стоит доверять полноценно, ИИ нуждается в постоянном обучении

Эта модель уже потихоньку внедряется в гугл, майкрософт, виза, тесла, сименс, агенство нац. Безопасности сша, сбер клауд и другие

Все, жду ваших вопросов:  
1. Вопрос по шестому правилу модели кларка вилсона: какие именно процедуры преобразования могут превращать UDI в СDI

ответ: например проверка и добавление данных в бд (получили данные от пользователя, преобразовали, добавили в бд – теперь данные подконтрольны системе), импорт данных из источников(пришли из внешнего файла, проверили, почистили от лишнего и добавили)

2. вопрос по модели мак: мандатная метка и мандатный уровень - чем они отличаются

Ответ: вообще это значение уровня доступа субъекта или уровень секретности объекта, отличаются тем, что мандатный уровень определяет только ранг, а мандатная метка определяет еще и полный набор атрибутов безопасности, например если бы у пекарен была МБ мак то было бы так: мандатный уровень «выпечка», мандатная метка «выпечка (вольчек 2025)»

3.чем отличается традиционная и облачная песочница

Ответ: традиционная это покупка настройка и содержание песочницы ложится на компанию, облачная это просто настройка подписки на сервисы, которые обеспечивают грамотную работу песочницы

4. как работает ускорение времени в песочнице?

Ответ: подмена ответов при вызове системного времени, основная сложность в том, чтобы подменить все значения, связанные с временем: выполнение процессов, задержки и тп, чтоб вирус не заметил, что вы ускорили выполнение.

5. что такое 0-day атаки?

Ответ: это неустраненные уязвимости или вирусы, для которых еще не разработаны защитные механизмы.

6. В чем заключается проблема троянского коня? (ну хз мб очевидный момент, но решила выделить в вопрос)

Ответ: в том, что пользователь может выдать права кому угодно – в тч и вредоносной программе, которая маскируется под безопасную.

7. Что такое аудиты?

Ответ: грубо говоря объект, в котором записываются действия в системе

8. модель на основе поведения – что это?

Ответ: проверка на «корректность» поведения пользователя, допустим пользователь работал обычно 5/2 с 9 до 18, а тут зашел в систему в вс в 3 часа ночи. Система обнаружит эту «подозрительную активность» и сообщит. Собственно, с таким подходом каждый из вас сталкивался)

9. Объясни, что за коллизии могут возникнуть в модели МАК

Ответ: например в мак все идет на уровнях безопасности, а в дак на выдаче прав, если их совместить то может быть коллизия

10. что такое многоуровневая безопасность (была упомянута в модели МАК)?

Ответ: многоуровневой безопасностью называются модели, в которых используются уровни безопасности, на чем и построена модель

11. что такое сигнатурный анализ в песочнице?

Ответ: сравнение характеристик проверяемого ПО с известными характеристиками вредоносных программ

12. какие конфликты могут возникать при переусложенении модели ABAC?

Ответ: противоречивые правила(разрешить доступ если менеджер, запретить, если ночь), перекрывающие условия(менеджер может редактировать факлы, все, кроме HR могут читать файлы – может ли менеджер читать?

13. что такое адаптивная аутентификация (была в модели zero-trust)

Ответ: в зависимости от степени риска ситуации определяется тип аутентификации: пароль + код, многофакторная аутентификация, биометрия + проверка поведения + код