# Управление уязвимостями

**Уязвимость** — это недостаток в информационной системе или ПО, который злоумышленник может использовать для проникновения в инфраструктуру, нарушения работы систем или доступа к ним. Уязвимости могут представлять различные уровни риска. Одним из широко упоминаемых и серьезных рисков является наличие эксплойта для уязвимости, особенно если им уже активно пользуются злоумышленники. Эксплойт - это вредоносная программа, которая содержит данные или исполняемый код, использующая уязвимости для проведения атаки.

Снизить риск от реализации угроз, обусловленных уязвимостью инфраструктуры, позволяет **управление уязвимостями (vulnerability management, VM)** — многоэтапный циклический процесс выявления, приоритизации и устранения уязвимостей с последующим контролем. В него также входит выбор способа реагирования на проблемы, возникающие с активами организации: обнаруженные уязвимости ПО, уязвимости конфигураций, небезопасно настроенные порты и другие уязвимости, которые могут быть использованы злоумышленниками. Основной целью данного процесса является минимизация рисков и защита систем от возможных атак, эксплойтов и других форм взлома или нарушений безопасности.

Управление уязвимостями состоит из следующих этапов:

1. Инвентаризация ИТ активов;

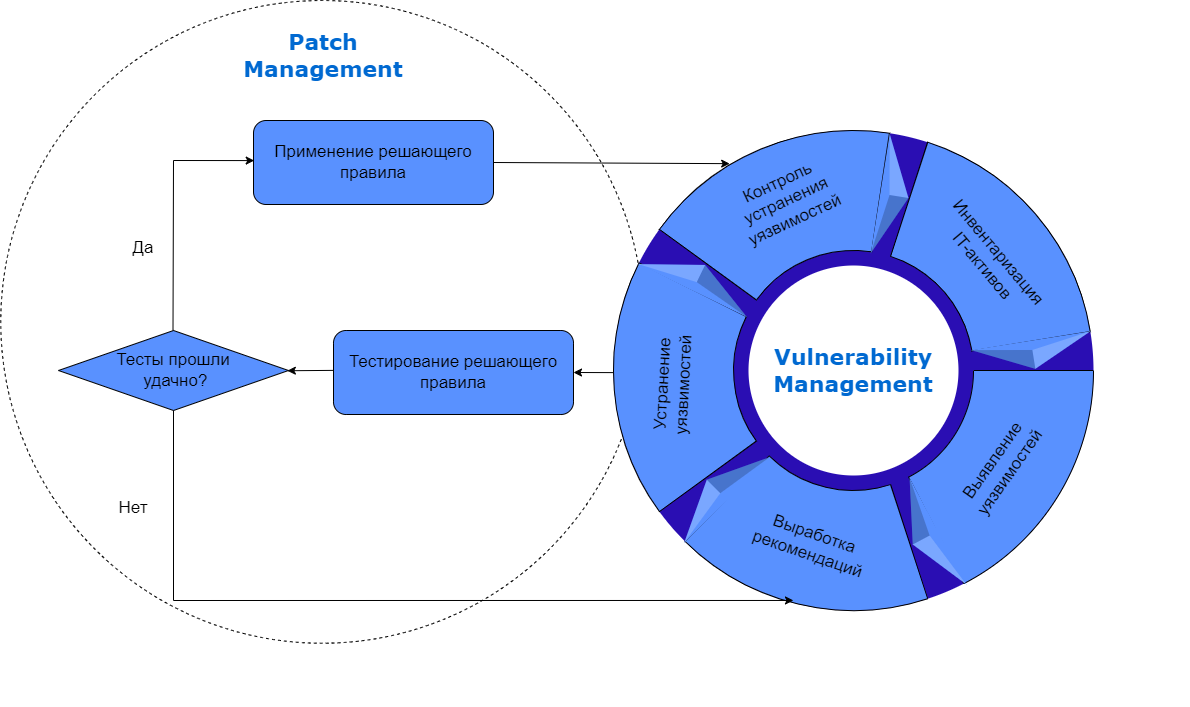
2. Выявление;

3. Приоритизация, выработка рекомендаций;

4. Устранение;

5. Контроль.

Процесс управления уязвимостями включает пять основных этапов



# Проблематика Vulnerability Management

### 1. Этап «Инвентаризация» ИТ активов

Фундаментом построения процесса управления уязвимостями является управление активами (asset management) - процесс планирования, учета и отслеживания состояния активов – элементов IT‑инфраструктуры, таких как оборудование, информационные системы, серверы и др. На этапе инвентаризации важно получить исчерпывающие сведения обо всех активах компании и понять, какие из них являются наиболее критичными и участвуют в значимых бизнес-процессах, какие хорошо защищены или, наоборот, больше всего уязвимы. Наличие информации обо всех активах сети позволит проводить наиболее полное сканирование, а значит, найти максимум всех брешей в защищаемой сети.

#### Проблематика и рекомендации:

Одна из основных проблем, с которыми организации могут столкнуться на этом этапе, – **отсутствие полной картины по уязвимостям из-за изменений в инфраструктуре,** где постоянно обновляются операционные системы и ПО, а также претерпевает изменения топология сети. Таким образом, защищаемые системы становятся уязвимыми, поскольку на их активах может быть большое количество уязвимостей, чье присутствие неизвестно.

Кроме этого, проблемой для компании может стать ситуация, при которой **в защищаемой сети произошли какие-то изменения, но специалисты ИБ о них не знают,** например:

* часть активов была перемещена в инфраструктуре и/или в них были изменены настройки, добавлены или удалены критичные сервисы, обновлено программное обеспечение или операционная система;
* произошло добавление новых неучтенных активов, в которых потенциально могут быть уязвимости. Такие активы нужно проверить и желательно несколькими режимами сканирования;
* часть активов выведена из эксплуатации, и их сканирование более не требуется.

Для того чтобы все было учтено, необходим контроль за изменениями в инфраструктуре. Необходимо начать **процесс управления активами** с первоначального наполнения базы активов, а затем ее обогащения и поддержания сведений в актуальном состоянии. Далее можно автоматизировать процесс или дополнять информацию об активах, интегрируясь с разнообразными системами. Налаженный процесс управления активами станет основой для построения процесса Vulnerability Management.

### 2. Этап «Выявление»

Выявление – это процесс обнаружения и анализа возможных уязвимостей в системе. Для выявления чаще всего используются сканеры (Vulnerability Scanner) - программы, осуществляющие поиск и анализ уязвимостей в инфраструктуре. Сканер уязвимостей проверяет сети, операционные системы, подключённые устройства и порты, анализирует все активные процессы и запущенные приложения и создает отчёты, в которых описывается, где и какие уязвимости были найдены. В этап выявления входит не только сканирование, но и проверка инфраструктуры на наличие конкретных BDU/CVE уязвимостей.

#### Проблематика и рекомендации:

Некоторые сканеры определяют наличие уязвимостей на самом хосте. Для этого база уязвимостей копируется на каждый хост, что сильно **увеличивает время сканирования**. Проблема усиливается в разы, если сеть большая и имеет сложную структуру. Вторичным фактором может являться низкая скорость передачи данных или проблемы соединения.  Помимо этого, **сканеры уязвимостей иногда могут нарушить работу сетей и систем, которые они сканируют.** В процессе работы сканер может отправлять специально сформированные запросы или пакеты на целевую систему, чтобы проверить ее на наличие уязвимостей. Это влечет за собой нарушение нормальной работы системы или сети.

В то же время, **сканеры не всегда верно определяют уязвимости** и часто выдают отчеты со множеством уязвимостей, которые на практике неэксплуатируемы, но формально их уровень риска высок. В итоге специалисты ИБ получают отчёт, который сложно интерпретировать, также им непонятно, какие уязвимости нужно исправлять в первую очередь. Кроме этого, сканеры не дают достаточно информации о способах устранения уязвимостей, поскольку в их базе знаний зачастую находятся лишь общие данные об уязвимости и способах её исправления. На практике их не хватает, и часто приходится проводить исследования на тему того, каким патчем закрыть ту или иную уязвимость.

Выявление уязвимостей — это динамический процесс, который требует постоянного обновления и адаптации к новым угрозам и методам атак. Поэтому для задач сканирования нужно в обязательном порядке настроить периодичность запуска. Только тогда можно быть уверенным в актуальности данных об инфраструктуре на приемлемом уровне. Для получения полной картины по уязвимостям можно использовать несколько источников данных, но стоит принять во внимание, что возможны дубликаты одних и тех же уязвимостей, обнаруженных разными сканерами.

### 3. Этап «Приоритизация. Выработка рекомендаций»

**3.1 Приоритизация.**

Информацию, которую специалисты ИБ получили из отчета сканера, нужно приоритизировать: определить, какие уязвимости следует устранять в первую очередь, а какие поставить в регулярный план на устранение. Уязвимости могут быть сгруппированы по степени их критичности и потенциального воздействия на систему: по CVSS (наиболее популярный способ приоритизации), по наличию эксплойта, нахождению уязвимости на критичных активах, по рейтингу уязвимости и уровню критичности.

**3.2 Выработка рекомендаций**

При выработке рекомендаций стоит выделить две сущности - mitigations (набор смягчающих решений) и официальное решение от производителя (разработчика).

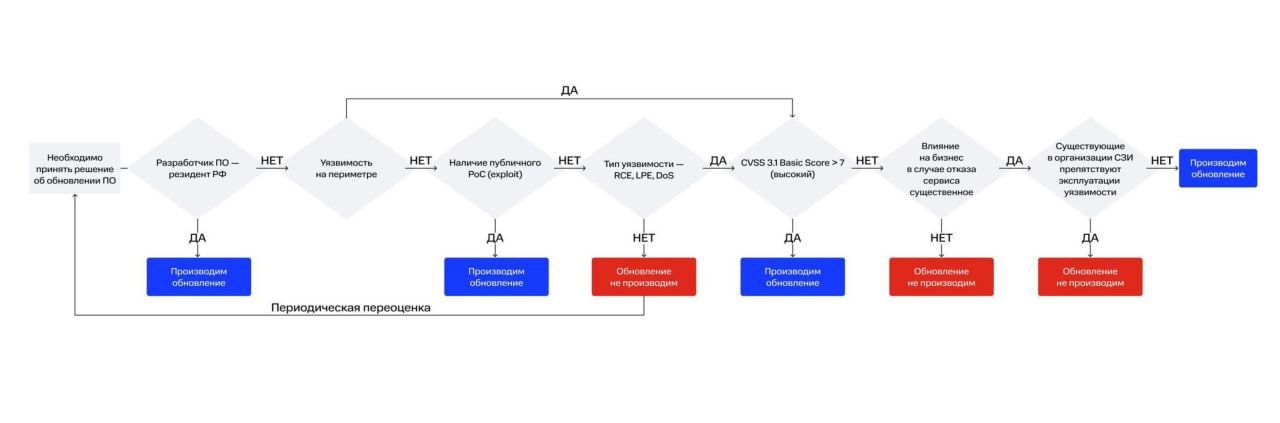
У обеих сущностей есть свои плюсы и минусы.

Mitigations - не всегда позволяют полностью закрыть уязвимость на ваших активах, однако снижают риск её эксплуатации.

Официальное решение, в свою очередь, почти всегда подразумевает под собой обновление программного обеспечения, операционной системы, прошивки и т.д., **НО** тут нужно устанавливать обновление

**Рекомендации регулятора:**

[НКЦКИ предлагает](https://safe-surf.ru/specialists/news/678042/) алгоритм принятия решения по обновлению критичного ПО, не относящегося к open-source:



Специалист, который будет выдавать рекомендации по закрытию уязвимости в организации должен оценить все возможные риски, которые они за собой повлекут. Это самый трудоемкий и требующий высокой квалификации этап, от рекомендаций специалиста будут зависеть безопасность информации (состояние защищенности) и непрерывность работы (отказоустойчивость) организации.

Безопасность информации, достигается закрытием возможных векторов компьютерных атак на организацию и повышением грамотности персонала в вопросах этики информационной безопасности, т.к. человеческий фактор может привести к реализации компьютерных атак (подкуп сотрудника организации злоумышленниками, фишинговые письма и т.д.).

#### Проблематика и рекомендации:

Количество выявленных уязвимостей может достигать сотни тысяч, а иногда и миллионы, из-за чего возникает **проблема определения их приоритета**.  Некоторые уязвимости являются критичными и требуют немедленного решения, в то время как другие могут быть менее значимыми, и их устранение может быть отложено на более поздний срок. Отсутствие четкой системы приоритизации приводит к неправильному фокусу для устранения уязвимостей и, соответственно, неправильному распределению ресурсов для их устранения. Чтобы избежать подобных трудностей, необходимы способы автоматизированного ранжирования уязвимостей по настраиваемым критериям. Таким образом, существенная часть аналитики будет проводиться с помощью VM‑системы, а ИБ‑специалисты смогут использовать для анализа уже подготовленные данные.

Кроме этого, далеко не во всех VM решениях имеется функционал гибкой настройки статусной модели уязвимостей, а традиционные статусы не всегда позволяют отразить весь бизнес-процесс работы по управлению ими. Поэтому у специалистов безопасности **нет возможности учитывать компенсирующие меры и исключать уязвимости как нерелевантные**.  Существенно сократить количество рутинной работы специалистов позволит наличие статусов, отражающих текущее состояние работы с конкретными уязвимостями, на основе которых можно бы было построить политики массовой обработки уязвимостей по утвержденным сценариям. При этом исключение уязвимостей из обработки дает возможность сконцентрировать внимание только на релевантных, актуальных для текущей сети уязвимостях.

Когда у организации есть данные инвентаризации активов, уязвимости выявлены и приоритизированы, можно перейти к их устранению.

### 4. Этап «Устранение»

Этап устранения – это принятие соответствующих мер для нейтрализации обнаруженных уязвимостей. Оно заключается в установке патчей/обновлений, остановке или отключении тех или иных служб и протоколов, в выполнении компенсирующих мер или принятии риска по не устранению. Принятые решения по обнаруженным уязвимостям, как правило, обозначаются в системе установкой соответствующего статуса (например, Компенсирующие меры, Риск принят или Ложное срабатывание). Это нужно для оценки дальнейших действий по уязвимостям и контроля их устранения.

Если принято решение об устранении уязвимости, то формируется задача в ИТ‑отдел и запускается патч‑менеджмент – процесс управления и применения патчей, обновлений и исправлений программного обеспечения.

#### Проблематика и рекомендации:

Как уже было сказано, уязвимости могут быть исправлены с помощью обновлений и патчей или изменением конфигурационных файлов, **однако их установка часто является сложной процедурой.** Это может быть вызвано несовместимостью с другими программами, необходимостью тестирования перед установкой или отсутствием доступа к системе для обновления.

Определенно, в инфраструктуре любой организации найдутся активы, ПО и сервисы, которые сложно патчить. И дело не в самой процедуре, а в зависимостях, одна за одной выводящих систему из строя и останавливающих бизнес-процесс.

Снизить риск поможет предварительное тестирование накатки патчей, но позволить себе такое может не каждая организация. Причина, как правило, в нехватке ресурсов: тестирование обновлений трудоемко и требует нетривиальных знаний и опыта.  Тем не менее, если нет возможности развернуть тестовую среду и найти ресурсы для тестовых обновлений и их анализа, можно провести ряд мер по структуризации информации об активах для более точечного распределения сил. Для начала сформировать список ОС и ПО с разделением списка на те, которые можно патчить автоматически, и те, которые можно обновлять только вручную, а также на те, в которые можно вносить изменения только после предварительных тестов и в профилактические окна, чтобы минимизировать риск нештатных ситуаций в рабочее время. Отдельно нужен список взаимозависимостей ПО, чтобы учитывать их при планирующихся обновлениях.

Кроме этого, острой «болью» также является **уровень детализации в решениях по устранению уязвимостей в отчетах от сканеров.** ИБ‑специалист получает из отчёта вендора информацию, содержащую краткие процедуры исправления уязвимостей, но очень часто этого недостаточно, и приходится проводить дополнительное исследование, а также добавление контекста по накатке нужной версии патча или же передавать отчет как есть. Если времени на дополнительную аналитику нет, то в итоге сотрудники ИТ-отдела обычно получают в заявках на устранение неструктурированный перечень уязвимостей и оборудования, на котором они обнаружены. Такую информацию анализировать и приоритизировать сложно, и у ИТ‑специалистов уходит много времени на обработку таких заявок, из-за чего могут быть нарушены сроки по их устранению. Особенно это критично при устранении уязвимостей с публичным эксплойтом.

Для облегчения жизни ИТ‑специалистам желательно в каждом отчете предоставлять сгруппированную информацию о патчах/обновлениях, о том, какую версию на какой актив поставить (а лучше - и где эту версию взять).

В то же время далеко **не всегда в организации присутствует четкий регламент процесса патч-менеджмента.** Однако, это тот самый жизненно необходимый процесс, от которого будет зависеть уровень защищенности периметра компании, а также всех остальных систем. Поэтому прежде чем заниматься уязвимостями, нужно организовать процесс патч-менеджмента, или, иными словами, организовать взаимодействие с ИТ-подразделением. В идеальной ситуации подразделения ИБ и ИТ плотно взаимодействуют и помогают друг другу. Специалисты обоих отделов, например, могут сверять графики патч-менеджмента по сегментам ПО или типам ОС, договариваться о будущих исправлениях и изменениях в сети. Желательно, если специалист ИБ будет участвовать в согласовании изменений в сети и учитывать их при поиске уязвимостей в дальнейшем. И если процесс участия ИБ в согласовании выстроить не удастся, то хотя бы своевременно получать информацию об изменениях в ИТ‑инфраструктуре.

Немаловажной частью является непрерывный процесс патчинга систем ИТ‑специалистами без каких-либо заявок извне. Когда в защищаемой сети будут на регулярной основе ставиться обновления на все возможные системы, то уязвимостей будет гораздо меньше. К тому же, установка патчей по расписанию для ИТ - гораздо удобнее и понятнее, чем закрытие большого списка задач от ИБ по устранению уязвимостей.

### 5. Этап «Контроль»

Этап контроля включает в себя проверку устранения уязвимостей и соблюдения сроков, а также дальнейшее отслеживание состояния инфраструктуры для выявления новых уязвимостей. Еще в этот этап входит анализ причин отклонений и неисполнений заявок на устранение и разработка на их основе решений по улучшению процесса.

После получения результатов последующих сканирований можно оценить количественные показатели по уязвимостям: сколько устранено, сколько найдено новых, какие уязвимости остались неустранёнными.

#### Проблематика и рекомендации:

Многие VM‑системы часто не имеют интеграций с системами Service Desk, а значит, автоматизированно **проверять и контролировать статусы и сроки устранения уязвимостей довольно сложно или вовсе нет возможности.**  Поэтому многие ИБ-специалисты не проверяют результаты патч-менеджмента, а смотрят результаты последующих сканирований: неустраненные уязвимости снова попадут в задачу на устранение. Однако, статусы этапов обработки уязвимостей в каждый момент времени и общая статистика по срокам устранения остаются неизвестными.

Другая проблема может заключаться **в отсутствии инструментов сопоставления результатов сканирования.** Многие сканеры хранят все отчеты о сканированиях отдельно, и сопоставлять результаты прошлых и последующих сканирований приходится вручную или же писать какую-то автоматизацию своими силами.

Для решения таких задач нам помогут VM‑решения с активоцентричным подходом, способные сохранять историю появления и устранения уязвимостей на каждом активе, чтобы затем отслеживать актуальное состояние и динамику по процессу при помощи дашбордов и построения отчетов.

# Заключение

Проблематика управления уязвимостями требует систематического подхода и использования эффективных инструментов для обнаружения, приоритизации и исправления уязвимостей.  Проблем много, они встречаются на каждом этапе процесса VM. Это говорит о его сложности и объясняется нетривиальными подходами к детектированию, многообразием систем в инфраструктурах компаний, а также недремлющими злоумышленниками, не оставляющими попыток воспользоваться эксплойтами и нанести урон компаниям.

Кроме этого, еще раз стоит обратить внимание на то, что управление уязвимостями - не разовая операция, а непрерывный цикличный механизм по комплексной защите ИТ активов. Наладить такой механизм возможно, но для этого нужно прежде всего соблюдать две рекомендации:

* построить ресурсно-сервисную модель активов своей сети, используя инвентаризацию, и постоянно поддерживать ее в актуальном состоянии проводя сканирования по расписанию
* согласовать и наладить регулярный процесс патч-менеджмента с ИТ‑департаментом

Далее - отслеживать изменения при каждом последующем сканировании, делать анализ случившихся отклонений и ложных срабатываний, корректировать по их результатам профили сканирования, закладываемое время на патчинг, где-то принимать риски, а где-то, наоборот, перезакладываться и усиливать защиту. Только каждодневное решение проблем и сложившиеся свой опыт и опыт коллег из ИТ позволят выстроить единую мощную стратегию противодействия угрозам.

Упростить и автоматизировать этот процесс позволяют специализированные системы по управлению уязвимостями, направленные на повышение скорости, эффективности и интеграции Vulnerability Management в организациях.

# Реализация эффективного vulnerabilty management на примере DigitalOcean

1. При выявлении новой уязвимости ей присваивается определенный уровень критичности.
2. В зависимости от уровня критичности уязвимости определяется рекомендуемое время для ее устранения. Это время известно как «Accepted Insecure Time» — период, в течение которого уязвимость должна быть устранена, чтобы не накапливать долг.
3. Если уязвимость не устранена в течение рекомендованного времени, начинает накапливаться долг по безопасности. Долг рассчитывается ежедневно: за каждый день, в течение которого уязвимость остается нерешенной после истечения срока, к общему долгу добавляется определенная величина, зависящая от критичности уязвимости. Чем выше критичность, тем быстрее накапливается долг.
4. Команды могут видеть свои показатели долга по безопасности через панель управления. Они могут анализировать, какие уязвимости вносят наибольший вклад в долг, и решать, на каких из них сосредоточить свои усилия в первую очередь.
5. Для каждой команды или бизнес-подразделения устанавливается определенный порог долга — максимально допустимая сумма накопленного долга по безопасности. Если долг превышает этот порог, это сигнализирует о необходимости устранить уязвимости, чтобы вернуть показатели в допустимые рамки.
6. Для всех бизнес-подразделений рассчитывается показатель соответствия порогу долга по безопасности (Security Debt Adherence), то есть процент команд, у которых долг по безопасности соответствует ожидаемому уровню (или, иначе говоря, установленному уровню обслуживания (SLO)). Вначале DigitalOcean достигали 75% и постепенно продвигались к цели в 95%. Этот порог является согласованным компромиссом между бизнес-подразделением и командой безопасности относительно того, насколько гибким может быть график выполнения работ по безопасности. Если ускорение разработки продукта, критическая инициатива, инцидент безопасности или другой фактор изменяет верхнеуровневую оценку этой гибкости, DigitalOcean может повысить или снизить порог долга для соответствующей части организации.