Рекомендации по обновлению иностранного программного обеспечения

Компании из недружественных государств (IBM Corporation, Cisco Systems, Fortinet и другие компании) официально заявили о приостановлении в Российской Федерации реализации и технической поддержки продукции. Указанные компании закрыли доступ для российских потребителей к репозиториям обновлений и базам сигнатур, обновления и лицензионная политика, предоставляемые данным компаниями, привели к прекращению функционирования оборудования или к невозможности выполнения отдельных функций безопасности.

Одновременно отмечаются факты внедрения через обновления иностранного программного обеспечения, разработанного в недружественных Российской Федерации государствах, деструктивного функционала и вредоносного программного обеспечения.

В целях недопущения нарушения функционирования иностранных систем и автоматизированных систем управления органов государственной власти и субъектов критической информационной инфраструктуры Российской Федерации в соответствии с рекомендациями ФСТЭК России приняты меры по приостановке работы по обновлению программного обеспечения.

Вместе с тем в указанном программном обеспечении выявляются критические уязвимости, которые могут быть использованы зарубежными хакерскими группировками для реализации компьютерных атак в отношении информационных систем органов государственной власти и объектов критической информационной инфраструктуры Российской Федерации.

В целях обеспечения защиты информации в информационных системах и устойчивого функционирования объектов критической информационной инфраструктуры, в том числе функционирующих на базе информационно-телекоммуникационной инфраструктуры (далее - информационные (автоматизированные) системы и (или) информационная инфраструктура) имеются потребности устранения уязвимостей путем обновления иностранного программного обеспечения.

Полагаем целесообразным при обновлении указанного программного обеспечения руководствоваться следующим порядком:

1. Необходимо определить уровень критичности уязвимости программного обеспечения для конкретных информационных (автоматизированных) систем и (или) информационной инфраструктуры в соответствии с **прилагаемым Временным порядком**.

2. В случае если по результатам применения *Временного порядка* уязвимости присвоен критичный или высокий уровень, рекомендуется принять решение об обновлении программного обеспечения.

3. Получить обновление программного обеспечения из официального источника (сайты разработчиков и дистрибьютеров, сайты международных сообществ открытого программного обеспечения). Произвести проверку электронных цифровых подписей и хеш-функций скаченных файлов с обновлениями и проверить актуальный статус цепочки сертификатов (при возможности). Необходимо учитывать, что обновление программного обеспечения также может устранять иные уязвимости программного обеспечения.

4. Перед установкой обновления программного обеспечения в информационную (автоматизированную) систему и (или) информационную инфраструктуру необходимо провести следующие мероприятия по его тестированию:

4.1. Для установки обновления программного обеспечения обеспечить создание и настройку изолированной тестовой среды, максимально приближенной к условиям функционирования информационной (автоматизированной) системы и (или) информационной инфраструктуры, в том числе с использованием средств виртуализации (при возможности).

4.2. Если создание тестовой среды невозможно, рекомендуется выделить для тестирования обновления наименее критичный для функционирования сегмент информационной (автоматизированной) системы и (или) информационной инфраструктуры.

4.3. В случае невозможности выделить для тестирования обновления тестовую среду предусмотреть аварийное отключение сегмента информационной (автоматизированной) системы, в котором будет применено обновление, от других сегментов системы (информационной инфраструктуры).

4.4. До начала обновления провести резервное копирование тестовой среды для возможности оперативного восстановления.

4.5. Настроить расширенное журналирование изменений и событий в тестовой среде.

4.6. Настроить системы мониторинга информационной безопасности и корреляции событий на уровень повышенной чувствительности (при их наличии).

4.7. Обновление программного обеспечения необходимо проверить на предмет наличия в нем вредоносного программного обеспечения с применением сертифицированных средств антивирусной защиты.

4.8. После установки обновления программного обеспечения в тестовой среде необходимо:

- оценить работоспособность программного обеспечения и доступность оборудования, на которое оно устанавливалось;

- при возможности запустить обновление программного обеспечения в замкнутой среде предварительного выполнения программ (в «песочнице» в режиме блокировки объекта в информационной системе до получения вердикта на нескольких инсталляциях со смещенными относительно текущих датами, временем и локализацией);

- провести мониторинг обращений обновленного программного обеспечения к компонентам информационной системы и (или) информационной инфраструктуры;

- провести контроль целостности программного обеспечения тестовой среды по наличию имен (идентификаторов) компонентов программного обеспечения и (или) по контрольным суммам после установки обновления и динамически в процессе тестирования обновления программного обеспечения;

- провести контроль состава программного обеспечения тестовой среды после установки обновления программного обеспечения в части появления нового программного обеспечения;

- провести контроль сетевого трафика из тестовой среды в сеть «Интернет» или иные информационные (автоматизированные) системы (при наличии такого доступа в тестовой среде).

4.9. При проведении тестирования обновлений программного обеспечения с открытым исходным кодом дополнительно рекомендуется провести мероприятия в соответствии с **прилагаемыми Рекомендациями**

по проведению тестирования (исследований) обновлений программного обеспечения с открытым исходным кодом на предмет наличия недекларированного функционала.

5. Обеспечить возможность восстановления функционирования информационных (автоматизированных) систем и (или) информационной инфраструктуры вследствие нарушения их функционирования после обновления программного обеспечения путем реализации следующих мероприятий:

5.1. Организовать или убедиться в актуальности уже существующих процессов по созданию регулярных образов информационных (автоматизированных) систем средствами виртуализации или системами хранения данных с достаточной глубиной хранения для возможности возвращения состояния системы до нарушения функционирования или выхода из строя. Образ информационной (автоматизированной) системы необходимо сохранять перед каждым обновлением программного обеспечения.

5.2. Для каждой обновляемой информационной (автоматизированной) системы необходимо создавать резервные копии данных, обрабатываемых в системе, путем выгрузки (экспорта) содержимого средствами этой системы для возможности восстановления работоспособности системы. Для восстановления информационной (автоматизированной) системы необходимо будет установить ее последнюю работоспособную версию данных из дистрибутива, а данные загрузить из резервной копии.

5.3. Необходимо создать независимую резервную копию информационной (автоматизированной) системы путем выгрузки информации из обновляемой системы для ее восстановления путем загрузки в аналогичной системе. Например, копии страниц, файлов, данных в форматах CSV, архивов и другой информации.

5.4. Создать локальную копию дистрибутивов, используемых информационной (автоматизированной) системой и (или) информационной инфраструктурой.

5.5. Резервные образы, копии данных, архивы и дистрибутивы программного обеспечения информационной (автоматизированной) системы следует хранить на отдельных от системы устройствах.

5.6. При создании или обновлении резервных копий по возможности рекомендуется проверить на предмет шифрования данных вирусом-шифровальщиком.

6. После установки обновления программного обеспечения необходимо реализовать следующие мероприятия, направленные на мониторинг работоспособности информационной (автоматизированной) системы и (или) информационной инфраструктуры (при наличии технической возможности):

6.1. Запретить по возможности исходящие соединения с сетью «Интернет» для сегментов информационной (автоматизированной) системы или отдельных серверов, в которых производилось обновление программного обеспечения. При невозможности необходимо до процедуры обновления программного обеспечения составить профиль сетевой активности, а после обновления обеспечить мониторинг отклонений от этого профиля.

6.2. Организовать мониторинг процессов функционирования обновленного программного обеспечения. Особое внимание необходимо уделить создаваемым обновленным программным обеспечением процессам, создать эталонный профиль активности процессов указанного программного обеспечения и периодически отслеживать изменения этого профиля.

6.3. Организовать мониторинг сетевой активности обновленных компонентов. Создать эталонный профиль сетевой активности этих компонентов до обновления и периодически анализировать отклонения текущего профиля от эталонного.

6.4. Организовать при наличии возможности мониторинг событий безопасности в информационной (автоматизированной) системе и (или) информационной инфраструктуре, целью которого является обнаружение возможного перемещения злоумышленника через закладки, предоставляющие удаленный доступ злоумышленнику в систему, появившейся с обновлением программного обеспечения. Для этого необходимо обеспечить мониторинг и анализ внутреннего трафика, сбор и хранение файлов, журналов событий информационной (автоматизированной) системы и (или) информационной инфраструктуры.

6.5. В случае выявления аномалий на предыдущих шагах исследования или наличия рекомендаций разработчика обновляемого программного обеспечения рекомендуется отключить средства антивирусной защиты информации и другие средства защиты информации для выявления

вредоносной активности, необходимо тщательный анализ вызывающих срабатывания модулей обновляемого программного обеспечения путем декомпиляции бинарного кода на предмет наличия в нем недекларированного функционала.

Отмечаем, что принятия решения об обновлении программного обеспечения без принятия указанных мер или при их реализации не в полном объеме создает существенные риски для функционирования информационных (автоматизированных) систем и (или) информационной инфраструктуры.

Дополнительно сообщаем, что в случае невозможности получения и установки обновлений программного обеспечения необходимо приять компенсирующие меры защиты информации с учетом архитектуры и особенностей функционирования информационных (автоматизированных) систем и (или) информационной инфраструктуры, а также особенностей эксплуатации уязвимостей программного обеспечения.

I. Временный порядок определения уровня критичности уязвимостей программного обеспечения в информационных (автоматизированных) системах

1. Общие положения

1.1. Настоящий Временный порядок определения критичности уязвимостей программного обеспечения в информационных (автоматизированных) систем (далее – Порядок) разработан в соответствии с подпунктом 4 пункта 8 Положения о Федеральной службе по техническому и экспортному контролю, утверждённого Указом Президента Российской Федерации от 16 августа 2004 г. № 1085.

1.2. Порядок определяет содержание работ по определению критичности уязвимостей программного обеспечения, используемого в информационных (автоматизированных) системах федеральных органов исполнительной власти, органов государственной власти субъектов Российской Федерации, органов власти местного самоуправления, субъектов критической информационной инфраструктуры Российской Федерации, в том числе функционирующих на базе информационно-телекоммуникационной инфраструктуры (далее – информационная инфраструктура).

1.3. Порядок применяется для определения приоритетов при принятии решения о необходимости устранения уязвимостей программного обеспечения, страной происхождения которого являются иностранные государства, совершающие в отношении России, российских компаний и граждан недружественные действия, утвержденные распоряжением Правительства Российской Федерации от 5 марта 2022 г. № 430-р, а также иного программного обеспечения.

1.4. Настоящий Порядок действует до утверждения ФСТЭК России методического документа, определяющего порядок определения критичности уязвимостей программного обеспечения в информационных (автоматизированных) систем.

2. Порядок определения критичности уязвимостей программного обеспечения в информационных (автоматизированных) системах

1. Определение уровня критичности уязвимостей проводится в целях обеспечения поддержки принятия оператором информационной (автоматизированной) системы и (или) информационной инфраструктуры решения о необходимости устранения уязвимостей.

2. Исходными данными для определения критичности уязвимостей являются:

а) база уязвимостей программного обеспечения, содержащаяся в банке данных угроз безопасности информации ФСТЭК России (bdu.fstec.ru), а также иные источники, содержащие сведения об известных уязвимостях;

б) официальные информационные ресурсы разработчиков программного

обеспечения и исследователей в области информационной безопасности;

в) документация на информационные (автоматизированные) системы и (или) информационную инфраструктуру (программная (конструкторская) и эксплуатационная документация, содержащая сведения о составе и архитектуре, о группах пользователей и уровне их полномочий, и типах доступа, о внешних и внутренних интерфейсах, а также иные документы);

г) результаты инвентаризации информационных (автоматизированных) систем и (или) информационной инфраструктуры, анализа уязвимостей в информационных (автоматизированных) системах и (или) информационной инфраструктуры.

Указанные исходные данные могут уточняться или дополняться с учетом особенностей области деятельности, в которой функционируют информационные (автоматизированные) системы.

3. Определение критичности уязвимостей программного обеспечения проводится подразделением по защите информации (отдельными специалистами, назначенными ответственными за обеспечение защиты информации (обеспечение безопасности)) обладателя информации или оператора с участием подразделений или специалистов, ответственных за эксплуатацию систем и сетей (ИТ-специалистов, специалистов

автоматизированных систем управления, специалистов связи и другие специалисты).

4. Определение критичности уязвимостей программного обеспечения применительно к конкретной информационной (автоматизированной) системе и (или) информационной инфраструктуре включает следующие этапы:

1) определение наличия в информационной (автоматизированной) системе и (или) информационной инфраструктуре подверженного уязвимости программного обеспечения;

2) определение расположения в информационной (автоматизированной) системе и (или) информационной инфраструктуре программного обеспечения, подверженного уязвимости (например, на периметре системы, во внутреннем сегменте системы, при реализации критических бизнес-процессов и других сегментах системы);

3) расчет уровня опасности уязвимости программного обеспечения применительно к конкретной информационной (автоматизированной) системе и (или) информационной инфраструктуре;

4) расчет показателя, определяющего влияние уязвимости программного обеспечения на функционирование информационной (автоматизированной) системы и (или) информационной инфраструктуре;

5) расчет уровня критичности уязвимости программного обеспечения в информационной (автоматизированной) системе и (или) информационной инфраструктуре V.

5. Расчет уровня критичности уязвимости программного обеспечения в информационной (автоматизированной) системе и (или) информационной инфраструктуре V осуществляется по следующей формуле:

$$V = I_{cvss} \times I_{infr}, \text{ где}$$

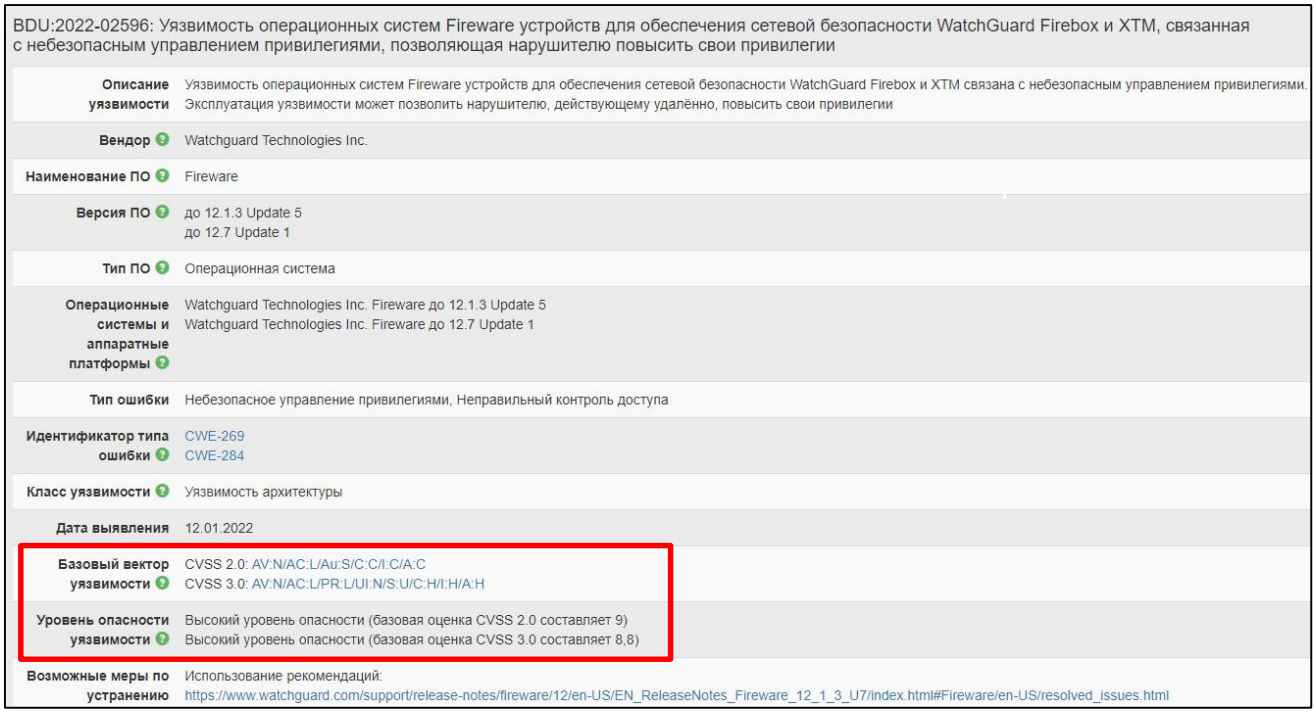
I_{cvss} – показатель, определяющий уровень опасности уязвимости, рассчитанный применительно к конкретной информационной системе и (или) информационной инфраструктуре по методике Common Vulnerability Scoring System 3.0 или 3.1;

I_{infr} – показатель, определяющий влияние уязвимости программного обеспечения на функционирование информационной (автоматизированной) системы и (или) информационной инфраструктуры.

6. Расчет показателя I_{cvss}, определяющего уровень опасности уязвимости, осуществляется в соответствии с методикой Common Vulnerability Scoring System 3.0 или 3.1. Показатель I_{cvss} определяется на основании описания уровня опасности уязвимостей программного обеспечения и входящих в его состав базовых, временных и контекстных метрик, путем их расчета применительно в конкретной информационной системе и (или) информационной инфраструктуре.

Исходные данные для расчета показателя I_{cvss} содержатся в описании уязвимостей, представленных в банке данных угроз безопасности информации

ФСТЭК России (bdu.fstec.ru), и иных источниках. Пример описания показателя I_{cvss}, рассчитанного по методике Common Vulnerability Scoring System 3.0 (CVSS 3.0) приведен на рисунке 1.



BDU:2022-02596: Уязвимость операционных систем Fireware устройств для обеспечения сетевой безопасности WatchGuard Firebox и XTM, связанная с небезопасным управлением привилегиями, позволяющая нарушителю повысить свои привилегии

Описание уязвимости	Уязвимость операционных систем Fireware устройств для обеспечения сетевой безопасности WatchGuard Firebox и XTM связана с небезопасным управлением привилегиями. Эксплуатация уязвимости может позволить нарушителю, действующему удалённо, повысить свои привилегии
Вендор	Watchguard Technologies Inc.
Наименование ПО	Fireware
Версия ПО	до 12.1.3 Update 5 до 12.7 Update 1
Тип ПО	Операционная система
Операционные системы и аппаратные платформы	Watchguard Technologies Inc. Fireware до 12.1.3 Update 5 Watchguard Technologies Inc. Fireware до 12.7 Update 1
Тип ошибки	Небезопасное управление привилегиями, Неправильный контроль доступа
Идентификатор типа ошибки	CWE-269 CWE-284
Класс уязвимости	Уязвимость архитектуры
Дата выявления	12.01.2022
Базовый вектор уязвимости	CVSS 2.0: AV:N/AC:L/Au:S/C:C/I:C/A:C CVSS 3.0: AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H
Уровень опасности уязвимости	Высокий уровень опасности (базовая оценка CVSS 2.0 составляет 9) Высокий уровень опасности (базовая оценка CVSS 3.0 составляет 8,8)
Возможные меры по устранению	Использование рекомендаций: https://www.watchguard.com/support/release-notes/fireware/12/en-US/EN_ReleaseNotes_Fireware_12_1_3_U7/index.html#Fireware/en-US/resolved_issues.html

Рисунок 1 – Пример описания уровня опасности уязвимостей программного обеспечения, содержащего в банке данных угроз безопасности информации ФСТЭК России (bdu.fstec.ru).

Показатель I_{cvss} определяется на основании описания уровня опасности уязвимостей программного обеспечения и входящих в его состав базовых, временных и контекстных метрик путем их расчета применительно в конкретной информационной (автоматизированной) системе и (или) информационной инфраструктуре.

Базовые метрики отражают основные характеристики уязвимостей, влияющих на доступность, целостность и конфиденциальность информации, которые не изменяются во времени и не зависят от среды функционирования программного обеспечения и включают вектор атаки, сложность атаки, уровень привилегий, взаимодействие с пользователем, влияние на конфиденциальность, целостность и доступность

Временные метрики отражают характеристики уязвимости, которые изменяются со временем, но не зависят от среды функционирования программного обеспечения и включают доступность средств эксплуатации, доступность средств устранения, степень доверия к информации об уязвимостях.

Контекстные метрики отражают характеристики уязвимости, зависящие от среды функционирования программного обеспечения.

Расчет показателя I_{cvss} может быть осуществлен вручную, а также с использованием калькулятора, содержащегося в разделе «Уязвимости» банка данных угроз безопасности информации ФСТЭК России (bdu.fstec.ru). Для того, чтобы рассчитать показатель I_{cvss} применительно к уязвимости в конкретной информационной (автоматизированной) системе и (или) информационной инфраструктуре с использованием указанного калькулятора необходимо перейти по ссылки через описание базового вектора уязвимости (рисунок 2).

Рисунок 2 – Переход к калькулятору для расчета уровня опасности уязвимости

После чего перед вами откроется калькулятор, в котором необходимо заполнить и (или) уточнить базовые, временные и контекстные метрики применительно к конкретной информационной (автоматизированной) системе и (или) информационной инфраструктуре (рисунок 3, 4, 5).

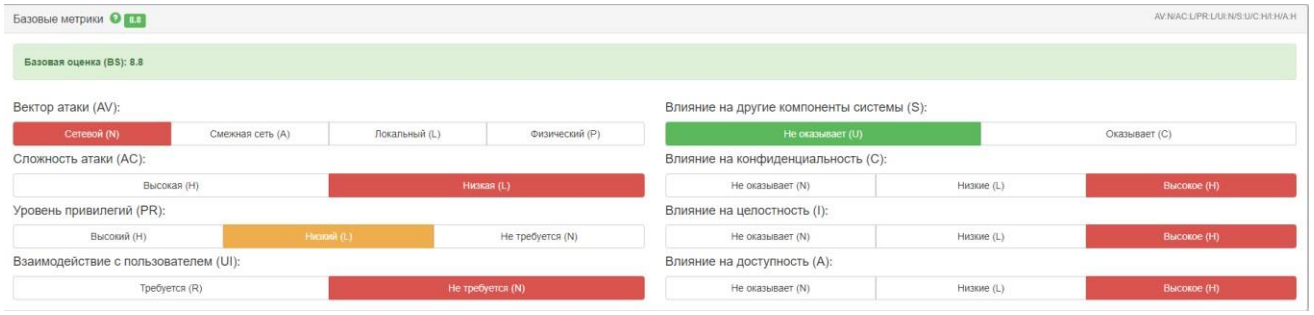


Рисунок 3 – Расчет базовых метрик уязвимости

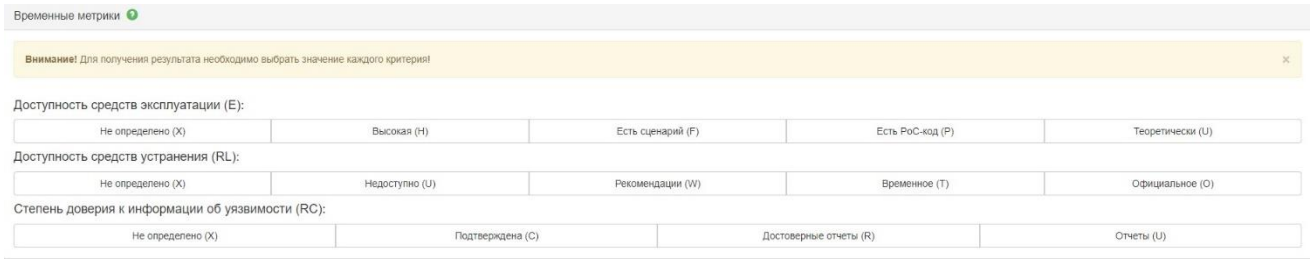


Рисунок 4 – Расчет временных метрик уязвимости

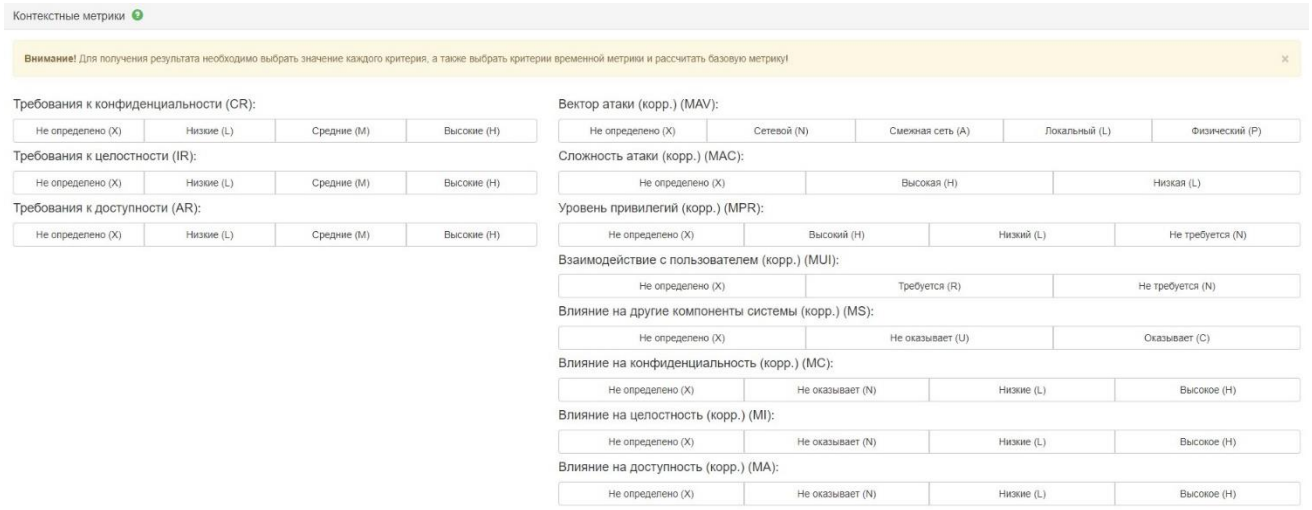


Рисунок 5 - Расчет контекстных метрик уязвимости

Расчет уровня опасности уязвимости применительно к конкретной информационной системе и (или) информационной инфраструктуре при задании оператором различным метрик в калькуляторе рассчитывается автоматически и отображается в поле «Контекстные метрики» (рисунок 6).

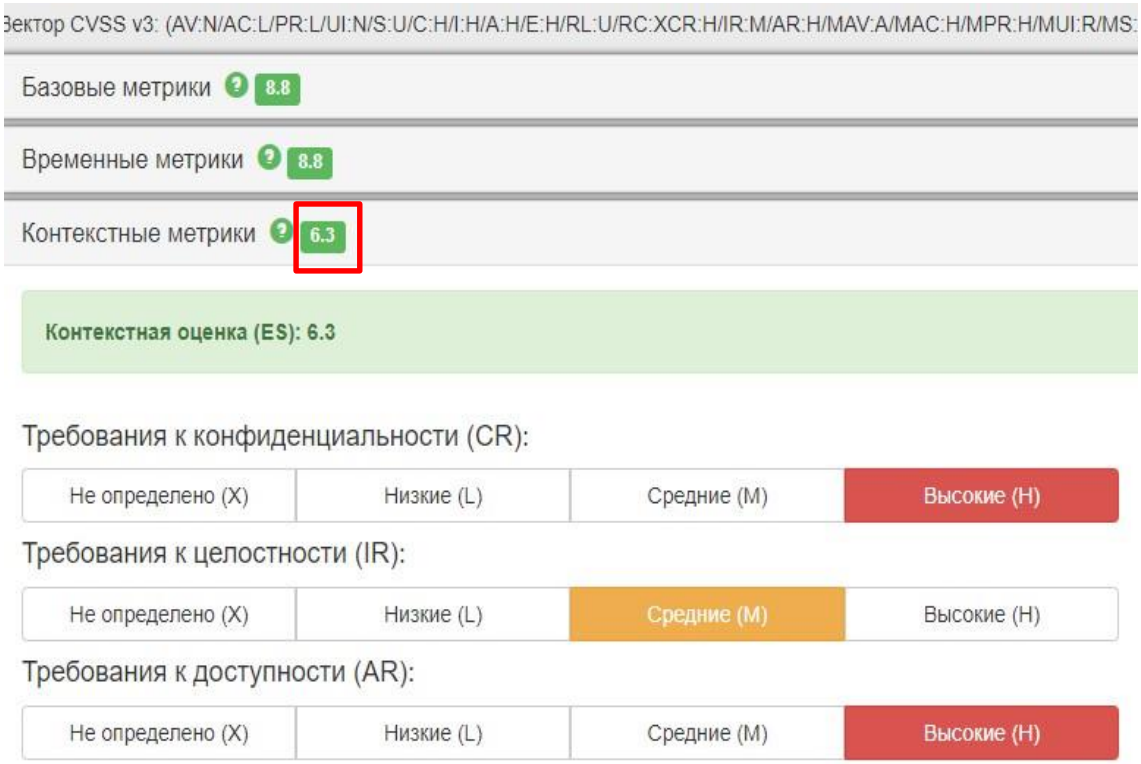


Рисунок 6 – Значение пересчитанного оператором информационной (автоматизированной) системы уровня опасности уязвимости

Рассчитанный указанным образом уровень опасности уязвимости программного обеспечения определяет показатель I_{cvss}.

7. Показатель, определяющий влияние уязвимости программного обеспечения на функционирование информационной (автоматизированной) системы и (или) информационной инфраструктуры, I_{infr} *определяется* по следующей формуле:

$$I_{infr} = k * K + l * L + p * P, \text{ где}$$

K – показатель, определяющий тип компонента информационной (автоматизированной) системы и (или) информационной инфраструктуры, подверженного уязвимости;

L – показатель, определяющий количество уязвимых компонентов информационной (автоматизированной) системы и (или) информационной инфраструктуры (автоматизированных рабочих мест, серверов, телекоммуникационного оборудования, средств защиты информации и других компонентов системы);

P – показатель, определяющий влияние уязвимого компонента на периметр информационной (автоматизированной) системы и (или)

информационной инфраструктуры; *k, l, p* – веса соответствующих показателей.

Расчет весов и оценок показателей, определяющих влияние уязвимости программного обеспечения на информационную (автоматизированную) систему и (или) информационной инфраструктуры, проводится в соответствии с таблицей 1.

Таблица 1

№ п/п	**Показатель**	**Вес**	**Значение**	**Оценка**	**Итог** *(k*Ki, l*Lj, p*Pm)*
1	Тип компонента информационной (автоматизированной) системы и (или) информационной инфраструктуры, подверженного уязвимости (К)	0,4	Уязвимости подвержены компоненты системы, обеспечивающие реализацию критических процессов (бизнес-процессов)	1	0,4
			Уязвимости подвержены сервера	0,8	0,32
			Телекоммуникационное оборудование и сеть передачи данных	0,8	0,32
			Автоматизированные рабочие места	0,5	0,20
			Другие компоненты	0,5	0,20
2	Количество уязвимых компонентов информационной (автоматизированной) системы и (или) информационной инфраструктуры (автоматизированных рабочих мест, серверов, телекоммуникационного оборудования, средств защиты	0,2	Более 70% компонентов от общего числа компонентов в системе	1	0,2
			50-70% компонентов от общего числа компонентов в системе	0,8	0,16
			10-50% компонентов от общего числа компонентов в системе	0,6	0,12
			Менее 10% компонентов от общего числа компонентов в системе	0,5	0,10

	информации и другие компоненты) (L)				
3	Влияние на периметр информационной (автоматизированной) системы и (или) информационной инфраструктуры (Р)	0,4	Уязвимое программное обеспечение доступно из сети (находится на периметре системы)	1	0,4
			Уязвимое программное обеспечение недоступно из сети находится не на периметре системы)	0,5	0,2

7. По результатам расчета уровень критичности уязвимости применительно к конкретной информационной (автоматизированной) системе и (или) информационной инфраструктуре может принимать значения, указанные в таблице 2.

Таблица 2

№ п/п	**Суммарное количество баллов уязвимости**	**Оценка уровня критичности уязвимости**
1	$7,0 \leq V \leq 10,0$	Критичный
2	$4,5 \leq V < 7,0$	Высокий
3	$1,5 \leq V < 4,5$	Средний
4	$V < 1,5$	Низкий

В зависимости от уровня критичности уязвимости программного обеспечения в конкретной информационной системы и (или) информационной инфраструктуре оператором принимается решение о необходимости устранения уязвимости.

II. Рекомендации по проведению тестирования (исследований) обновлений программного обеспечения с открытым исходным кодом на предмет наличия недекларированного функционала

1. Выделить в обновлениях программного обеспечения все потоки данных и все внутренние состояния, которые зависят от недоверенных данных (параметры командной строки, параметры HTTP-запросов, данные, получаемые по сокетам или читаемые из внешних файлов или хранилищ и другие данные) или на которые возможно влиять через такие данные. При невозможности корректного выделения в область исследования попадают все потоки данных в сегменте информационной инфраструктуры, где применяется обновляемое программное обеспечение.

2. Убедиться, что ко всем перечисленным потокам данных применяется валидация[1] при их передаче в измененные функции и санитизация[2] при их возврате из измененных функций. Если не применяются, тогда нужно отслеживать эти потоки глубже, по неизмененному коду, чтобы убедиться, что валидация и санитизация не применяются к измененному коду.

3. Проверить наличие в коде обновления программного обеспечения ключевой информации (паролей, приватных ключей, крипто-токенов и другой информации). Подобной информации в программном коде не должно быть.

4. Если в измененном коде присутствуют функции ввода-вывода, необходимо определить, когда, при каких условиях и какие из этих функций будут выполняться, а также – каков характер передаваемой с их помощью информации. Если они используются для передачи чувствительной информации по каналам передачи данных, убедиться, в безопасности канала передачи информации и актуальности используемых алгоритмов шифрования, электронной подписью или другими способами в зависимости от требований к конфиденциальности и целостности этой информации.

---

[1] Валидация — проверка данных на соответствие каким-либо критериям; возможна валидация двух видов: синтаксическая (например, проверка на соответствие регулярному выражению) и семантическая (например, проверка числа на вхождение в определенный диапазон).

[2] Санитизация — преобразование входных строковых данных в вид, безопасный для их использования в качестве выходных (примерами являются HtmlEncode, UrlEncode, addslashes и другие выходные данные).

5. Убедиться, что в случае назначения прав доступа пользователям в измененном программном коде, всегда соблюдается принцип явного присвоения наименьших привилегий для всех используемых ролей.

Убедиться, что в коде обновления не реализован механизм автоматического получения обновлений без прямого утверждения администратором информационной инфраструктуры.

6. Все зависимости, появившиеся в проекте с изменениями, подлежат хотя бы версионной проверке (например, с помощью решений типа Software Composition Analysis) и сверки их аутентичности с оригинальными версиями (при возможности).

7. Убедиться в отсутствии обфусцированного и скрытого кода (неотображаемые символы Unicode). Например, некоторые среды разработки подсвечивают скрытые символы Unicode, а также такие символы также можно обнаружить, если просматривать исходные коды через HEX-редактор.