Приложение 3

Рекомендации по повышению защищенности информационной инфраструктуры в части устранения типовых организационных и технических недостатков

1. Несоответствие заявленных в техническом паспорте объекта информатизации программных и программно-аппаратных средств

1.1. Обеспечить соответствие заявленным в техническом паспорте на информационную (автоматизированную) систему (или иной документации) программного обеспечения, программно-аппаратных средств и средств защиты информации.

1.2. Обеспечить применение средств защиты информации с актуальными сертификатами соответствия требованиям по безопасности информации.

1.3. Применять сертифицированные ФСТЭК России средства защиты информации (в том числе сертифицированные средства резервного копирования, системы управления базами данных, системы виртуализации).

2. Недостаточная строгость парольной политики

2.1. Реализовать политику управления паролями такую, что длина пароля должна быть не менее 15 символов, пароль должен содержать буквы верхнего и нижнего регистра (А-Я, A-Z, а-я, a-z), специальные символы (!, », №, %, *, /), в пароле не должно быть персонифицированной информации (имен, адресов, даты рождения, телефонов). Запретить использование комбинаций соседних клавиш и словарных слов. Пароли от учетных записей на разных сервисах должны отличаться. Не использовать пароли «по умолчанию».

2.2. Утвердить парольную политику органа (организации) и обеспечить ее реализацию.

2.3. Проинструктировать работников органа (организации) о правилах установленной парольной политики, а также о необходимости ее соблюдения.

3. Уязвимое и устаревшее программное обеспечение

3.1. Обеспечить обновление устаревших компонент, библиотек, а также обновление программного обеспечения, разработчиками которого являются отечественные производители, в соответствии с рекомендациями разработчиков, размещенных на их официальных сайтах (порталах) с целью установки актуальных обновлений безопасности.

3.2. Обеспечить обновление устаревших компонент, библиотек, а также обновление программного обеспечения, разработчиками которого являются иностранные производители, в соответствии с Методикой тестирования обновлений безопасности программных, программно-аппаратных средств, утвержденной ФСТЭК России 28.10.2022, с целью установки актуальных обновлений безопасности и минимизации риска использования известных уязвимостей.

4. Недостатки управления доступом

4.1. Произвести инвентаризацию прав доступа учетных записей (в том числе привилегированных и служебных) и ограничить их в соответствии с принципом наименьших привилегий.

4.2. Исключить доступ пользователям к компонентам информационной инфраструктуры с использованием анонимных (гостевых) учетных записей.

4.3. Обеспечить отключение (удаление) неактивных учетных записей пользователей.

4.4. Обеспечить централизованное управление средствами антивирусной защиты, а также обновление их баз данных.

5. Наличие уязвимой конфигурации центров сертификации Active Directory Certificate Services (AD CS)

В случае применения в информационной инфраструктуре органа (организации) серверной инфраструктуры на базе операционной системы Windows необходимо:

5.1. Обеспечить резервирование выданных сертификатов и проверку наличия их актуальных резервных копий.

5.2. Осуществлять регистрацию событий из журнала безопасности Windows Event ID, включая:

4886 (Удостоверяющий центр получил запрос на выдачу сертификата);

4887 (Удостоверяющий центр одобрил запрос на выдачу сертификата);

4768, 4769, 4771,4776 (аутентификация Kerberos и NTLM).

5.3. Отключить в оснастке шаблонов сертификатов (certtmpl.msc) для Subject Name флага «Supply in the request», ограничить «ClientAuthentication» в шаблонах.

5.4. Ограничить возможность NTLM-аутентификации на серверах через настройку параметров ветки реестра HKEY_LOCAL_MACHINE\SYSTEM\ CurrentControlSet\Control\Lsa\MSV1_0.

На всех серверах, кроме контроллера домена, установить:

RestrictReceivingNTLMTraffic = 2;

RestrictSendingNTLMTraffic = 2.

На контроллере домена установить:

RestrictReceivingNTLMTraffic = 1;

RestrictSendingNTLMTraffic = 1.

При необходимости разрешить доступ с определенных узлов, используя список исключений ServerAllowedToReceiveNTLMClients = <*список*>.

При использовании в информационной инфраструктуре устаревших версий клиентов, программного обеспечения без поддержки аутентификации по протоколу Kerberos или при отсутствии корректной настройки имени субъекта-службы (SPN) перед ограничением NTLM-аутентификации необходимо провести аудит информационной безопасности информационной инфраструктуры и оценить зависимость систем и сервисов от указанного способа аутентификации.

Для всех служб Active Directory Certificate Services (CertSrv, CEP, CES и других) включить шифрование и защиту канала. В Microsoft IIS включить параметр Extended Protection = Required.

5.5. Включить на сервере центра сертификации флаг IF_ENFORCEENCRYPTICERTREQUEST, который отвечает за требование шифрования запросов на выдачу сертификатов. Для этого необходимо в командной строке сервера выполнить команду:

certutil -setreg CA\InterfaceFlags +IF_ENFORCEENCRYPTICERTREQUEST.

После изменения параметра службу Active Directory Certificate Services необходимо перезагрузить командой:

net stop certsvc && net start certsvc.

6. Недостаточная защита веб-сайтов от сетевых атак

6.1. Нейтрализация угроз безопасности информации, связанных с реализацией XSS-атак.

Использовать HTTP-заголовок Content-Security-Policy для защиты содержимого запроса, указав в качестве источников только доверенные поддомены, а также ограничив внедрение программного кода (скриптов).

Настроить корректную обработку данных, вводимых пользователем, следующим образом:

не встраивать напрямую в страницу данные, полученные из недоверенных источников;

заменять потенциально небезопасные символы в синтаксисе HTML (например, «<», «>», «&») на их эквиваленты, которые не являются символами форматирования;

проверять данные, вводимые пользователем, как на стороне клиента, так и на стороне сервера.

Не использовать потенциально опасные JavaScript-функции и DOM-объекты, которыми может воспользоваться злоумышленник для проведения атак.

Для cookie-файлов необходимо установить специальные флаги «Secure» и «HttpOnly».

6.2. Нейтрализация угроз безопасности информации, связанных с внедрением внешних сущностей XML (XXE).

Исключить использование при обработке XML-данных внешних сущностей (External Entity), внешних параметров сущностей (External Parameter Entity) и внешних описаний типа документа (External Doctype). Обновить до последних актуальных версий обработчики XML-документов и библиотеки для работы с XML-документами, используемые веб-сервером.

6.3. Нейтрализация угроз безопасности информации, связанных с внедрением HTML-инъекций.

Обеспечить валидацию данных, экранирование пользовательского ввода (потенциально небезопасные символы, которые могут быть использованы при форматировании HTML-страницы (например, «<», «>»,

«&»), заменить на их эквиваленты, не являющиеся символами форматирования).

6.4. Нейтрализация угроз безопасности информации, связанных с внедрением SQL-инъекции.

Использовать параметризованные запросы при вводе данных пользователем. Например, в веб-приложениях, реализованных на языке PHP, для реализации данной меры защиты информации может быть использован модуль PHP Data Objects (PDO).

В случаях, когда возможные значения пользовательского ввода известны заранее, рекомендуется использовать механизмы «белого списка» для проверки входных данных.

Экранировать входные данные через фильтрацию ввода в веб-формах и URL-запросах.

Минимизировать права доступа пользователей сайтов к базам данных. Пользователи сайта должны получать доступ к базе данных после прохождения аутентификации и с минимальными необходимыми привилегиями.

6.5. Нейтрализация угроз безопасности информации, связанных с реализацией уязвимости типа «локальное включение файлов» (Local File Inclusion (LFI)).

Обеспечить корректную обработку контролируемых пользователями данных, исключать использование данных, содержащих специальные и другие непредусмотренные символы (например, «../»), запретить использование абсолютных путей.

6.6. Нейтрализация угроз безопасности информации, связанных с удаленным выполнением кода (RCE).

Отключить использование опасных функций (например, eval, exec, system) на веб-сервере. Обеспечить строгую валидацию входных данных, настроив «белые списки» разрешенных форматов, а также настроив фильтрацию потенциально вредоносных символов и кода.

6.7. Нейтрализация угроз безопасности информации, связанных с наличием небезопасных прямых ссылок на объекты (IDOR).

Внедрить контроль доступа и управление сеансами пользователей перед доступом к защищаемой информации. Реализовать проверку прав доступа к объекту по contract_id, использовать UUID вместо предсказуемых ID.

7. Раскрытие конфиденциальной информации. Доступ к API веб-сервиса

7.1. Ограничить доступ к API-запросам на уровне сети. Настроить на веб-сервере обработку ошибок для предотвращения раскрытия внутренней информации. Ограничить информацию, возвращаемую в ответах на запросы, в том числе при выводе ошибок.

7.2. Внедрить механизмы проверки прав доступа к объектам, следуя принципу наименьших привилегий – запретить доступ по умолчанию для всех

ресурсов, за исключением общедоступных, и предоставлять доступ к ресурсам только их владельцам и уполномоченным пользователям.

7.3. Запретить прямой доступ к служебным файлам. Поместить подключаемые файлы в одну директорию и запретить к ней прямой доступ. В файл .htaccess добавить строку deny from all.

7.4. Отключить вывод ошибок веб-приложения пользователю. Например, в веб-приложениях, реализованных на языке PHP, путем добавления в файл .htaccess следующей строки:

php_flag display_errors offphp_value error_reporting 0.

7.5. Реализовать «белый список» разрешенных доменов для параметра return_url. Реализовать проверку целевого URL-запроса на стороне сервера: разрешать только заранее определённые внешние домены. Ограничить список допустимых доменов или URL-запросов для перенаправления (переадресации).

8. Наличие конфиденциальных данных в исходном коде

Исключить хранение конфиденциальных (аутентификационных) данных, которые жестко закодированы в исходном коде. Хранить аутентификационные данные в зашифрованных файлах конфигурации или базах данных, либо в переменных среды.

9. Небезопасная конфигурация CORS

В случае применения совместного использования ресурсов из разных источников (CORS) в информационной инфраструктуре необходимо настроить «белый список» доменов, с которых возможно обращение к целевому домену. Проверять вхождение значения заголовка Origin в этот список. Не использовать значение заголовка «*».

10. Недостатки защиты от атак подбора учетных данных

Для защиты от атак подбора учетных данных необходимо унифицировать ответы сервера при вводе неправильных учетных данных с целью исключения получения списка имен пользователей. Внедрить механизмы ограничения количества попыток ввода учетных данных (например, CAPTCHA) для предотвращения массовых автоматизированных запросов. Внедрить аудит многократных или аномальных попыток ввода учетных данных для последующего анализа и реагирования на инциденты.

11. Публично доступные служебные порты

11.1. Выявить перечень сервисов (служб), функционирующих в информационной инфраструктуре органа (организации), которые в перечне публичных (внешних) IP-адресов доступны из сети «Интернет». Определить легитимность доступных по открытым портам сервисов, IP-адресов, сетевых служб, доступ к которым возможен за периметром информационной инфраструктуры органа (организации). В случае невозможности отнесения их к легитимным осуществить их блокировку.

11.2. Исключить удаленное администрирование сетевых (пограничных) устройств, в том числе публикацию интерфейсов удаленного управления

(SSH, RDP, VNC) на сетевом периметре или в демилитаризованной зоне сетевой инфраструктуры органа (организации).

11.3. Ограничить доступ к сетевым портам, по которым доступен Java Debug Wire Protocol, разрешив обращаться к ним только с определенных IP-адресов.

12. Использование небезопасных протоколов аутентификации

Отключить устаревшие протоколы и механизмы аутентификации, включая NTLMv1, Net-NTLMv1, LM Hash, Digest Authentication, Wdigest, Basic Authentication, Password Authentication Protocol, Challenge Handshake Auth Protocol, MS-CHAPv1/v2.

13. Небезопасная настройка JWT-токенов доступа

При использовании для аутентификации клиента в веб-сервисе JWT-токенов необходимо использовать алгоритмы шифрования с хранением секретного ключа вне кода приложения (например, RS256, ES256). Включить обязательную проверку подписи JWT-токенов для всех конечных точек. Внедрить механизм отзыва токенов (blacklist). Не передавать JWT-токен в URL-запросах — использовать только secure HTTP-only cookies. Использовать разные секреты для JWT-токенов в тестовой и рабочей средах, а также разные секреты JWT для разных сервисов.

14. Некорректная настройка регистрации событий безопасности

Настроить мониторинг событий информационной безопасности в соответствии с разделами 4 и 5 ГОСТ Р 59547-2021, обратив особое внимание на обеспечение мониторинга событий информационной безопасности при предоставлении удаленного доступа к информационной инфраструктуре.