
# Ключевые уязвимости информационных систем российских компаний

март 2022 – март 2023

rt-solar.ru
rt.ru

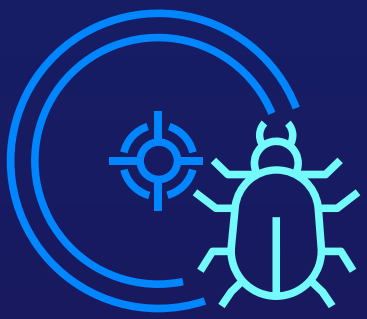
Ростелеком
Солар

# Оглавление

# Об отчете

Настоящий отчет содержит результаты аналитических исследований, основанных на статистических данных проектов по анализу защищенности и тестированию на проникновение, проведенных экспертами отдела анализа защищенности центра противодействия кибератакам Solar JSOC компании «Ростелеком-Солар» с марта 2022 года по март 2023 года.

В отчете приведены сведения о распространенных уязвимостях, угрозах и векторах проникновения в корпоративные сети. Представленная аналитика базируется на результатах порядка 80 проектов. За услугами по анализу защищенности и тестированию на проникновение обращались компании из различных городов России. Отраслевая принадлежность исследованных организаций также разнообразна: госсектор, телекоммуникации, информационные технологии, маркетинг, энергетика, торговля и т. д.



# Методология

При исследовании обнаруженных уязвимостей использовались:

- результаты проектов по анализу защищенности веб- и мобильных приложений;
- итоги работ по внешнему и внутреннему тестированию на проникновение.

В одном проекте каждый тип уязвимости учитывался только один раз. Иначе говоря, из нескольких одинаковых уязвимостей в статистику попала только одна – с наибольшей критичностью и наименьшей сложностью эксплуатации. Здесь основными критериями оценки уязвимости были:

- возможность и последствия эксплуатации;
- местонахождение;
- роль уязвимой функциональности или системы;
- необходимость особых условий для эксплуатации.

Такой выбор критериев обусловлен тем, что разработчикам свойственно допускать однотипные ошибки в различных местах. При составлении статистики учитывалось количество проектов с определенным типом уязвимости, а не общее количество уязвимостей конкретного типа.

В проектах по тестированию на проникновение анализировались успешно реализованные векторы атак, которые позволили достигнуть поставленных целей (получить контроль над доменом, доступ во внутреннюю сеть извне и т.п.). Длина вектора отражает количество атак или действий, которые были совершены.

# Ключевые выводы

Преодолеть **внешний периметр** удалось в **65%** исследованных компаний. Традиционно наиболее уязвимые системы внешнего периметра – это веб-приложения.

Преодолеть **внутренний периметр** удалось **во всех** проектах. Известные уязвимости и небезопасная конфигурация используемых сервисов – наиболее распространенные проблемы внутренней инфраструктуры.

Самые частые **недостатки веб-приложений** (встретились в **86%** проектов) связаны с некорректной реализацией контроля доступа. При этом в трети приложений уязвимость имеет высокую степень критичности.

Самая распространенная уязвимость клиентской части **мобильных приложений** – небезопасное хранение данных на устройстве. А серверной части – недостатки контроля доступа. Указанные уязвимости встретились в **64%** и **82%** исследованных приложений соответственно.

# Внешнее тестирование на проникновение

Внешнее тестирование на проникновение направлено на поиск уязвимостей и недостатков с высоким уровнем критичности, эксплуатация которых может привести к получению доступа во внутреннюю сеть организации или к критическим внешним системам. При проведении работ моделируются действия потенциального внешнего нарушителя, не обладающего данными об инфраструктуре. Подобный подход позволяет получить независимую оценку эффективности методов и средств защиты информации в компании.

## Результаты работ

В **65%** проектов специалисты «Ростелеком-Солар» успешно преодолели внешний периметр. Еще в **24%** компаний удалось получить доступ к различным внешним системам и приложениям.

**Результаты внешнего пентеста**

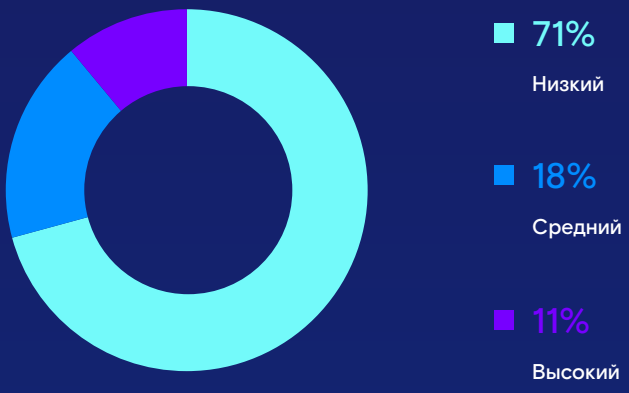
- 29% — Получен доступ во внутреннюю сеть
- 36% — Доступ в DMZ/компрометация отдельных узлов
- 24% — Доступ к критичным внешним системам и приложениям
- 11% — Не получен доступ к критичным системам и узлам

Фатальные последствия доступа хакеров во внутреннюю сеть компании очевидны. Однако и проникновение в сеть DMZ (Demilitarized Zone – сегмент, где расположены сетевые устройства, взаимодействующие с внешними сетями) не менее опасно, ведь в дальнейшем оттуда злоумышленник также может проникнуть во внутренний периметр. Как видно из диаграммы выше, доступ в DMZ был успешно реализован более чем в трети проектов.

Компрометация отделенных от внутренней сети серверов и приложений – также серьезная угроза. Контроль над ними позволяет злоумышленникам вносить изменения в конфигурацию (например, создавать новые страницы для размещения нелегитимного контента, блокировать доступ к функциональности или удалять основное содержимое и данные). Такие атаки значительно снижают доверие пользователей к ресурсам, ударяют по репутации компании и нарушают работу устоявшихся бизнес-процессов.

Уровень защищенности внешнего периметра был оценен как низкий в 71% исследованных компаний. Это означает, что в них злоумышленник легко может проникнуть во внутреннюю сеть. Стоит отметить, что в представленном исследовании не учитываются векторы проникновения, связанные с социотехническими атаками (социальная инженерия) и беспроводными сетями (то есть взломами Wi-Fi-сетей).

**Уровни защищенности исследованных внешних периметров**




## Векторы преодоления внешнего периметра

В исследовании учитывались только векторы, реализованные до получения первоначального доступа во внутреннюю сеть и сеть DMZ без дальнейшего развития и продвижения во внутренней сети.

Минимальный вектор состоял из 1 шага, то есть реальному злоумышленнику достаточно было бы выполнить одно действие для компрометации узла сети. Яркий пример – эксплуатация известных уязвимостей, например CVE-2022-27228 в популярной CMS-системе Bitrix. При этом уже выпущено обновление ПО, которое могло бы закрыть вектор проникновения, но оно установлено далеко не во всех организациях.

**9**

**Максимальное число векторов проникновения в одном проекте**

**21**

**Максимальное число скомпрометированных узлов в одном проекте**

Максимальное количество векторов проникновения чаще всего встречается в больших инфраструктурах, где на внешнем периметре расположено множество различных ресурсов (сетевые устройства, корпоративные порталы, приложения для клиентов, тестовые приложения и т.д.). В малых же инфраструктурах обнаруживались 1–2 вектора проникновения.

Отметим, что при анализе не учитывались одинаковые векторы, приведшие к компрометации разных серверов. Различие в количестве векторов и скомпрометированных узлов показывает, что одинаковые уязвимости могут находиться на разных узлах внешнего периметра, и при их устранении чаще всего требуется комплексный подход по защите всего периметра.

Отдельно обратим внимание на вектор, который включал в себя проведение атаки на пользователя, а именно **«Межсайтовое внедрение сценариев (XSS)»**. Успешные векторы проникновения обычно включают атаки на сервер, однако получить доступ можно и через пользователя. В одном из проектов был отправлен отзыв с нагрузкой для компрометации сессии читающего отзыв пользователя. Этим пользователем оказался администратор, который обладал доступом к функциональности загрузки PHP-сценариев. Получив сессию администратора, мы смогли загрузить веб-шелл (т.е. вредоносный скрипт) через недоступную ранее функциональность. Таким образом, успешное проведение XSS стало начальной точкой для проникновения во внутреннюю сеть.

## Уязвимости

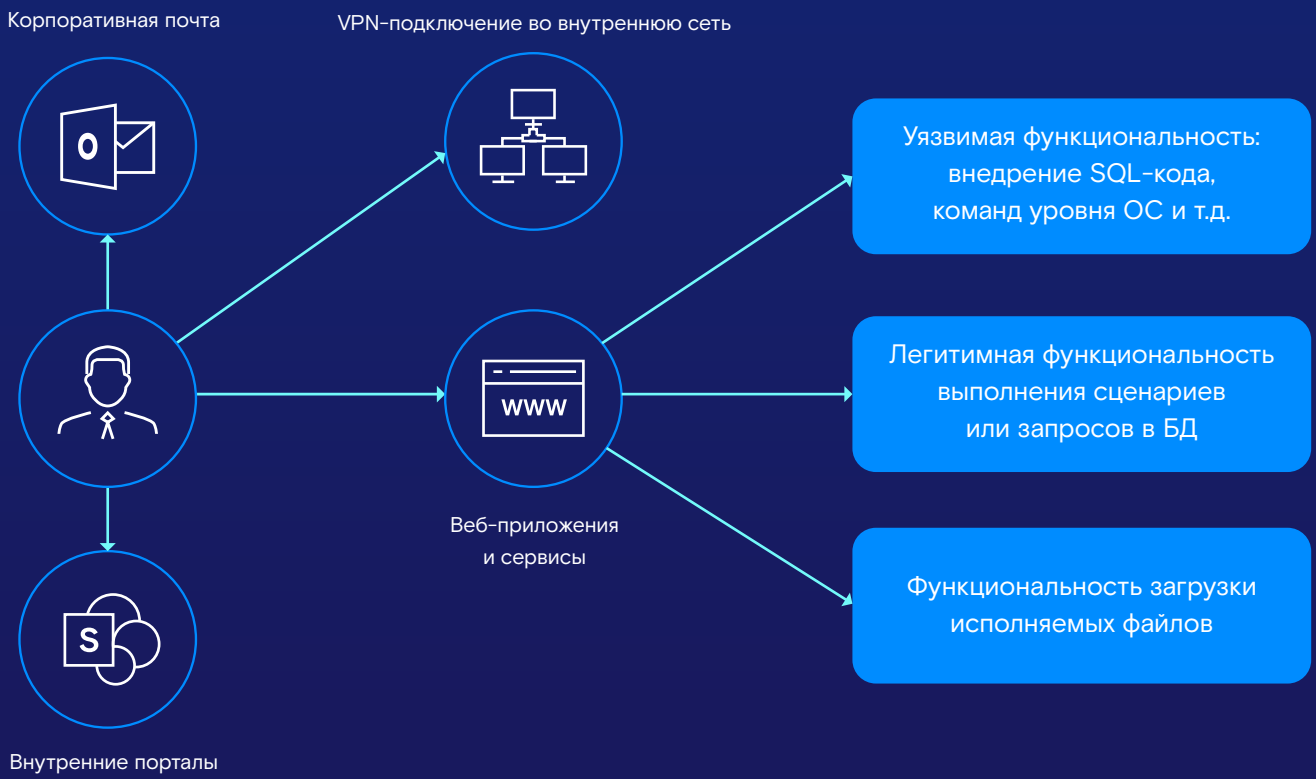
Внешнее тестирование на проникновение направлено на поиск уязвимостей с высоким уровнем критичности, поскольку именно они чаще всего позволяют получить доступ к ресурсам компаний. На внешних периметрах компаний наиболее часто встречались следующие критические уязвимости:

- использование слабых паролей для учетных записей;
- внедрение SQL-кода в запросы к базе данных;
- недостатки контроля доступа;
- использование программного обеспечения с известными уязвимостями;
- выход за пределы назначенного каталога (Directory traversal).

## Распространенные критические уязвимости внешних периметров


А наиболее распространенной начальной точкой проникновения оказалась атака подбора учетных данных приложений и сервисов. Подобный шаг был выполнен в 41% реализованных векторов. При этом были подобраны как словарные учетные данные test:test и admin:admin, так и слабые пароли для ранее обнаруженных учетных записей пользователей. Скомпрометированные учетные данные пользователей применялись далее для установки удаленного подключения к VPN, реализации административной функциональности и эксплуатации других уязвимостей.

### Возможности использования скомпрометированных паролей при внешнем тестировании на проникновение:



# Внутреннее тестирование на проникновение

Внутреннее тестирование на проникновение направлено на проверку возможности повышения привилегий во внутренней инфраструктуре, получения доступа к критически важной информации или системам.

Мы проводили как отдельные работы по внутреннему тестированию на проникновение, так и в качестве продолжения внешнего пентеста. В первом случае работы проводились с предоставленного заказчиком доступа, во втором – с доступа, полученного в результате преодоления внешнего периметра.

## Результаты работ

Перед проведением работ заказчиками ставились как общие цели (получение контроля над доменом), так и актуальные для определенной компании (выгрузка данных из системы X). Во всех выполненных проектах поставленные цели были достигнуты.

В итоге **93%** исследованных инфраструктур были отмечены низким уровнем защищенности. Высокий уровень при этом не был обнаружен ни в одной компании.

Подобные результаты демонстрируют недостаточную защищенность корпоративных сетей. Кроме того, по результатам комплексных проектов (где проводился и внутренний, и внешний пентест), можно сделать вывод, что компании уделяют больше внимания защите внешнего периметра, чем внутренних ресурсов.

## Векторы повышения привилегий

Работы по внутреннему тестированию на проникновение состояли из 2 основных этапов:

- получение необходимых привилегий в домене;
- использование полученных привилегий для выполнения определенных действий согласно поставленной цели.

Поскольку общим этапом работ во внутренней сети являлось получение определенных привилегий в домене, рассмотрим именно эти векторы атак. Под получением привилегий не всегда имеется в виду уровень администратора домена. Для выполнения некоторых задач достаточно было, например, привилегий локального администратора на рабочих станциях или членства в определенной группе.

**Минимальный реализованный вектор состоял из**

**2 шагов**

Одним из примеров такого вектора является эксплуатация уязвимости MS17-010, известной еще с 2017 года. Используя такую старую уязвимость вместе с извлеченными ранее учетными данными администратора домена из хранилища LSA (Local Security Authority), в одном из проектов нам удалось получить полный контроль над доменной инфраструктурой. Этот вектор был отмечен низким уровнем сложности, так как для его реализации достаточно использовать общеизвестное автоматизированное ПО.

В целом большинство векторов в минувшем году включали именно эксплуатацию известных уязвимостей. Среди них можно выделить:

- CVE-2022-26923
- CVE-2021-42278
- CVE-2021-42287

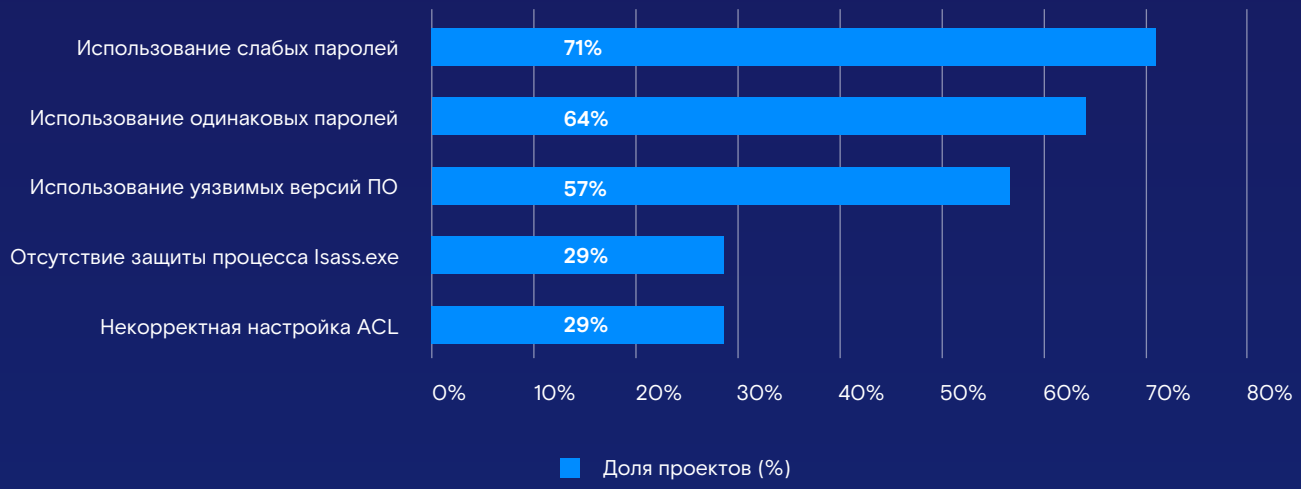
Успешная эксплуатация уязвимостей CVE-2021-42278 и CVE-2021-42287 позволяет компьютеру выдать себя за контроллер домена и запросить сервисный билет с более высокими привилегиями. Уязвимость CVE-2022-26923 связана с центром сертификации и позволяет получить сертификат для контроллера домена, который затем может быть использован для проведения других атак, например, DCSync.

**Примеры атак с использованием известных уязвимостей:**


Другим популярным вектором оказалась эксплуатация небезопасной настройки центра сертификации и шаблонов сертификатов. Используемые конфигурации позволяли выпустить непривилегированному пользователю сертификаты администраторов, а затем использовать их для полной компрометации учетных записей.

## Уязвимости

Наиболее частые уязвимости во внутренних сетях:




Одной из наиболее распространенных уязвимостей уже не первый год остается использование слабых паролей. Эта проблема встретилась нам в **71%** проектов. В компаниях используют пароли по умолчанию, словарные и простые пароли (например, Qwerty123). Также было обнаружено использование паролей, совпадающих с именем учетной записи. Даже если такой пароль соответствует строгой парольной политике, он все еще остается легко подбираемым.

С использованием слабых и повторяющихся паролей связана актуальность таких атак, как Kerberoasting[1] и Password Spraying[2]. В одном из проектов проведение атаки Password Spraying привело к компрометации более 180 доменных учетных записей. При этом на всех был установлен один и тот же словарный пароль. Вероятно, этот пароль использовался по умолчанию и его просто не сменили.

---

[1] Kerberoasting – атака на реализацию Kerberos в домене Active Directory. Позволяет получить сервисные билеты для последующего перебора паролей для учетных записей, от имени которых запущены сервисы.

[2] Password Spraying – атака, направленная на подбор учетных данных. Особенность атаки заключается в выполнении одной попытки ввода пароля для множества учетных записей, что предотвращает возможные блокировки пользователей.

# Анализ защищенности веб-приложений

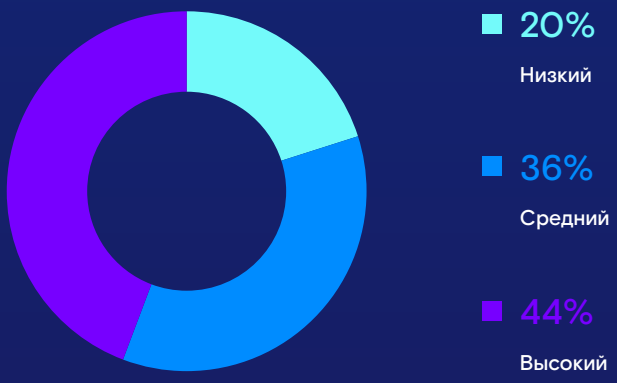
Подобные работы направлены на поиск максимального числа уязвимостей, демонстрацию возможностей их эксплуатации и оценку общего уровня защищенности приложения. Практика показывает, что веб-приложения все еще остаются наиболее уязвимыми системами на внешнем периметре, поэтому проведение их анализа так же актуально, как и внешний пентест.

## Результаты работ

По итогам работ по анализу защищенности оценивается общий уровень защищенности приложения. При его оценке учитываются все найденные уязвимости, сложность их эксплуатации и возможные последствия для приложения и его пользователей.

Низким уровнем защищенности было отмечено **20%** исследованных веб-приложений. То есть каждый пятый ресурс содержал критические уязвимости, приводящие к исполнению произвольного кода, повышению привилегий, доступу к конфиденциальной информации и прочим негативным для организации последствиям. При этом хотя бы одна уязвимость с высоким уровнем критичности была обнаружена в **77%** исследованных приложений.

**Уровни защищенности веб-приложений**



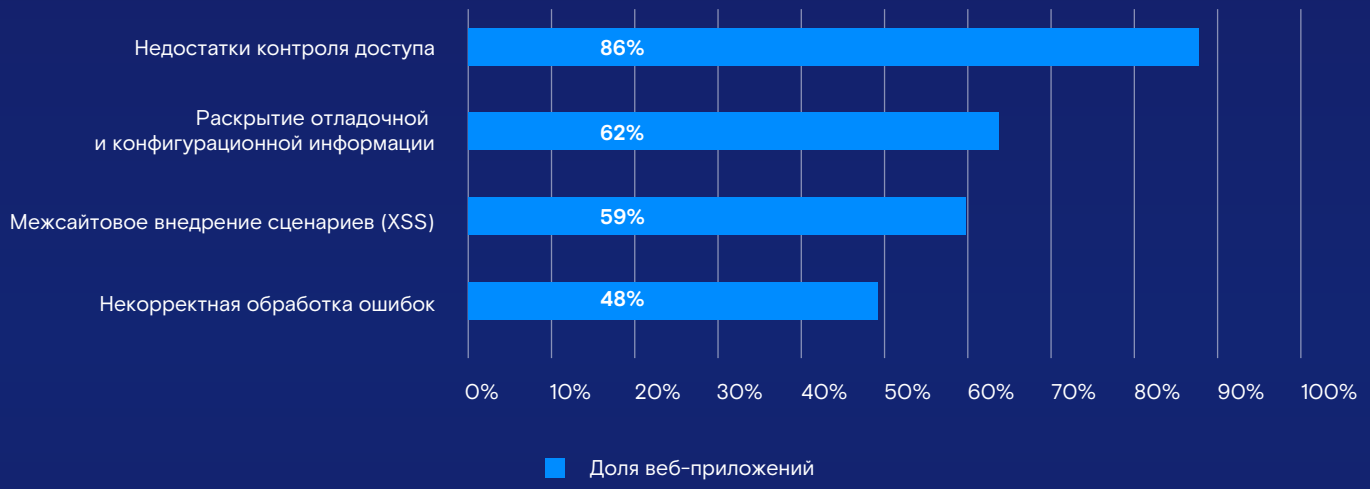

## Уязвимости

Недостатки контроля доступа были обнаружены в большей части исследованных веб-приложений **(86%)** и остаются самой распространенной проблемой для них. При этом в **34%** проектов уязвимость имела высокий уровень критичности. Самым распространенным примером некорректной настройки доступа являются небезопасные прямые ссылки на объекты, которые были выявлены в **55%** всех исследованных приложений.

В **62%** проектов мы обнаружили сценарии, раскрывающие дополнительные сведения о структуре, компонентах или работе приложения. При этом в **14%** приложений подобные уязвимости были отмечены высоким уровнем критичности. Примерами здесь являются: раскрытие учетных данных пользователей в журналах приложений, данных для подключения к внутренним сервисам, ключей доступа или прочей критичной информации.

Среди наиболее распространенных также присутствуют уязвимости, позволяющие проводить атаку **«Межсайтовое внедрение сценариев (XSS)»**. Она направлена на пользователей приложений и может привести к компрометации данных или сессий пользователей, выполнению действий от их имени и другим серьезным последствиям.

**Уязвимости веб-приложений**




Кроме того, среди распространенных уязвимостей с высокой степенью критичности можно выделить возможность проведения атаки «Внедрение SQL-кода в запросы к базе данных». Это очень старая уязвимость, которая в 80% случаев позволяет полностью сломать приложение и попасть во внутреннюю сеть. И хотя сегодня она встречается реже, в **14%** исследованных приложений нам удалось обнаружить эту уязвимость.

Помимо критичности, оценивалась и сложность эксплуатации обнаруженных уязвимостей. Этот критерий определяется необходимостью конкретных условий или действий, которые не зависят от атакующего (определенное состояние приложения, выполнение действия пользователем и т. п.). **80%** всех обнаруженных уязвимостей были отмечены низким уровнем сложности. То есть для их эксплуатации не нужны какие-либо специальные условия или особенные технологии.

## Сложность эксплуатации уязвимостей веб-приложений

20%
Высокая

80%
Низкая

Также оказалось, что для эксплуатации **56%** уязвимостей не требуется авторизованный доступ к приложению, а значит,  уязвимая функциональность доступна любому внешнему пользователю, даже если у него нет каких-либо привилегий в приложении.

## Условия эксплуатации уязвимостей веб-приложений

44%
Авторизованный пользователь

56%
Неавторизованный пользователь

# Анализ защищенности мобильных приложений

Анализ защищенности мобильных приложений направлен на поиск максимального числа уязвимостей, демонстрацию возможностей их эксплуатации и оценку общего уровня защищенности. В ходе каждого проекта проводится проверка приложений для двух операционных систем: iOS и Android. Исследованные приложения организаций предназначены для взаимодействия с клиентами и партнерами.

## Уязвимости

Поиск уязвимостей осуществляется как в клиентской, так и в серверной части приложений.

**Самые распространенные уязвимости в серверной части:**

**82%** **Недостатки контроля доступа**

Контроль доступа ограничивает действия пользователей за пределами установленных для них привилегий и защищает данные от несанкционированного доступа. Наиболее частый недостаток этого класса – небезопасные прямые ссылки на объекты. Он связан с возможностью получения доступа к каким-либо объектам приложения при прямом обращении к ним. Следовательно, злоумышленник может получить несанкционированный доступ к информации (ее чтение, изменение или удаление).

**64%** **Недостатки бизнес-логики**

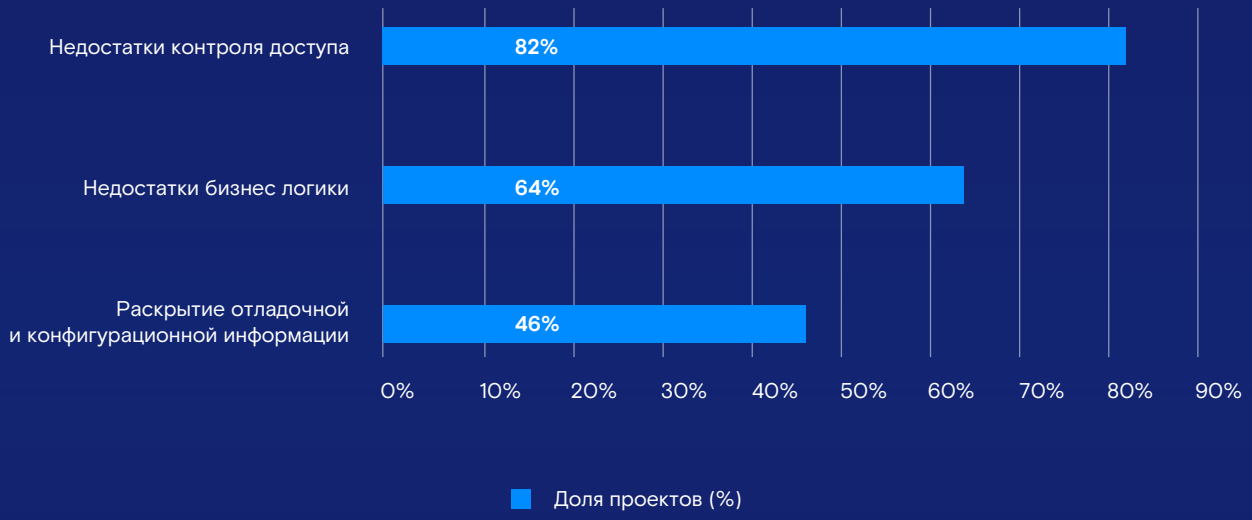
Они связаны с недостаточной проработкой сценариев использования приложения. Чаще всего недостатки возникают из-за того, что при разработке приложения не учитывается возможность прямого взаимодействия пользователя с серверной частью. Примерами таких недостатков являются использование отсутствующей в интерфейсе функциональности и обход ограничений путем подмены параметров.

## 46% Раскрытие отладочной и конфигурационной информации

Избыточные сообщения об ошибках, журналы, отладочные интерфейсы – все это предоставляет дополнительные данные о приложении и может использоваться для его исследования и поиска уязвимостей.

Указанные уязвимости являются распространенными не только для интерфейсов мобильных приложений, но и для программных интерфейсов в целом.

**Распространенные уязвимости серверных частей мобильных приложений**




**Самые распространенные уязвимости клиентской части:**

## 64% Небезопасное хранение данных на устройстве

Почти две трети исследованных приложений сохраняют чувствительные данные пользователя в журналах или других файлах непосредственно на устройствах.

В частности, при проведении работ мы обнаружили сессии пользователей, их личные данные (ФИО, номер телефона, адрес электронной почты), истории запросов и прочее. При этом после выхода пользователя из приложения и аккаунта данные с устройств не удалялись.

## 55% Недостатки работы с сессиями/токенами пользователя

Здесь можно выделить некорректное завершение сессии пользователей. Так, при выходе из аккаунта сессия удалялась на стороне клиента, но запрос на ее удаление на сервер не отправлялся. В результате после выхода пользователя из аккаунта его сессия оставалась активной в течение времени ее жизни (она удалялась на сервере не сразу, а только спустя несколько часов).

## 27% Раскрытие информации в исходном коде клиентской части приложения

В исходных кодах клиентской части приложений были обнаружены учетные записи разработчиков, токены авторизации и прочие данные, которые не были удалены перед выводом приложения в продуктивное использование.

Несмотря на то, что исходный код не направляется пользователям в открытом виде, содержащаяся в нем информация становится доступна после распаковки и декомпиляции приложения.

**Распростаненные уязвимости клиентских частей мобильных приложений**

# Рекомендации

В очередной раз считаем необходимым подчеркнуть, что своевременное обнаружение и закрытие уязвимостей позволяет не только обезопасить инфраструктуру компании, но и защитить клиентов и пользователей от действий потенциальных злоумышленников.

**Для повышения общего уровня кибербезопасности рекомендуется проводить:**

- анализ защищенности приложений перед их выводом в продуктивное использование. Это позволит избежать появления критических уязвимостей на внешнем или внутреннем периметре в связи с внедрением новых решений. А компаниям-разработчикам проверка приложений по окончании разработки позволит заранее устранять уязвимости и предоставлять своим клиентам только безопасные продукты;

- регулярные внешние и внутренние тестирования на проникновение. При этом внутреннее тестирование на проникновение не менее важно, чем внешнее. Стоит помнить, что цель внешнего пентеста – это выявление способов преодоления внешнего периметра, а внутреннего – возможностей развития атаки внутри сети. Только комплексный подход способен обеспечить высокий уровень защищенности от внешних и внутренних злоумышленников;

- регулярное автоматизированное сканирование, которое позволяет с минимальными затратами выявить известные уязвимости и недостатки систем (в том числе из базы CVE), отслеживать актуальность используемых компонентов и появление в них новых уязвимостей. Эти задачи решает сервис контроля уязвимостей (Vulnerability Management, VM).