?????????? ?????????? ??????????

Классификация по типам проверки подлинности:

- Знание (пароль или ответ на вопрос безопасности, одноразовый код). Факторы знаний.
- Обладание (физический объект - мобильный телефон или маркер безопасности - магнитная карта). Фактор владения.
- Биометрия (персональные данные или модели поведения - конфиденциальная информация). Фактор согласованности.

Механизмы аутентификации уязвимы в случаях:

- Не могут адекватно защитить от атак с применением методов грубой силы.
- Логические ошибки или плохой код в реализации позволяют полностью обойти аутентификацию.

Уязвимости Password-based аутентификации

Пользователи регистрируются для получения учетной записи, или администратор присваивает им учетную запись. Учетная запись связана с уникальным именем пользователя и паролем.

Безопасность сайта будет скомпрометирована, если получить или угадать учетные данные другого пользователя.

Уязвимости в этом механизме возникают по различным причинам, одни из них:

- Возможные brute-force атаки
- Уязвимая защита от brute-force атак
- HTTP-basic аутентификация

Возможные brute-force атаки

Brute-forcing пользовательских имен

Работает если соответствуют узнаваемому шаблону, например, адресу электронной почты. Очень часто логины бывают в формате firstname.lastname@somecompany.com

Иногда высокопривилегированные учетные записи создаются с использованием предсказуемых имен пользователей, таких как admin или administrator.

Brute-forcing паролей

Часто бывают настроены политики паролей с высокой энтропией, которые сложнее взломать одним перебором. Возможные условия:

- Минимального количества символов
- Смеси строчных и заглавных букв
- Как минимум одного специального символа

Но пользователи часто берут пароль, который они могут запомнить, и пытаются использовать его в соответствие с политикой паролей. Например, если не разрешено использование password, пользователи могут попробовать что-нибудь вроде P@ssw0rd или P4$$w0rd!.

Предположение предсказуемых закономерностей означает, что атаки с применением грубой силы часто могут быть гораздо более изощренными и более эффективными, чем простые итерации через всевозможные комбинации символов.

Уязвимая защита от brute-force атак
Для успешной компроментации аккаунта необходимо провести множество неудачных попыток. Защита строится на замедлении скорости перебора пароля. Наиболее распространенные способы предотвращения атак:

- Блокировка учетной записи в случае большого количества неудачных попыток входа.
- Блокировка IP-адреса удаленного пользователя в случае большого количества попыток входа в систему.

Иногда счетчик количества неудачных попыток сбрасывается при успешном входе владельца IP-адреса. Это означает, что злоумышленнику просто придется входить в систему под своей учетной записью каждые несколько попыток, чтобы этот лимит никогда не был достигнут.

В этом случае достаточно просто включать свои собственные учетные данные для входа в систему через регулярные интервалы времени в течение перебора всего словаря, чтобы сделать эту защиту практически бесполезной.

Basic-аутентификация HTTP

Несмотря на старость метода, ее часто используют. Клиент получает маркер аутентификации, который строится путем сцепления имени пользователя и пароля, а также их кодировки в Base64. Этот токен хранится в браузере, который автоматически добавляет его в заголовок авторизации каждого последующего запроса следующим образом:

Authorization: Basic base64(username:password)

Причины уязвимости:

- Многократная отправка учетных данных при каждом запросе. Если на веб-сайте также не реализована HSTS, могут быть перехвачены через Man-in-the-Middle.
- Часто не поддерживает защиту от переборов грубой силы. Поскольку токен состоит из статических значений.
- Уязвима к атакам, связанным с сеансом, в частности к CSRF

В некоторых случаях использование уязвимой базовой HTTP-аутентификации может дать злоумышленнику доступ только к, казалось бы, неинтересной странице. Однако, в дополнение к обеспечению дополнительной поверхности атаки, учетные данные, раскрытые таким образом, могут быть повторно использованы в других, более конфиденциальных контекстах.

Уязвимости мультифакторной аутентификации

Проверка биометрических факторов является непрактичной для большинства веб сайтов. Чаще встречается двухфакторная аутентификация (2FA). Для многофакторной аутентификации необходимы различные факторы. Проверка одного фактора разными способами не является двухфакторной аутентификацией.

Одним из таких примеров является 2FA через электронную почту. Хотя пользователь должен предоставить пароль и проверочный код, доступ к коду предполагается на основе того, что пользователь знает учетные данные для входа в свою учетную запись электронной почты. Поэтому фактор проверки подлинности знания просто проверяется дважды.

Обход двухфакторной аутентификации:

- Несовершенная реализация
- Иногда ошибочная логика двухфакторной аутентификации означает, что после того, как пользователь выполнил начальный шаг входа в систему, веб-сайт не может адекватно проверить, что этот же пользователь выполняет второй шаг.
- Отсутствие мер по предотвращению перебора проверочного кода 2FA.

Уязвимости механизмов смены пароля, восстановления пароля, поддержания сессии
Большинство сайтов предоставляют дополнительные функциональные возможности управления учетной записью. Например, изменение и сброс пароля. Эти механизмы также могут добавлять уязвимости.

Разработчики стараются избежать известных уязвимостей на страницах входа. Однако забывают об аналогичные шагах на связанной функциональности. Это особенно важно при возможности пользователя самому создавать учетную запись и имеет доступ к изучению этих дополнительных страниц.

Сторонние механизмы аутентификации, которые также могут быть уязвимы:

- Механизм поддержания сессии «запомнить меня»
- Механизм сброса пароля через E-mail
- Механизм сброса пароля с использованием URL и одноразового токена

- Механизм смены пароля

Дополнительные ссылки

- Серия нерегулярного подкаста с обсуждением основных атак на аутентификацию и угон аккаунтов
- OWASP-памятка по реализации механизмов аутентификации

Тренироваться на упражнениях можно на открытых платформах. Одна из лучших платформ по изучению проблем безопасности веб-приложений: PortSwigger Academy.

- Попрактикуемся в атаках на аутентификацию

Типичные ошибки в BurpSuite

Так же настоятельно рекомендуем ознакомится с различными инструментами, используемыми для брутфорсинга (hydra, medusa, patator и др.):

- https://www.kali.org/tools/hydra/
- https://www.kali.org/tools/medusa/
- https://www.kali.org/tools/patator/

Пример атаки

Атака производилась на стенд stepik Тестовые стенды Задача в поиске строки в формате 32 букв и цифр из кода страницы панели администратора.

Сначала получим список возможных страниц.

```
ffuf -u http://192.168.1.199:1337/FUZZ/ -w fuzz.txt
```

Нашли страницу админки. Включаем Burp и находим способ отправки логина и пароля. Сохраняем запрос в payload.txt

Копируем словарь паролей https://github.com/empty-jack/YAWR раздел brute -> passwords -> reallyBest.txt Я переименовал его в passlist.txt

Теперь попробуем clusterbomb. Я столкнулся с интересной проблемой перебора через ffuf. Непонятно, какой пароль подошел. Потом, после размышлений, догнал. Однако это очень-простое-приложение) Burp все-таки удобнее.

```
ffuf -request payload.txt -request-proto http -mode clusterbomb -w loginlist.txt:HFUZZ -w passlist.txt:WFUZZ -mc 200
```

А не понял из-за ограничения на статус 200. Не знал, что после корректной авторизации приложение переадресовывает эту сессию на следующую страницу. В реальных задачах потребуется проверка на блокировку, ...

Теперь подбираем одноразовый код аналогично. Надоело ждать перебор через Burp, сделал простой скрипт для формирования списка из 999 чисел и отдал его в ffuf

---

Revision #6
Created 27 September 2025 04:32:04 by Admin
Updated 27 September 2025 17:33:22 by Admin