1. Перечислите три вида http-аутентификации, какие из них описаны в RFC?

**BASIC, Digest, (**Forms**), TSL**

1. Опишите схему BASIC-аутентификации.

При использовании данного вида аутентификации имя пользователя и пароль включаются в состав веб-запроса ([HTTP POST](https://ru.wikipedia.org/wiki/POST_(HTTP)) или [[GET](https://ru.wikipedia.org/wiki/HTTP#GET%7CHTTP)]). Любой перехвативший пакет легко узнает секретную информацию. Даже если контент с ограниченным доступом не слишком важен, этот метод лучше не использовать, так как пользователь может применять один и тот же пароль на нескольких [веб-сайтах](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B0%D0%B9%D1%82). Также из недостатков парольной аутентификации следует отметить невысокий уровень безопасности — пароль можно подсмотреть, угадать, подобрать, сообщить посторонним лицам и т. д.

1. Опишите схему DIGEST-аутентификации.

Это аутентификация, при которой пароль пользователя передается в [хешированном](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D1%88-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F) виде. пароль хешируется всегда с добавлением произвольной строки символов, которая генерируется на каждое соединение заново. Таким образом при каждом соединении генерируется новый хеш пароля и перехват его ничего не даст. Дайджест-аутентификация поддерживается всеми популярными серверами и [браузерами](https://ru.wikipedia.org/wiki/%D0%91%D1%80%D0%B0%D1%83%D0%B7%D0%B5%D1%80).

1. Опишите схему FORMS-аутентификации.

Аутентификация по [cookies](https://ru.wikipedia.org/wiki/Cookies) токену. Если cookie удастся похитить, то, подделав его, можно аутентифицироваться в качестве другого пользователя. В случае, когда вводимые данные плохо фильтруются или не фильтруются вовсе, похитить cookies становится не очень сложным предприятием[[3]](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D0%B2_%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D0%B5#cite_note-3). Чтобы как-то улучшить ситуацию используется защита по [IP-адресу](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81), то есть cookies [сессии](https://ru.wikipedia.org/wiki/%D0%A1%D0%B5%D1%81%D1%81%D0%B8%D1%8F_(%D0%B2%D0%B5%D0%B1-%D0%B0%D0%BD%D0%B0%D0%BB%D0%B8%D1%82%D0%B8%D0%BA%D0%B0)) связываются с IP-адресом, с которого пользователь изначально авторизовался в системе. Однако IP-адрес можно подделать используя [IP-спуфинг](https://ru.wikipedia.org/wiki/IP-%D1%81%D0%BF%D1%83%D1%84%D0%B8%D0%BD%D0%B3), поэтому надеяться на защиту по [IP-адресу](https://ru.wikipedia.org/wiki/IP-%D0%B0%D0%B4%D1%80%D0%B5%D1%81) тоже нельзя. На данный момент большинство браузеров[[4]](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D0%B2_%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D0%B5#cite_note-4) используют куки с флагом HTTPonly[[5]](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F_%D0%B2_%D0%98%D0%BD%D1%82%D0%B5%D1%80%D0%BD%D0%B5%D1%82%D0%B5#cite_note-microsoft533046-5), который запрещает доступ к cookies различным скриптам.

1. Почему в чистом виде http-аутентификация не является надежной?

передача пароля в открытом виде