

Авторизация Яндекса

□ вводим логин test@yandex.ru, браузер шлет аjax-запрос:
https://passport.yandex.ru/registration-validations/auth/multi_step/start
csrf_token:
login: test@yandex.ru
process_uuid:
retpath: https://mail.yandex.ru/
origin: home_desktop_ru
□ вводим пароль 123, браузер шлет аjax-запрос:
https://passport.yandex.ru/registration-validations/auth/multi_step/commit_password
csrf_token:
track_id:
password: 123
retpath: https://mail.yandex.ru/

Авторизация на mail.ru

□ вводим логин test@mail, браузер шлет аjax-запрос:
https://auth.mail.ru/api/v1/pushauth/info?login=test@mail.ru
login: test@mail.ru
□ вводим пароль testme, браузер шлет аjax-запрос:
login: test@mail.ru
password: testme
saveauth: 0
token:
project: e.mail.ru

Авторизация на гос.услугах

□ вводим логин в виде снилса, вводим пароль testme
https://esia.gosuslugi.ru/idp/login/pwd/do
login: xxx-xxx-xxx xx
password: tesme
command: ...
idType: snils

Авторизация в сбербанк-онлайн

□ вводился логин test и пароль pass
https://online.sberbank.ru/CSAFront/authMainJson.do
deviceprint:
jsEvents:
domElements:
operation: button.begin
parameters:
{ "login": "test", "password": "pass", "pageInputType": "INDEX", "loginInputType": "BY_LOGIN", "storeLogin": false }

Авторизация ВКонтакте

□ вводим ящик test@mail.ru и пароль 1234
https://login.vk.com/?act=login
act: login
role: al_frame
expire:

to:
recaptcha:
captcha_sid:
captcha_key:
_origin: https://vk.com
ip_h:
lg_h:
ul:
email: test@mail.ru
pass: **1234**

Авторизация распространенного в регионах «Сетевой Город. Образования».

□ Вводим логин **Родитель**, пароль **secret**.

Вначале идет запрос на <https://sgo.edu-74.ru/webapi/auth/getdata>, где нам присылают соль:

lt: "684976415"

salt: "20209483060"

ver: "712829123"

Далее идет запрос на адрес <https://sgo.edu-74.ru/webapi/login> с параметрами:

LoginType: 1

cid: 2

sid: 1

pid: -1

cn: 1

sft: 2

scid: 3331

UN: **Родитель**

PW: **bb39c7**

lt: 684976415

pw2: **bb39c7f7277f17754f6ddb8b74d4c20b**

ver: 712829123

Реверс-инжиниринг выявил, что поле pw2 = MD5(salt, MD5(pass)). Похоже на схему 4, только здесь параметр challenge называется salt. Достоверно судить о том, в какой форме хранятся пароли в БД, я не решусь. Может быть в хешированной (но без соли), а может быть и в открытой.

Авторизация Гугл.

□ ВВОДИМ ЛОГИН testlogin@gmail.com

https://accounts.google.com/_lookup/accountlookup?hl=ru&_reqid=39516&rt=j

continue: <https://myaccount.google.com/?...>

service: accountsettings

flowName: GlifWebSignIn

flowEntry: ServiceLogin

f.req: ["testlogin@gmail.com", "здесь какой-то треш", ...]

bgRequest: ["identifier", "еще один треш"]

azt: **возможно nonce + текущее время**

cookiesDisabled: false

deviceinfo:

gmscoreversion: undefined

checkConnection:

checkedDomains: youtube

pstMsg: 1

□ вводим пароль 123456:

https://accounts.google.com/_signin/challenge?hl=ru&TL=здесь_треш&_reqid=239516&rt=j

...

f.req: ["здесь вообще рандомная печалька",null,1,null,[1,null,null,null,["123456",null,true]]]

azt: возможно nonce + текущее время

Вначале, я подумал, что гугл явно использует продвинутый вариант парольной авторизации. Но потом разглядел-таки в потоке треша передаваемых параметров заветный пароль. Мда. Похоже это современная практика.