



Web

Что такое и зачем?



Web

Что такое и зачем?



Что мы рассмотрим?

- Как работает сайт.
 - Как происходит подключение к сайту?
 - Что такое порт?
 - Какие есть протоколы и зачем нам прокси?
 - Как создать сайт и что нам искать?



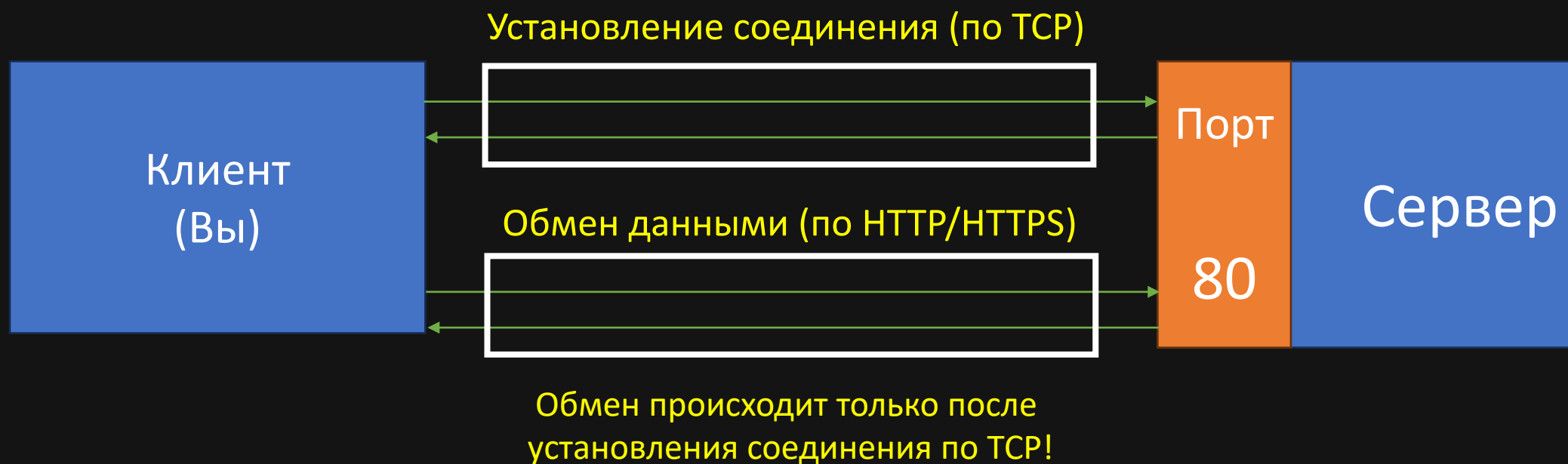
Зачем это знать?

Ответ дня

Хачу взломать гексагон (зачем взламывать пентагон, ведь $5 < 6$)



Как происходит подключение к сайту?



* Порт может быть любым, не обязательно 80 или 443

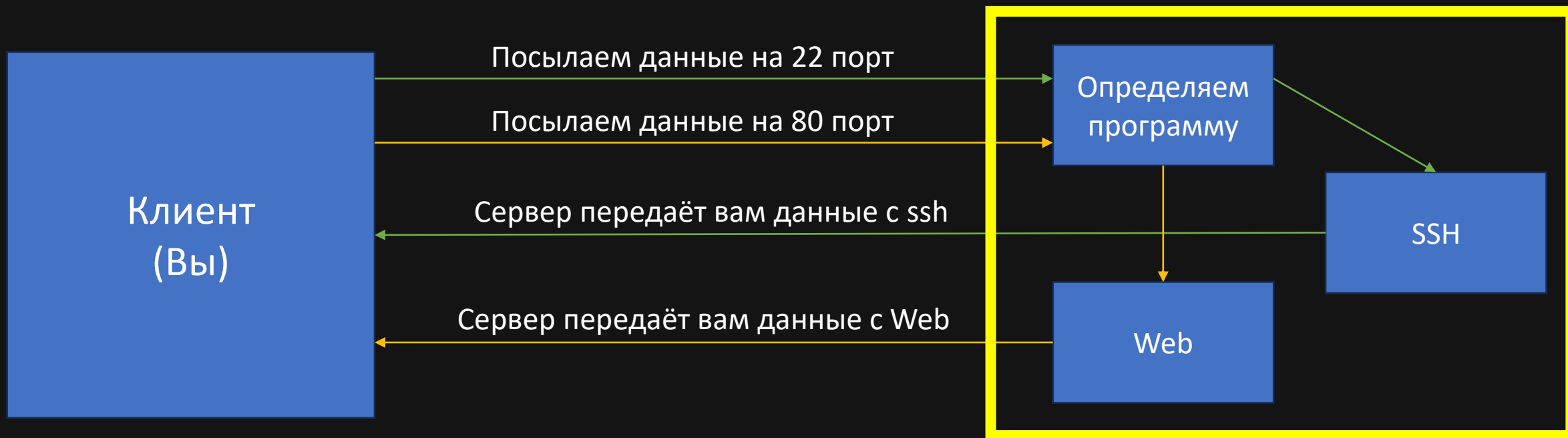


Что такое порт?

* Грубо говоря, **порт** – это номер программы, зарезервированный компьютером. На порт будут приходить данные, а компьютер в свою очередь соотносит данные с порта и программу, которой выделили порт и направляет их в программу

Пример

Сервер





Какие есть протоколы и зачем нам прокси?

Главные протоколы (на данный момент) :

TCP

HTTP

UDP

HTTPS (HTTP + SSL)

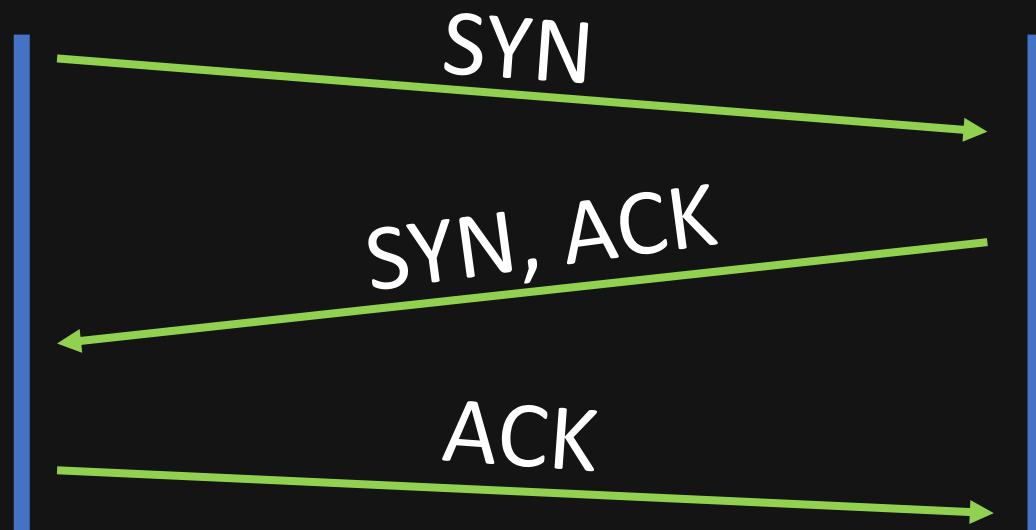


Какие есть протоколы и зачем нам прокси?

TCP

Клиент

Сервер





Какие есть протоколы и зачем нам прокси?

Как увидеть это?

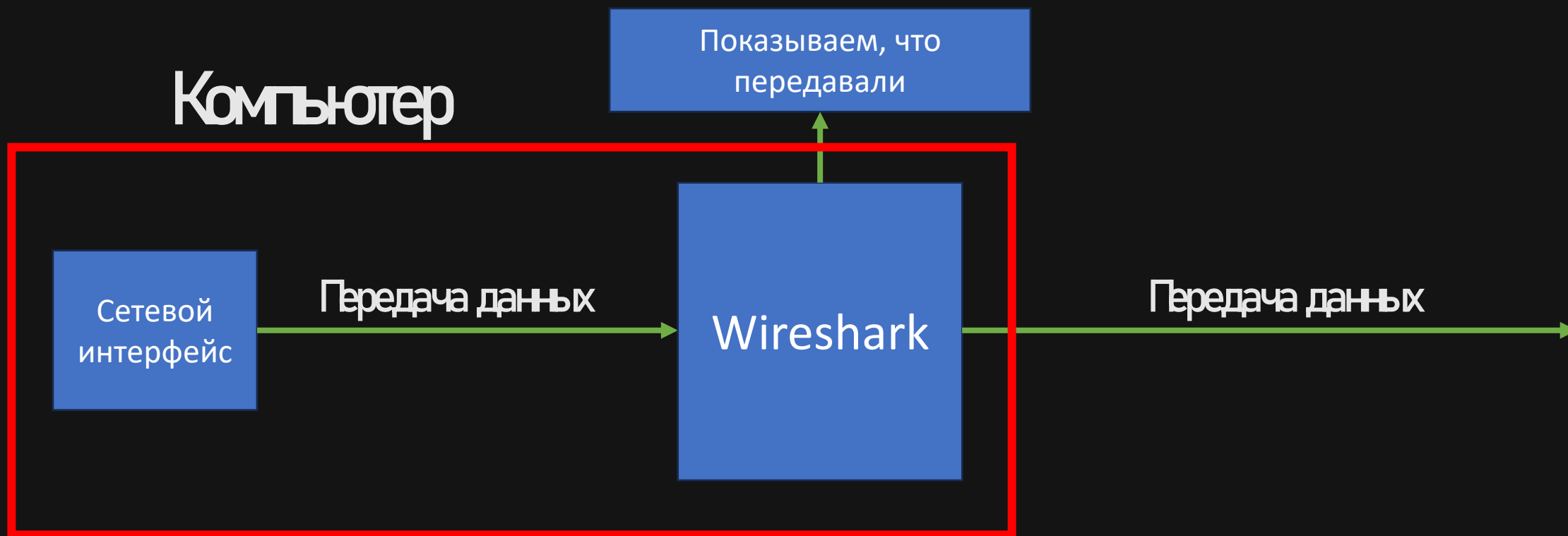
→ Через перехват трафика

* Мы используем **wireshark** для перехвата трафика



Какие есть протоколы и зачем нам прокси?

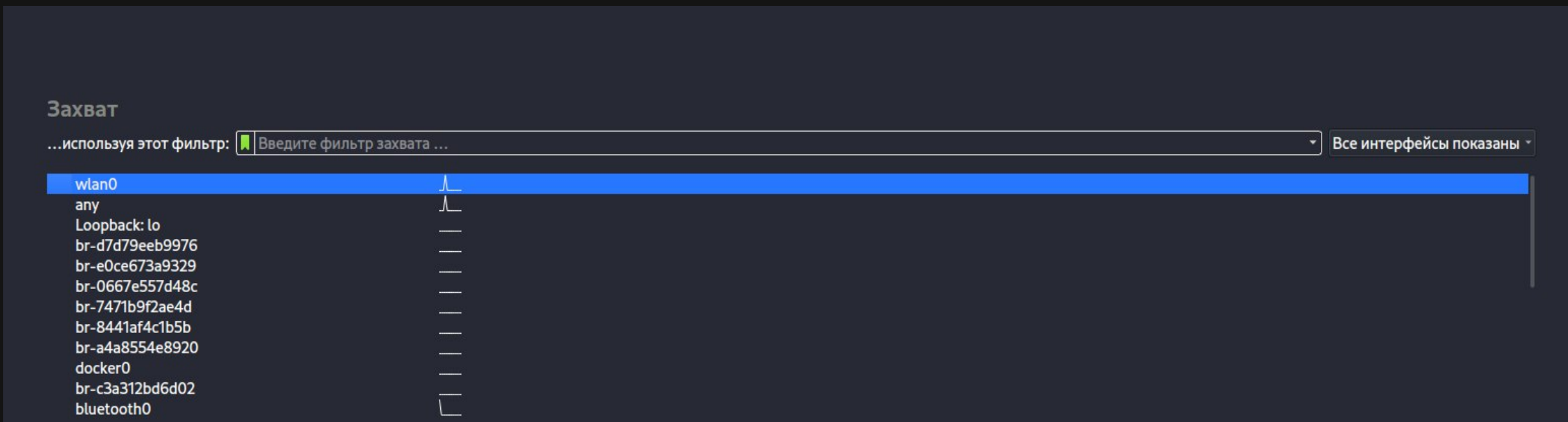
Как работает **wireshark**?





Какие есть протоколы и зачем нам прокси?

Открываем Wireshark и выбираем интерфейс (с которого будем снимать трафик) (wlan0 / eth0)





Какие есть протоколы и зачем нам прокси?

Wireshark начинает перехватывать трафик. Появляется следующее:

File Редактирование Просмотр Запуск Захват Анализ Статистика Телефония Беспроводной Инструменты Помощь

Применить дисплейный фильтр ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
315	12.391832303	192.168.43.21	149.154.167.99	TCP	66	34790 → 443 [ACK] Seq=2726 Ack=2663 Win=3940 Len=0 TSval=2697416574 TSecr=3945151371
316	12.965441216	149.154.167.99	192.168.43.21	TLSv1.2	501	Application Data
317	12.965507028	192.168.43.21	149.154.167.99	TCP	66	34790 → 443 [ACK] Seq=2726 Ack=3098 Win=3937 Len=0 TSval=2697417148 TSecr=3945151520
318	12.997556860	192.168.43.21	149.154.167.99	TLSv1.2	249	Application Data
319	13.430223454	149.154.167.99	192.168.43.21	TCP	66	443 → 34790 [ACK] Seq=3098 Ack=2909 Win=182 Len=0 TSval=3945151631 TSecr=2697417180
320	13.947580581	192.168.43.21	192.124.249.24	TCP	66	[TCP Dup ACK 28#1] 38350 → 80 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=64412822 TSecr=1455946092
321	13.947620422	192.168.43.21	192.124.249.24	TCP	66	[TCP Dup ACK 29#1] 38352 → 80 [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=64412822 TSecr=1455946092
322	14.004345185	192.168.43.21	149.154.167.99	TLSv1.2	265	Application Data
323	14.107198615	192.168.43.21	5.188.150.79	TCP	74	44800 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3851139217 TSecr=0 WS=128
324	14.132253848	192.124.249.24	192.168.43.21	TCP	66	[TCP Dup ACK 34#1] [TCP ACKed unseen segment] 80 → 38352 [ACK] Seq=1 Ack=2 Win=85 Len=0 TSval=1455956661 TSecr=64371396
325	14.132254823	192.124.249.24	192.168.43.21	TCP	66	[TCP Dup ACK 35#1] [TCP ACKed unseen segment] 80 → 38350 [ACK] Seq=1 Ack=2 Win=85 Len=0 TSval=1455956661 TSecr=64371396
326	14.132255729	149.154.167.99	192.168.43.21	TCP	66	443 → 34790 [ACK] Seq=3098 Ack=3108 Win=182 Len=0 TSval=3945151871 TSecr=2697418186
327	14.132257122	149.154.167.99	192.168.43.21	TLSv1.2	179	Application Data
328	14.132317858	192.168.43.21	149.154.167.99	TCP	66	34790 → 443 [ACK] Seq=3108 Ack=3211 Win=3940 Len=0 TSval=2697418314 TSecr=3945151871
329	14.293399213	5.188.150.79	192.168.43.21	TCP	74	8000 → 44800 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1400 SACK_PERM=1 TSval=724344050 TSecr=3851139217 WS=128

Frame 1: 199 bytes on wire (1592 bits), 199 bytes captured (1592 bits) on interface wlan0, id 0
Ethernet II, Src: Chongqin_8a:d5:01 (a8:93:4a:8a:d5:01), Dst: XiaomiCo_7b:89:5a (00:ec:0a:7b:89:5a)
Internet Protocol Version 4, Src: 192.168.43.21, Dst: 149.154.167.99
Transmission Control Protocol, Src Port: 34790, Dst Port: 443, Seq: 1, Ack: 1, Len: 133
Transport Layer Security



Какие есть протоколы и зачем нам прокси?

Wireshark перехватывает многие протоколы. Введём фильтр **TCP** (чтобы видеть передачи только по нему)

The screenshot shows the Wireshark interface with the filter bar set to 'tcp'. The packet list pane displays several packets, with packet 1 selected. The packet details pane shows the structure of the selected packet, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol.

No.	Time	Source	Destination	Protocol	Length	Info
3	0.187458457	149.154.167.99	192.168.43.21	TLSv1.2	179	Application Data
1	0.000000000	192.168.43.21	149.154.167.99	TLSv1.2	199	Application Data
498	19.706028227	149.154.167.99	192.168.43.21	TCP	66	443 → 34790 [ACK] Seq=4725 Ack=4122 Win=182 Len=0 TSval=3945153206 TSecr=2697423507
496	19.297009539	5.188.150.79	192.168.43.21	TCP	66	8000 → 44810 [ACK] Seq=484 Ack=1244 Win=64128 Len=0 TSval=724349051 TSecr=3851144203
495	19.297008075	5.188.150.79	192.168.43.21	TCP	66	8000 → 44802 [ACK] Seq=472 Ack=1026 Win=64256 Len=0 TSval=724349057 TSecr=3851144203
494	19.297007169	5.188.150.79	192.168.43.21	TCP	66	8000 → 44806 [ACK] Seq=644 Ack=1550 Win=64128 Len=0 TSval=724349050 TSecr=3851144203
493	19.297079024	192.168.43.21	149.154.167.99	TCP	66	34790 → 443 [ACK] Seq=3973 Ack=4725 Win=3939 Len=0 TSval=2697423479 TSecr=3945153115
491	19.093329987	192.168.43.21	5.188.150.79	TCP	66	44802 → 8000 [FIN, ACK] Seq=1025 Ack=472 Win=64128 Len=0 TSval=3851144203 TSecr=724348764
490	19.093295767	192.168.43.21	5.188.150.79	TCP	66	44810 → 8000 [FIN, ACK] Seq=1243 Ack=484 Win=64128 Len=0 TSval=3851144203 TSecr=724348865
489	19.093232136	192.168.43.21	5.188.150.79	TCP	66	44806 → 8000 [FIN, ACK] Seq=1549 Ack=644 Win=64128 Len=0 TSval=3851144203 TSecr=724348767
488	19.092635689	5.188.150.79	192.168.43.21	TCP	66	8000 → 44804 [ACK] Seq=238 Ack=510 Win=64768 Len=0 TSval=724348868 TSecr=3851143999
487	19.092634714	5.188.150.79	192.168.43.21	TCP	66	8000 → 44810 [FIN, ACK] Seq=483 Ack=1243 Win=64128 Len=0 TSval=724348865 TSecr=3851142115
486	19.092633250	5.188.150.79	192.168.43.21	TCP	66	8000 → 44814 [ACK] Seq=666 Ack=840 Win=64512 Len=0 TSval=724348867 TSecr=3851143999
485	19.092632274	5.188.150.79	192.168.43.21	TCP	66	8000 → 44806 [FIN, ACK] Seq=643 Ack=1549 Win=64128 Len=0 TSval=724348767 TSecr=3851142006
484	19.092630880	5.188.150.79	192.168.43.21	TCP	66	8000 → 44802 [FIN, ACK] Seq=471 Ack=1025 Win=64256 Len=0 TSval=724348764 TSecr=3851142003

Frame 1: 199 bytes on wire (1592 bits), 199 bytes captured (1592 bits) on interface wlan0, id 0
Ethernet II, Src: Chongqin_8a:d5:01 (a8:93:4a:8a:d5:01), Dst: XiaomiCo_7b:89:5a (00:ec:0a:7b:89:5a)
Internet Protocol Version 4, Src: 192.168.43.21, Dst: 149.154.167.99
Transmission Control Protocol, Src Port: 34790, Dst Port: 443, Seq: 1, Ack: 1, Len: 133
Transport Layer Security



Какие есть протоколы и зачем нам прокси?

Значение столбцов (в **wireshark**):

Source – источник (кто отправляет данные)

Destination – приёмник (кто принимает данные)

Protocol – какой протокол передачи используется

Info – краткая информация (какие флаги, порты...)



Пример: `tcp && ip.dst = 5.188.150.79` (Нашctf)

(Мы ищем все данные, переданные по TCP к ip-адресу 5.188.150.79)

No.	Time	Source	Destination	Protocol	Length	Info
491	19.093329987	192.168.43.21	5.188.150.79	TCP	66	44802 → 8000 [FIN, ACK] Seq=1025 Ack=472 Win=64128 Len=0 TSval=3851144203 TSecr=724348764
490	19.093295767	192.168.43.21	5.188.150.79	TCP	66	44810 → 8000 [FIN, ACK] Seq=1243 Ack=484 Win=64128 Len=0 TSval=3851144203 TSecr=724348865
489	19.093232136	192.168.43.21	5.188.150.79	TCP	66	44806 → 8000 [FIN, ACK] Seq=1549 Ack=644 Win=64128 Len=0 TSval=3851144203 TSecr=724348767
483	18.888676035	192.168.43.21	5.188.150.79	TCP	66	44814 → 8000 [FIN, ACK] Seq=839 Ack=666 Win=64128 Len=0 TSval=3851143999 TSecr=724348698
482	18.888594004	192.168.43.21	5.188.150.79	TCP	66	44804 → 8000 [FIN, ACK] Seq=509 Ack=238 Win=64128 Len=0 TSval=3851143999 TSecr=724348521
476	18.166908011	192.168.43.21	5.188.150.79	TCP	66	44812 → 8000 [FIN, ACK] Seq=616 Ack=237 Win=64128 Len=0 TSval=3851143277 TSecr=724348007
468	17.611256100	192.168.43.21	5.188.150.79	TCP	66	44800 → 8000 [FIN, ACK] Seq=856 Ack=24316 Win=64128 Len=0 TSval=3851142721 TSecr=724347474
466	17.004898790	192.168.43.21	5.188.150.79	TCP	66	44810 → 8000 [ACK] Seq=1243 Ack=483 Win=64128 Len=0 TSval=3851142115 TSecr=724346863
463	16.896036248	192.168.43.21	5.188.150.79	TCP	66	44806 → 8000 [ACK] Seq=1549 Ack=643 Win=64128 Len=0 TSval=3851142006 TSecr=724346767
461	16.892803412	192.168.43.21	5.188.150.79	TCP	66	44802 → 8000 [ACK] Seq=1025 Ack=471 Win=64128 Len=0 TSval=3851142003 TSecr=724346763
459	16.885253252	192.168.43.21	5.188.150.79	TCP	66	44808 → 8000 [ACK] Seq=940 Ack=24127 Win=64128 Len=0 TSval=3851141995 TSecr=724346741
457	16.885221409	192.168.43.21	5.188.150.79	TCP	66	44808 → 8000 [ACK] Seq=940 Ack=24099 Win=64128 Len=0 TSval=3851141995 TSecr=724346741
455	16.826664815	192.168.43.21	5.188.150.79	TCP	66	44814 → 8000 [ACK] Seq=839 Ack=665 Win=64128 Len=0 TSval=3851141937 TSecr=724346697
449	16.689970866	192.168.43.21	5.188.150.79	TCP	66	44804 → 8000 [ACK] Seq=509 Ack=237 Win=64128 Len=0 TSval=3851141800 TSecr=724346520
446	16.539613381	192.168.43.21	5.188.150.79	TCP	574	[TCP Retransmission] 44804 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=508 TSval=3851141650 TSecr=724345476

▶ Frame 323: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface wlan0, id 0
 ▶ Ethernet II, Src: Chongqin_8a:d5:01 (a8:93:4a:8a:d5:01), Dst: XiaomiCo_7b:89:5a (00:ec:0a:7b:89:5a)
 ▶ Internet Protocol Version 4, Src: 192.168.43.21, Dst: 5.188.150.79
 ▶ Transmission Control Protocol, Src Port: 44800, Dst Port: 8000, Seq: 0, Len: 0



Какие есть протоколы и зачем нам прокси?

Как выглядит подключение по TCP в wireshark

172271	53...	97.55.55.8	97.55.55.2	TCP	74	39374 → 21 [YN]	Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2546847757 TSecr=0 WS=128
172272	53...	97.55.55.2	97.55.55.8	TCP	4	21 → 39374 [SYN, ACK]	Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM TSval=3296917685 TSecr=2546847757 WS=128
172273	53...	97.55.55.8	97.55.55.2	TCP	66	39374 → 21 [CK]	Seq=1 Ack=1 Win=64256 Len=0 TSval=2546847758 TSecr=3296917685

На какой порт сервера мы передаём данные?

21

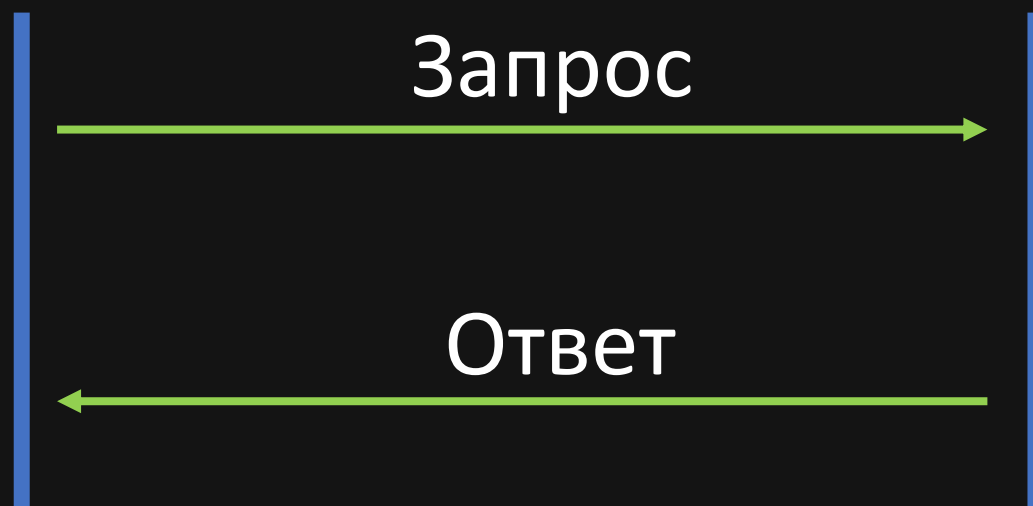


Какие есть протоколы и зачем нам прокси?

UDP

Клиент

Сервер





Какие есть протоколы и зачем нам прокси?

udp						
No.	Time	Source	Destination	Protocol	Length	Info
1311...	8...	97.55.55.8	97.55.55.2	UDP	342	48579 → 43100 Len=300
1311...	8...	97.55.55.2	97.55.55.8	ICMP	370	Destination unreachable (Port unreachable)
2647...	10...	97.55.55.2	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question
3954...	36...	97.55.55.2	224.0.0.251	MDNS	87	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _ipp._tcp.local, "QM" question

TCP используется для передачи данных без повреждения(CRC), а UDP – с повреждением данных, но с большой скоростью

→ **TCP** – Используем для критически важных данных везде (нам не нужно, чтобы на сайте появилось что-то взломное)

→ **UDP** – Используем где нужна скорость, а не качество данных (Видеоролики например)



Какие есть протоколы и зачем нам прокси?

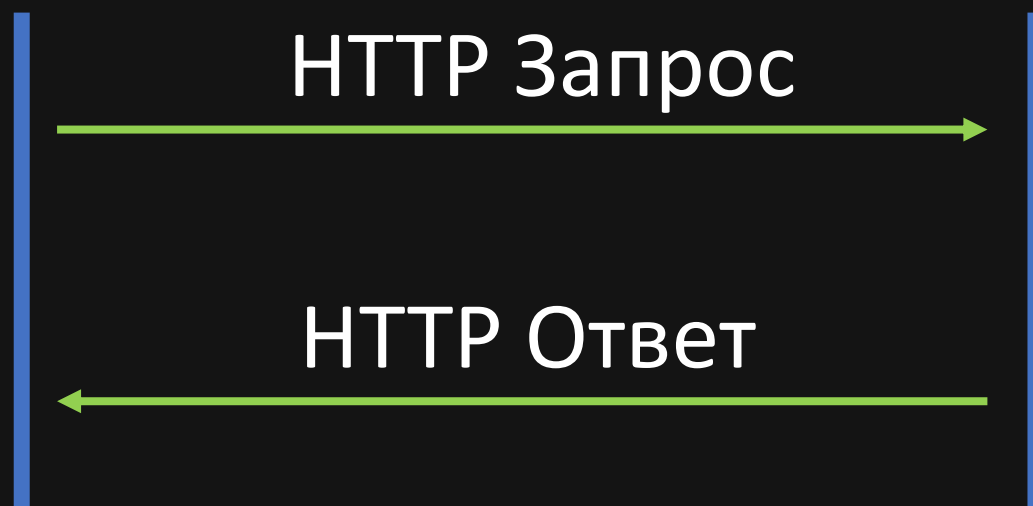
Протокол **HTTP** (используется для сайтов)



HTTPS - актуальнее

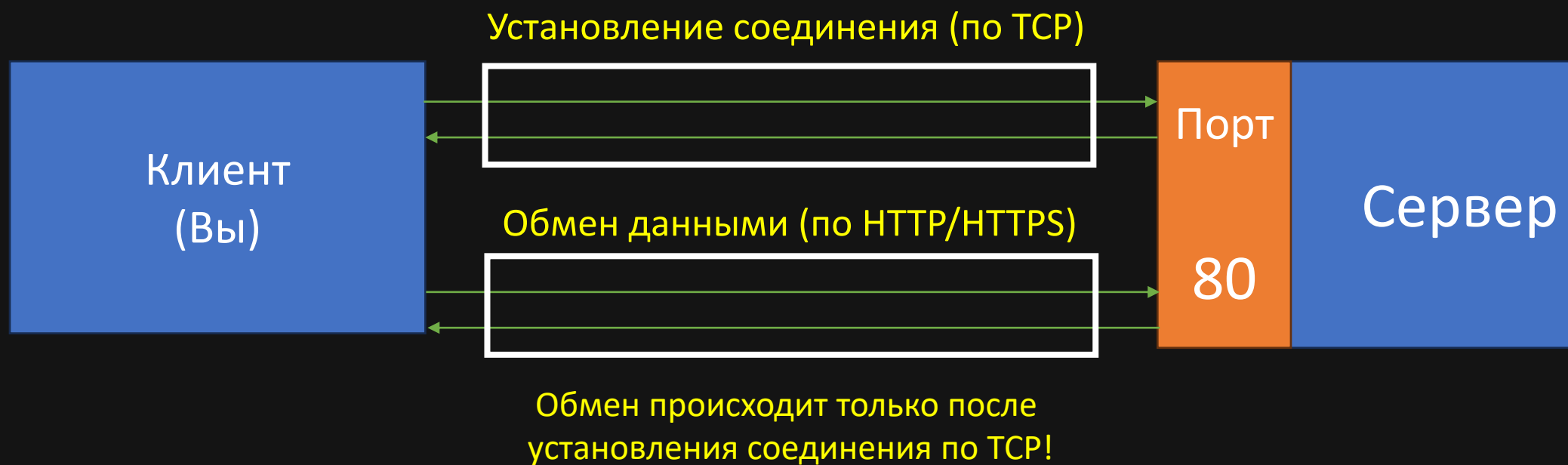
Клиент

Сервер





Вспоминаем



* Порт может быть любым, не обязательно 80 или 443



Какие есть протоколы и зачем нам прокси?

Используем фильтр **http** и смотрим весь трафик по http

http						
No.	Time	Source	Destination	Protocol	Length	Info
131097	7....	97.55.55.8	97.55.55.2	HTTP	84	GET / HTTP/1.0
131099	7....	97.55.55.2	97.55.55.8	HTTP	513	HTTP/1.1 200 OK (text/html)
131156	9....	97.55.55.8	97.55.55.2	HTTP	238	GET /nmaplowercheck1681850705 HTTP/1.1
131158	9....	97.55.55.8	97.55.55.2	HTTP	84	GET / HTTP/1.0
131160	9....	97.55.55.8	97.55.55.2	HTTP	680	POST /sdk HTTP/1.1
131162	9....	97.55.55.2	97.55.55.8	HTTP	513	HTTP/1.1 200 OK (text/html)
131164	9....	97.55.55.2	97.55.55.8	HTTP	544	HTTP/1.1 404 Not Found (text/html)
131165	9....	97.55.55.2	97.55.55.8	HTTP	523	HTTP/1.1 404 Not Found (text/html)
131182	9....	97.55.55.8	97.55.55.2	HTTP	219	GET /HNAP1 HTTP/1.1
131185	9....	97.55.55.8	97.55.55.2	HTTP	224	GET /evox/about HTTP/1.1
131187	9....	97.55.55.2	97.55.55.8	HTTP	525	HTTP/1.1 404 Not Found (text/html)
131188	9....	97.55.55.2	97.55.55.8	HTTP	530	HTTP/1.1 404 Not Found (text/html)
131200	9....	97.55.55.8	97.55.55.2	HTTP	84	GET / HTTP/1.0
131202	9....	97.55.55.2	97.55.55.8	HTTP	513	HTTP/1.1 200 OK (text/html)
131210	9....	97.55.55.8	97.55.55.2	HTTP	102	GET / HTTP/1.1
131212	9....	97.55.55.2	97.55.55.8	HTTP	494	HTTP/1.1 200 OK (text/html)



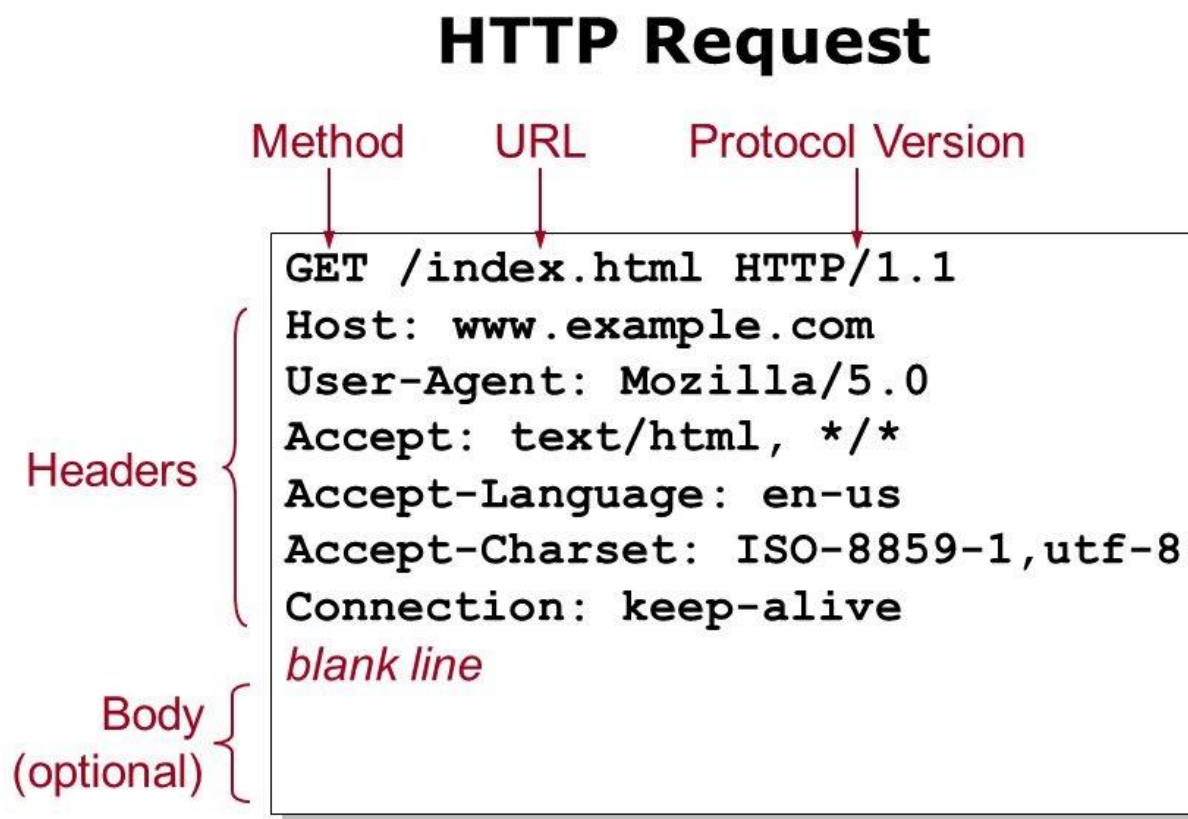
Какие есть протоколы и зачем нам прокси?

Структура HTTP запроса:

Метод URL Версия протокола

Заголовки (Указывают на доп.информацию)

Тело (в нём находятся данные)





Какие есть протоколы и зачем нам прокси?

Методы (говорят, что мы хотим)



GET

Получение веб-ресурса
(запрашиваем страничку. Простая загрузка сайта)



POST

Отправление данных на веб-ресурс
(Авторизация, регистрация. Всё, куда мы вводим данные)

* Есть ещё методы, но на данный момент ограничимся этими



Какие есть протоколы и зачем нам прокси?

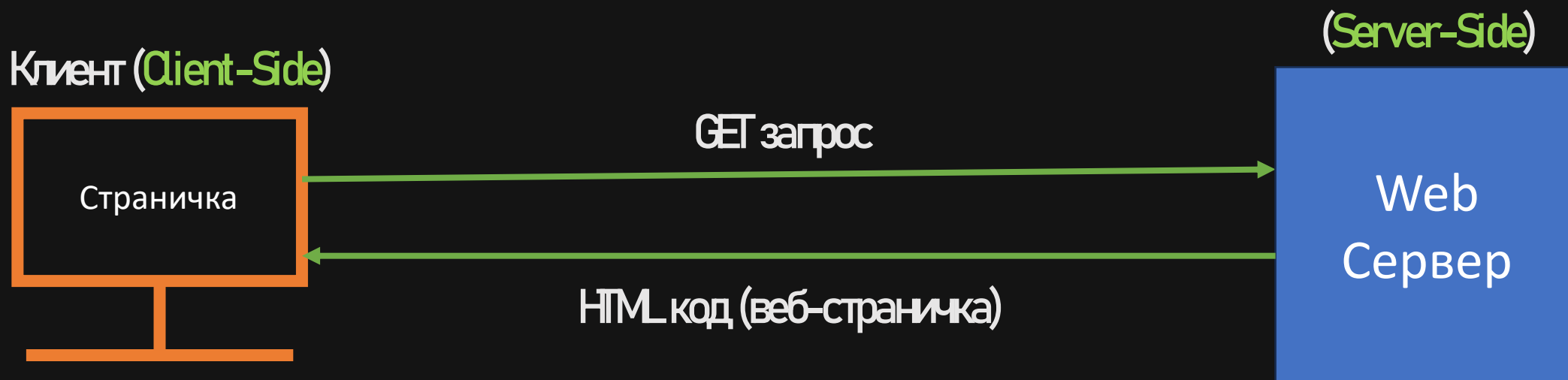
А теперь сравните трафик по TCP и HTTP. Какой из этих протоколов нам интересен с точки зрения атаки на сайты?

(Откройте Wireshark и посмотрите пакеты TCP и HTTP. Где информация, которую вы передаёте для сайта)



Какие есть протоколы и зачем нам прокси?

Как мы получаем сайт?

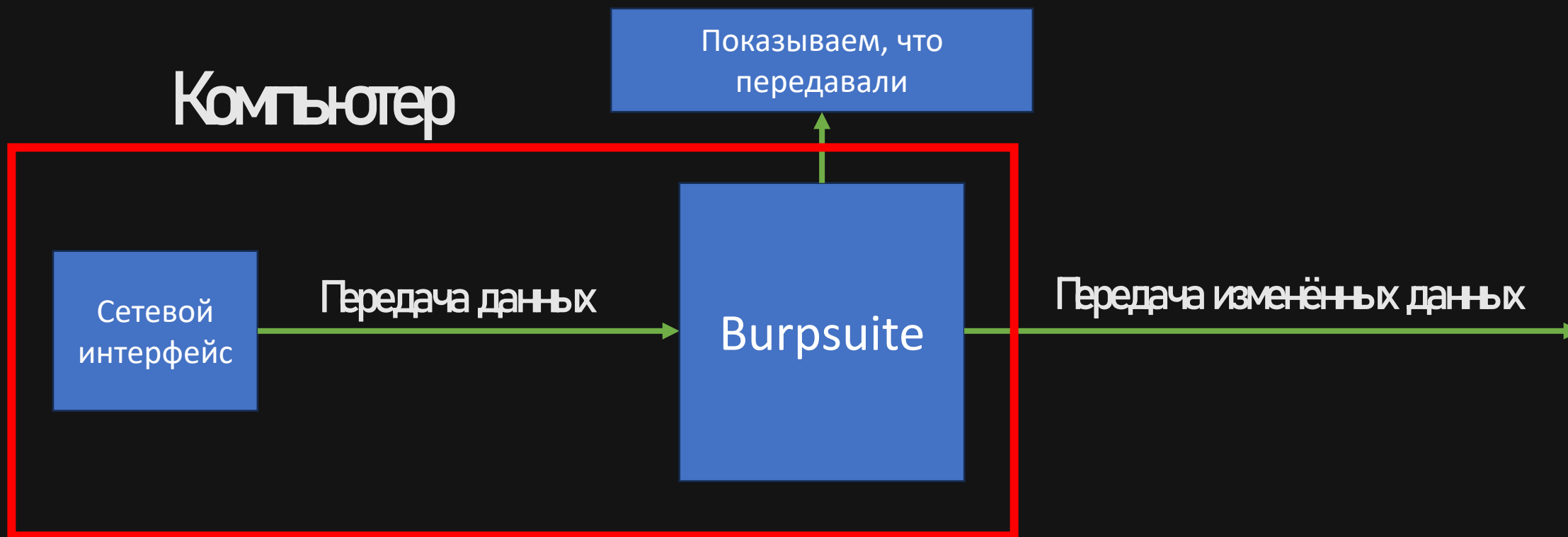


Браузер обрабатывает HTML код и выдаёт страничку



Какие есть протоколы и зачем нам прокси?

Как работает **burpsuite**?





Как создать сайт и что нам искать?

Создание сайта:

→ Создаём файл `index.html`

→ Запускаем PHP или XAMPP

Исмотрим, как все работает в [wireshark](#) и [burpsuite](#)



Как создать сайт и что нам искать?

Создание сайта:

→ Создаём файл `index.html`

→ Запускаем PHP или XAMPP

Исмотрим, как все работает в [wireshark](#) и [burpsuite](#)



Как создать сайт и что нам искать?

HTML – язык разметки

(НЕ ПРОГРАММИРОВАНИЯ!!)

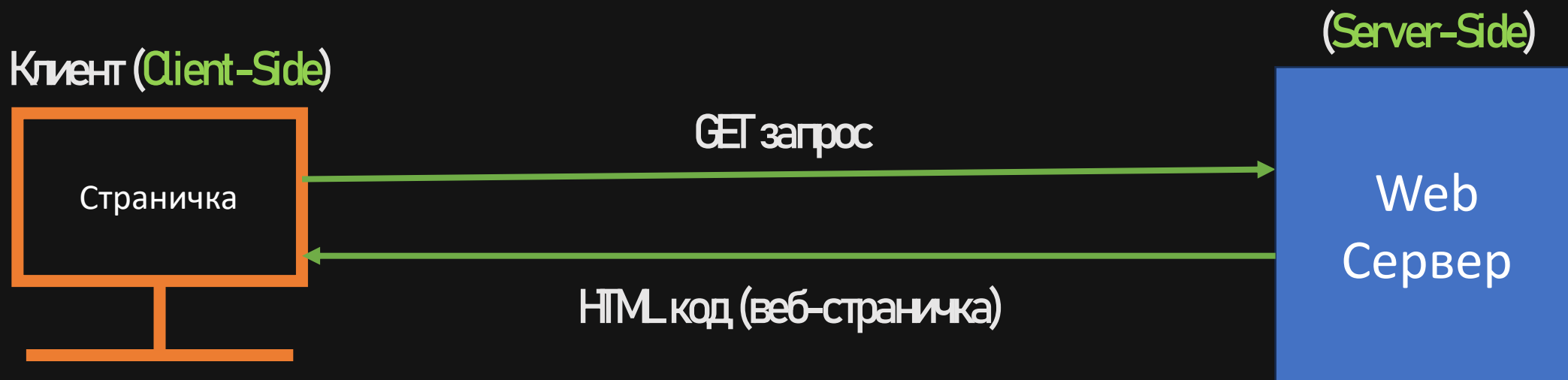
(Он определяет расположение элементов и структуру страниц)

```
1
2  <!DOCTYPE html>
3
4  <html>
5
6      <head>
7
8          <title>JS</title>
9          <meta charset="utf-8">
10
11      </head>
12
13      <body>
14
15          <script src="main.js"></script>
16
17      </body>
18
19  </html>
20
```



Ещё раз

Как мы получаем сайт?



Браузер обрабатывает HTML код и выдаёт страничку



Как создать сайт и что нам искать?

HTTP Request

Method URL Protocol Version

↓ ↓ ↓

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html, */*
Accept-Language: en-us
Accept-Charset: ISO-8859-1,utf-8
Connection: keep-alive
blank line
```

Headers {

Body (optional) {



Как создать сайт и что нам искать?

Если занятие ещё не закончилось – рассказываю про html, cookie и прочее. Решаем `natas`

Наш чат во вконтакте



Наш чат во телеграмме



Как меня найти



