

# Основы построения защищенных компьютерных сетей

Лекция 2

Базовые сетевые протоколы и их безопасность



Семён Новосёлов

2024

**БФУ**  
ИМЕНИ И. КАНТА

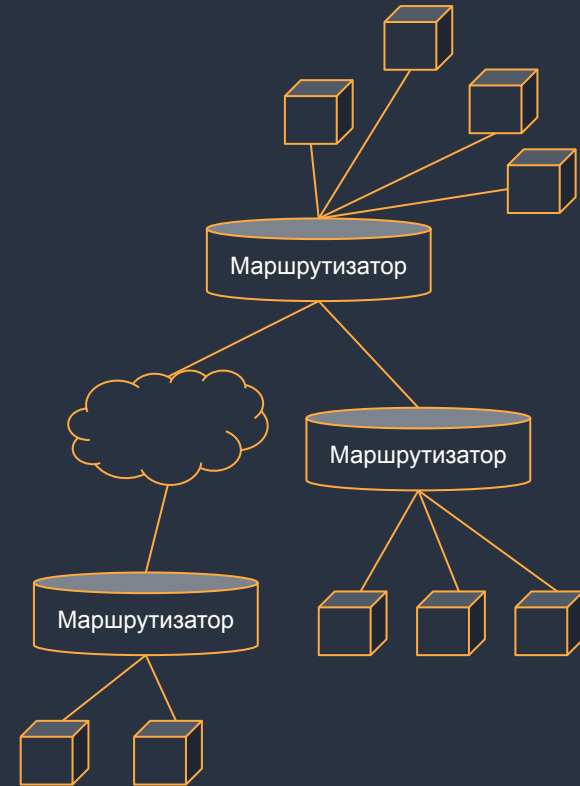
# Устройство сети Интернет

**Стек TCP/IP** — базовый набор протоколов на котором строится Интернет.

- данные разбиваются на кусочки (сегменты, кадры, пакеты) и передаются по очереди
- организован по принципу “**матрёшки**”
- делится на уровни, которые вкладываются друг в друга:
  - **Канальный уровень** - физическая передача данных (Ethernet, Wi-Fi, L2TP)
  - **Сетевой уровень** - передача данных между физ. сетями (IPv4, IPv6, ICMP)
  - **Транспортный уровень** - установление базовых каналов передачи данных (TCP, UDP)
  - **Прикладной уровень** - протоколы приложений и сервисов (DNS, HTTP, SSH)

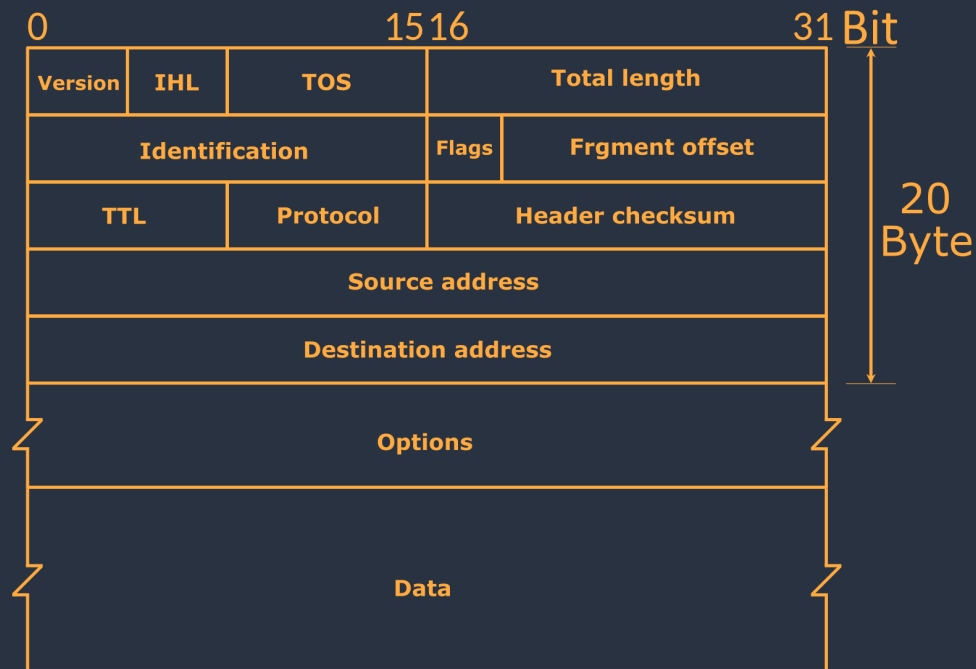
# Протокол IP

- предполагается, что можно передавать данные по физическому каналу
- отвечает за маршрутизацию
  - каждому узлу назначается адрес (IP-адрес, 32 бит в IPv4, 128 бит в IPv6)
- для передачи пакетов по физическим сетям с разной пропускной способностью используется **фрагментация**
- нет гарантий доставки пакета



# Формат пакета

- чтобы передать пакет необходимо сформировать заголовок с IP-адресом отправителя и получателя и данными



# Traceroute. 1/2

Проследить путь пакета до заданного узла позволяют инструменты `tracert`/`tracert`/`tracert`

Они используют поле “время жизни пакета” (TTL) - число промежуточных хостов.

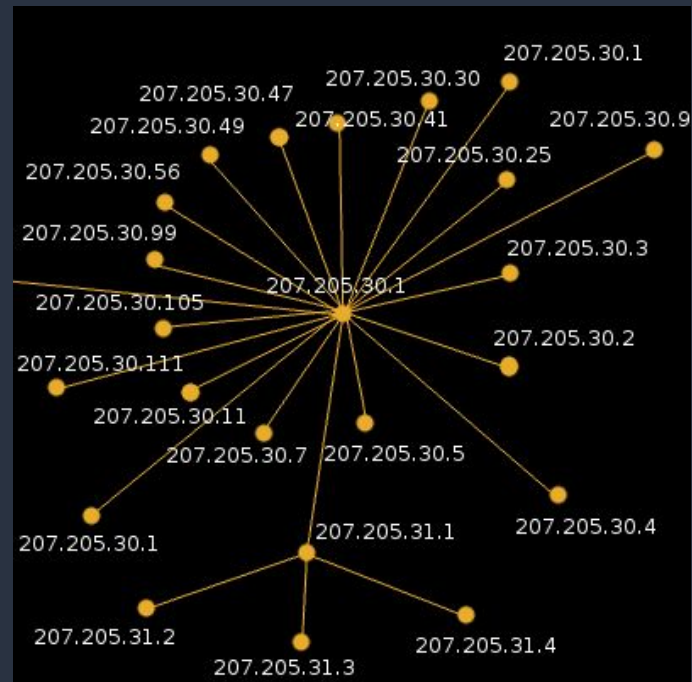
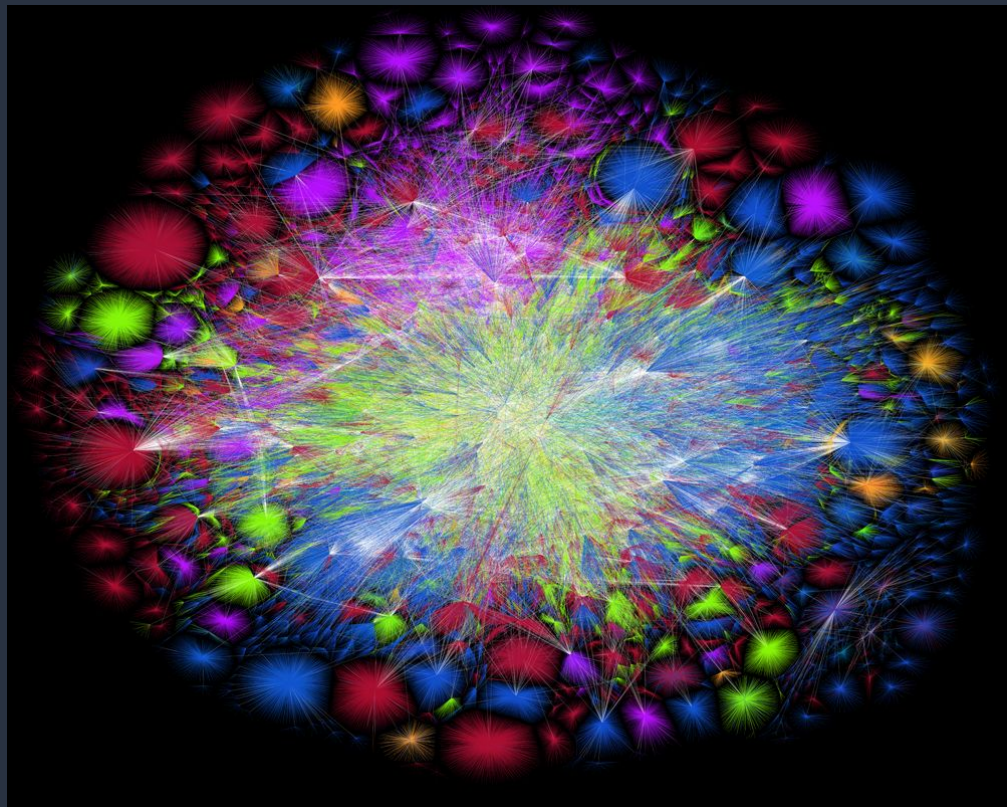
- TTL уменьшается на единицу при прохождении каждого узла
- если TTL = 0, то хост отправляет сообщение ICMP Time Exceeded
- утилита отправляет последовательно пакеты с TTL=1,2,3, ...
- промежуточные узлы возвращают пакет ICMP Time Exceeded с IP-адресом узла в адресе отправителя

# Traceroute. 2/2

```
n7v@n7v-laptop:~$ tracepath -b yandex.ru
```

```
1?: [LOCALHOST] pmtu 1400
1:  _gateway (192.168.1.1) 1.343ms
1:  _gateway (192.168.1.1) 0.780ms
2:  37-147-176-1.broadband.corbina.ru (37.147.176.1) 1.460ms
3:  81.211.111.30 (81.211.111.30) 1.861ms
4:  spb-62-141-124-161.sovintel.spb.ru (62.141.124.161) 4.811ms
5:  pe16.Moscow.gldn.net (79.104.235.207) 36.149ms
6:  no reply
```

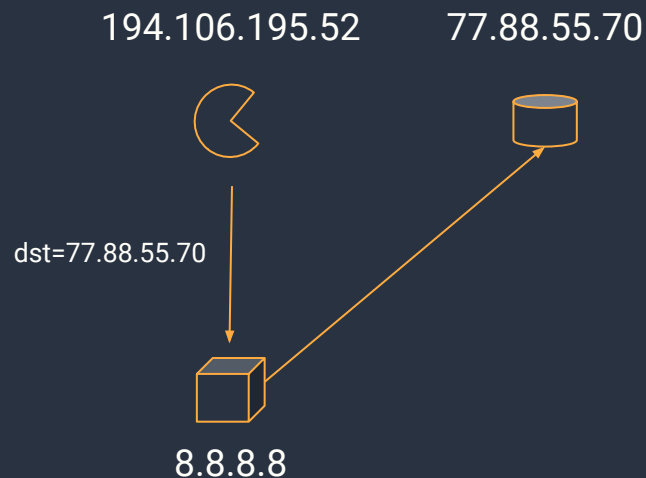
# Карта сети Интернет - 2015 (The Opte Project)



Пример узла сети

# Протокол IP. Безопасность I

- адрес отправителя можно подделать (**IP-spoofing**)
- можно послать пакет от любого IP-адреса
- ответный пакет придёт на поддельный адрес
- **DoS-атаки**
  - сокрытие источника атаки
  - усиление атаки за счёт провоцирования больших ответных пакетов (например, **DNS**)





# Протокол IP. Безопасность II

- **фрагментация** может использоваться для обхода систем защиты
  - для анализа содержимого пакета его нужно собрать
  - для этого требуется хранить фрагментированные пакеты в буфере, который имеет ограниченный размер
  - хакер может:
    - посылать пакеты с задержками по времени
    - в разном порядке
  - в нагруженных системах это может привести к:
    - переполнению буфера пакетов
    - пропуску непроверенных пакетов в сеть (обход антивирусов, COV, файерволов)
    - DoS

# Протокол IP. Безопасность III

- перебор адресов получателя и отправка ping-запросов по протоколу ICMP позволяет идентифицировать узлы сети
- в протоколе отсутствует шифрование

# Порты

- протокол IP позволяет передать данные между узлами сети
- для распределения данных по сервисам (напр., программам на компьютере) служат **порты** - номера сервисов
- Протоколы: UDP/TCP

## Примеры

80	HTTP/Skype
443	HTTPS
22	SSH

# Транспортный уровень. Протокол UDP

UDP = IP + Порты + Контрольная сумма (заголовка)

- Доставка пакета и порядок пакетов не гарантируется
- Используется там, где это не критично (напр., передача видео)

# Транспортный уровень. Протокол ТСР

ТСР = IP + Порты + Гарантии доставки пакета

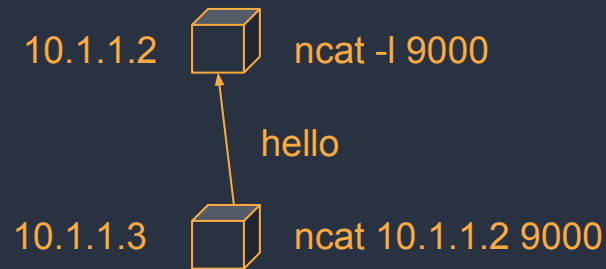
- Поддерживается создание соединения
- Пакеты нумеруются – гарантия правильного порядка сборки пакетов
- **Основной принцип:** если не получено подтверждение доставки от получателя (ответный пакет с флагом АСК) в течении некоторого времени, то отослать пакет заново.

# Утилита netcat

- используется для подключения по TCP/UDP и отсылки данных, введенных в консоли
- **nc** в Linux
- **ncat** из состава Nmap в Windows/Linux
- подключится можно к любому компьютеру/серверу в сети Интернет

# Пример. Чат по TSP между двумя компьютерами. 1/2

1. Для создания соединения следует один из компьютеров выбрать в качестве сервера
  2. Второй компьютер к нему подключается
  3. После подключения все введенные данные отправляются в обе стороны
- Связь работает в рамках локальной сети
    - например, компьютеры подключены к одной Wi-Fi точке доступа и нет изоляции клиентов
  - Для связи по сети Интернет необходимо, чтобы один из компьютеров имел внешний (не локальный) IP-адрес



# Пример. Чат по TCP между двумя компьютерами. 2/2

Компьютер-сервер

```
Командная строка - ncat -l 9000
C:\Users\snovalov>ncat -l 9000
Coolhacker: The world is doomed
Server: Sure
```

Порт, можно выбрать любое значение

Компьютер-клиент:

```
Командная строка - ncat 127.0.0.1 9000
C:\Users\snovalov>ncat 127.0.0.1 9000
Coolhacker: The world is doomed
Server: Sure
```

IP-адрес сервера, может быть любой адрес в Интернете



# Как это выглядит на уровне сетевых пакетов?

Для просмотра пакетов используются анализаторы сетевого трафика.



Wireshark - наиболее известный и продвинутый анализатор  
([www.wireshark.org](http://www.wireshark.org))

Захват из Adapter for loopback traffic capture

Файл Редактирование Просмотр Запуск Захват Анализ Статистика Телефония Беспроводной Инструменты Помощь

tcp.port == 9000 && ip.addr==127.0.0.1

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	1571 → 9000 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=25
2	0.000041	127.0.0.1	127.0.0.1	TCP	56	9000 → 1571 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=
3	0.000065	127.0.0.1	127.0.0.1	TCP	44	1571 → 9000 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	12.720132	127.0.0.1	127.0.0.1	S101	76	1571 → 9000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=32
5	12.720197	127.0.0.1	127.0.0.1	TCP	44	9000 → 1571 [ACK] Seq=1 Ack=33 Win=2619648 Len=0
6	19.551216	127.0.0.1	127.0.0.1	S101	57	9000 → 1571 [PSH, ACK] Seq=1 Ack=33 Win=2619648 Len=13
7	19.551305	127.0.0.1	127.0.0.1	TCP	44	1571 → 9000 [ACK] Seq=33 Ack=14 Win=2619648 Len=0
12	56.725784	127.0.0.1	127.0.0.1	TCP	44	9000 → 1571 [RST, ACK] Seq=14 Ack=33 Win=0 Len=0

> Frame 1: 56 bytes on wire (448 bits), 56 bytes captured (448 bits) on interface \Device\NPF\_Loopback, id 0

- Null/Loopback
  - Family: IP (2)
    - Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
    - Transmission Control Protocol, Src Port: 1571, Dst Port: 9000, Seq: 0, Len: 0

```

0000  02 00 00 00 45 00 00 34  84 b0 40 00 80 06 00 00  . . . . E . . 4 . . @ . . . .
0010  7f 00 00 01 7f 00 00 01  06 23 23 28 52 d6 bc da  . . . . . ##(R . .
0020  00 00 00 00 80 02 ff ff  3d ee 00 00 02 04 ff d7  . . . . . = . . . . .
0030  01 03 03 08 01 01 04 02  . . . . .
  
```

Family (null.family), 4 байт | Пакеты: 24 · Показаны: 8 (33.3%) | Профиль: Default

Пакеты TCP для примера с чатом.

```
4 12.720132 127.0.0.1 127.0.0.1 S101 76 1571 → 9000 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=32
> Frame 4: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_Loopback, id 0
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 1571, Dst Port: 9000, Seq: 1, Ack: 1, Len: 32
Data (32 bytes)
Data: 436f6f6c6861636b65723a2054686520776f726c642069732064666f6d65640a
[Length: 32]

0000 02 00 00 00 45 00 00 48 84 b3 40 00 80 06 00 00 .....E..H..@.....
0010 7f 00 00 01 7f 00 00 01 06 23 23 28 52 d6 bc db .....##(R...
0020 e8 78 3a a6 50 18 27 f9 3d 1e 00 00 43 6f 6f 6c .x:P.'..=...Cool
0030 68 61 63 6b 65 72 3a 20 54 68 65 20 77 6f 72 6c hacker: The worl
0040 64 20 69 73 20 64 6f 6f 6d 65 64 0a d is doo med.
```

Пакет TCP сообщением от компьютера-клиента.

Шифрование не предусмотрено (нужно использовать TLS-протокол).

# Сканирование портов

- Для нахождения сервисов и программ доступных на целевой системе можно использовать перебор
- В протоколе TCP/UDP предусмотрено всего 65535 портов
- Для сканирования есть множество различных техник с целью добиться ответа от системы в том числе при наличии средств защиты
- В основном техники основаны на манипуляциях с полями в заголовке протокола

# Сканирование портов. Пример для роутера

```
Командная строка
C:\Users\snoyoselov>nmap 192.168.1.1
Starting Nmap 7.70 ( https://nmap.org ) at 2021-01-21 21:27 RTZ 1 (ceia)
Nmap scan report for 192.168.1.1
Host is up (0.00028s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
23/tcp    open  telnet
53/tcp    open  domain
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: EC:43:F6:04:AF:28 (ZyXEL Communications)

Nmap done: 1 IP address (1 host up) scanned in 1.51 seconds
```

как видно здесь есть telnet, можно попытаться подобрать пароль:

```
nmap -p 23 --script telnet-brute 192.168.1.1
```

# Определение версий ПО. 1/2

- определение версии ОС
  - анализ ответов хоста на нестандартные IP/UDP/TCP пакеты
  - значения по-умолчанию для заголовков пакетов (например, размер окна TCP)
  - используемые начальные значения последовательностей
  - статистические характеристики генераторов случайных чисел
- определение версий сервисов и программ
  - часто при подключении сервер возвращает строку-баннер с версией  
пример: "Server: nginx/1.4.2"
  - "отпечатки" с ответами на специально составленные запросы

## Определение версий ПО. 2/2

- сканер **nmap** содержит реализацию большинства техник и вместе с базами запросов (nmap-service-probes)
- после определения версий ПО, их можно проверить по базам уязвимостей и эксплойтов (NVD, exploit-db.com)

# Пример. Определение версий с помощью Nmap

Определение версий  
сервисов на машине  
с Metasploitable 2

`nmap -sV 192.168.1.45`

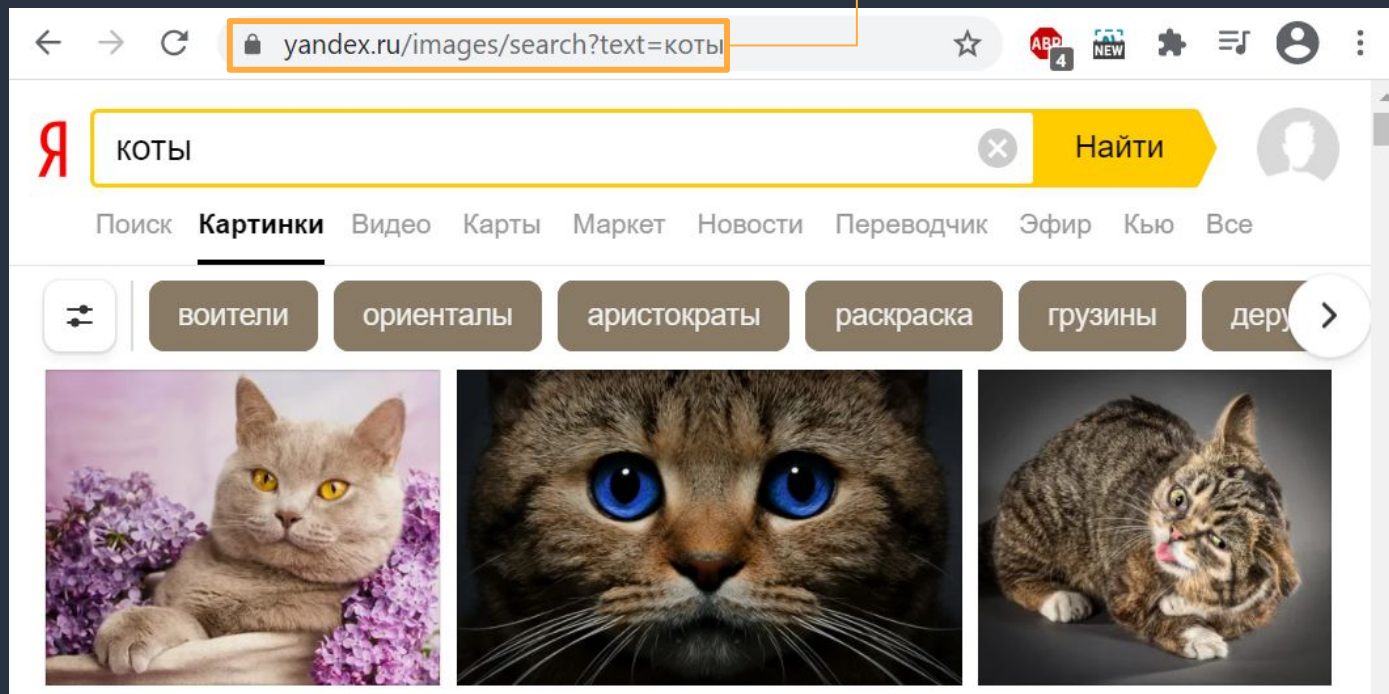
```
Nmap scan report for 192.168.1.45
Host is up (0.000015s latency).
Not shown: 977 filtered ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp         vsftpd 2.3.4
22/tcp    open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet      Linux telnetd
25/tcp    open  smtp        Postfix smtpd
53/tcp    open  domain      ISC BIND 9.4.2
80/tcp    open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind     2 (RPC #100000)
139/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec        netkit-rsh rexecd
513/tcp   open  login?
514/tcp   open  tcpwrapped
1099/tcp  open  rmiregistry GNU Classpath grmiregistry
1524/tcp  open  bindshell   Metasploitable root shell
2049/tcp  open  nfs         2-4 (RPC #100003)
2121/tcp  open  ftp         ProFTPD 1.3.1
3306/tcp  open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc         VNC (protocol 3.3)
6000/tcp  open  x11         (access denied)
6667/tcp  open  irc         UnrealIRCd
8009/tcp  open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp  open  http        Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploi
table.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel
```



# Прикладные протоколы

- **DNS** (Domain Name System) — преобразование удобных для человека строк (доменных имён) в IP-адреса.
- **HTTP** — протокол лежащий в основе сети World Wide Web (WWW)
- **HTTPS** — протокол HTTP, вложенный в протокол TLS

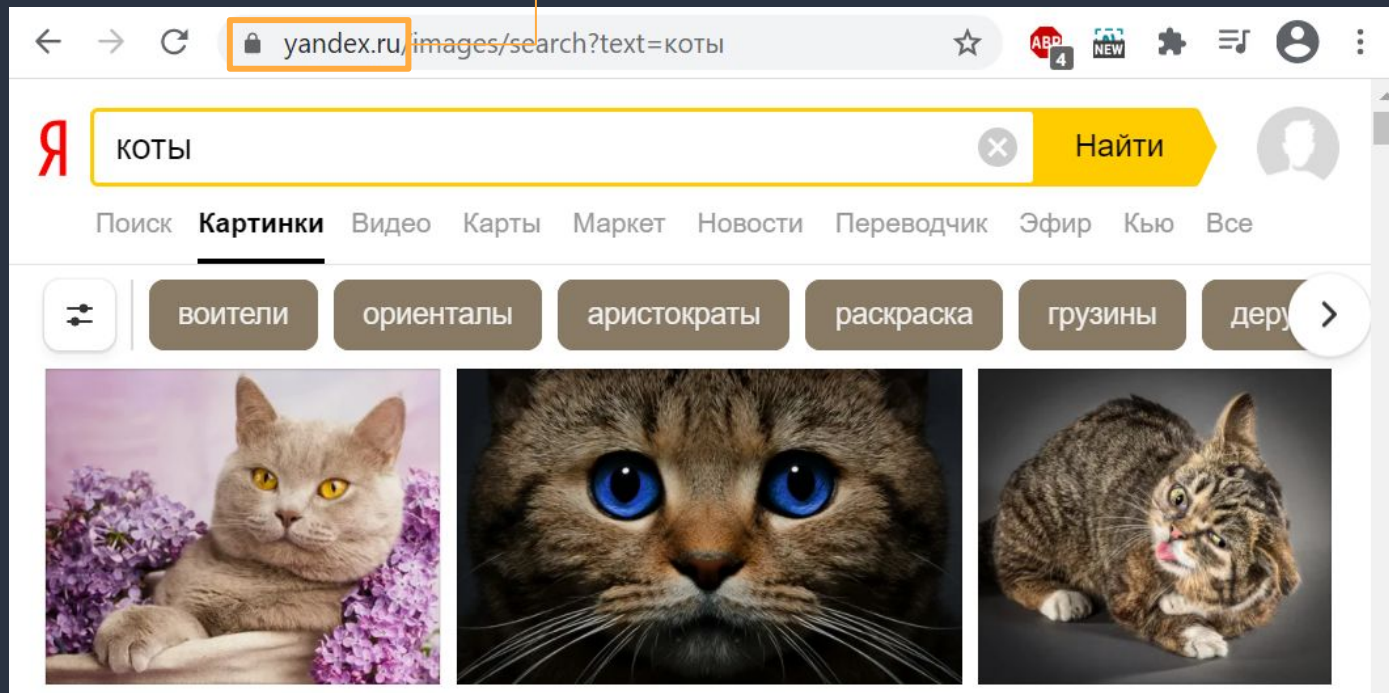
# Что происходит при открытии ссылки в браузере? 1/5



Шаг 1. Браузер разбивает ссылку (URL) на части:

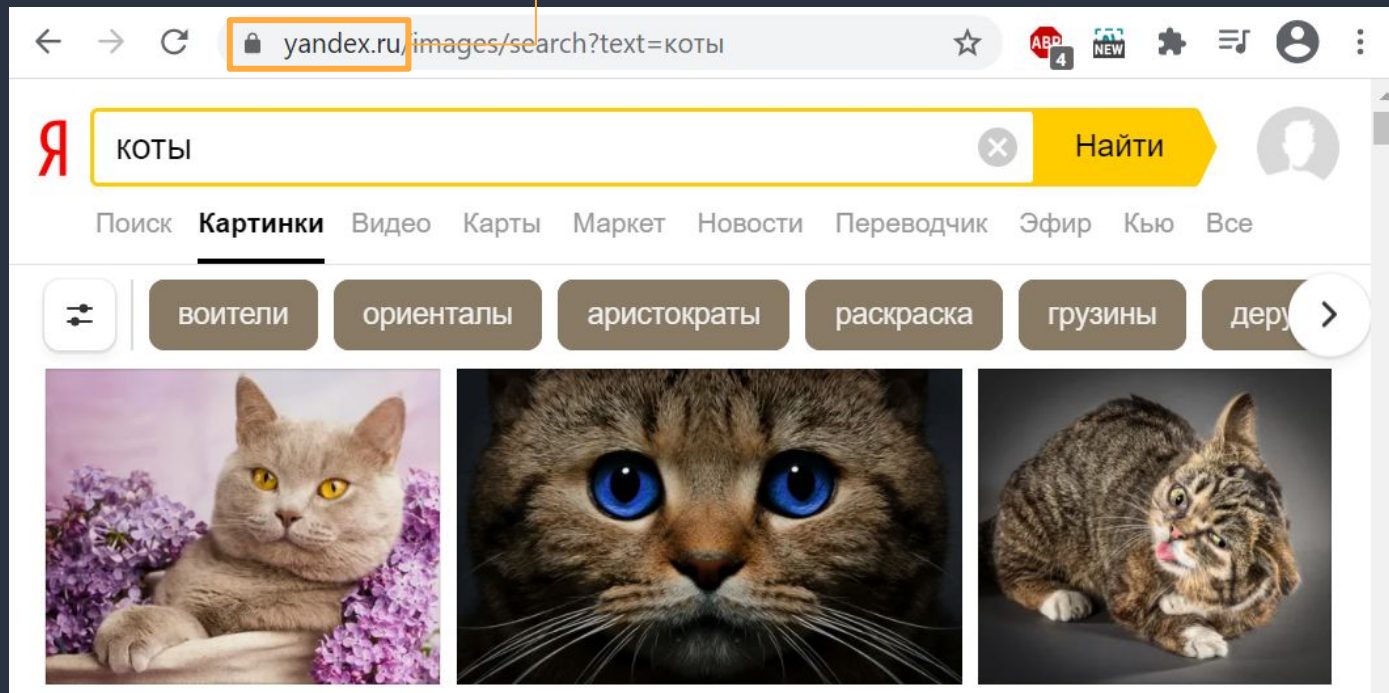
- 1) **yandex.ru** - имя хоста
- 2) **/images/search?text=коты** - путь и параметры запроса

# Что происходит при открытии ссылки в браузере? 2/5



Шаг 2. Браузер преобразует **yandex.ru** в IP-адрес **77.88.55.50** с помощью DNS

# Что происходит при открытии ссылки в браузере? 3/5



Шаг 3. Браузер подключается по протоколу TCP к IP-адресу **77.88.55.50** с портом **80** (HTTP) или **443** (HTTPS, в этом случае дополнительно используется TLS)

# Что происходит при открытии ссылки в браузере? 4/5

Шаг 4. Браузер посылает строку (HTTP-запрос) вида:

```
GET /images/search?text=%D0%BA%D0%BE%D1%82%D1%8B HTTP/1.1
Host: yandex.ru
```

В ответ получает:

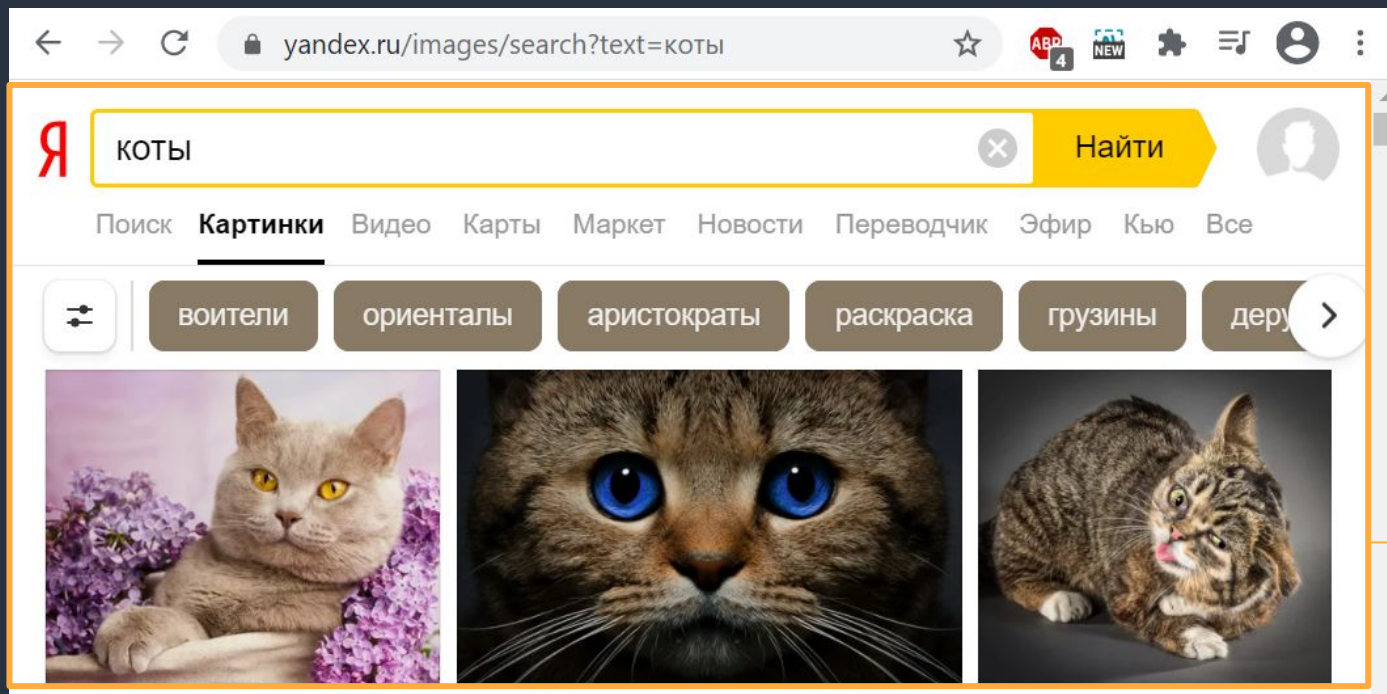
```
HTTP/1.1 200 OK
Cache-Control: private, max-age=3600
Content-Type: text/html; charset=utf-8
...
Date: Thu, 21 Jan 2021 15:35:47 GMT
```

HTTP-заголовки  
ответа

```
<!DOCTYPE html><html class="i-ua_js_no i-ua_css_standard" lang="ru">
<head><meta charset="utf-8"><meta http-equiv="X-UA-Compatible"
content="IE=edge"><title>Яндекс.Картинки</title>
...
```

html-код

# Что происходит при открытии ссылки в браузере? 4/4



Шаг 5. Браузер разбирает и отрисовывает html-код, подгружая аналогичным образом картинки (GET-запросы HTTP) и другие ресурсы

# Протокол HTTP

Основные HTTP-методы: GET/POST/HEAD

- **GET** — запрос данных
- **HEAD** — запрос только заголовков, без тела ответа
- **POST** — отправка данных на сервер (пример: формы, файлы)

Протокол HTTP не хранит состояние, т.е. сервер не хранит информацию о прошлых запросах.

Информацию нужно передавать в заголовках запроса (Cookie) и/или её запоминают в базе скрипты, вызываемые сервером (пример: PHP, Ruby, Python)

# Протокол HTTP. Безопасность

- протокол не включает в себя шифрование
  - например, в публичной Wi-Fi сети все данные по HTTP можно перехватить
  - это один из каналов утечки паролей/логинов пользователей при заходе на сайт
- необходимо использовать **HTTPS = HTTP + TLS**
- для обработки запроса на сервере используются скрипты (PHP/Python/Ruby), куда передаются параметры запроса (пример, **text=коты**)
- ошибки в работе скриптов - основной источник уязвимостей на сайтах
- как правило, уязвимости находятся подстановкой нестандартных значений в параметры HTTP-запросов



# Литература и ссылки

- Дуглас Э. Камер - Сети TCP IP. Принципы, протоколы и структура (2003)
- Metasploitable 2 и 3 - системы для тренировки
  - <https://sourceforge.net/projects/metasploitable/>
  - <https://github.com/brimstone/metasploitable3/releases>
- Gordon Lyon. Nmap Network Scanning. The Official Nmap Project Guide to Network Discovery and Security Scanning (2011) (<http://nmap.org/book/>)
- Список скриптов Nmap:  
<https://nmap.org/nsedoc/>