

Тема: Анализаторы трафика.

The image shows a Wireshark window titled "(Untitled) - Wireshark" with a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Help) and a toolbar. Below the toolbar is a filter bar with a "Filter:" input, a dropdown arrow, and buttons for "Expression...", "Очистить", and "Применить". The main display area shows a packet list table with columns: No., Time, Source, Destination, Protocol, and Info.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.3.254	224.0.0.22	IGMP	V3 Membership Report
2	0.180020	192.168.3.254	224.0.0.251	MDNS	Standard query ANY stilo.local
3	0.355964	192.168.3.254	224.0.0.22	IGMP	V3 Membership Report
4	0.709465	192.168.3.254	224.0.0.251	MDNS	Standard query ANY stilo.local
5	0.715974	192.168.3.254	224.0.0.22	IGMP	V3 Membership Report
6	0.960072	192.168.3.254	224.0.0.251	MDNS	Standard query ANY stilo.local
7	1.076006	192.168.3.254	224.0.0.22	IGMP	V3 Membership Report
8	1.436031	192.168.3.254	224.0.0.22	IGMP	V3 Membership Report
9	1.556163	192.168.3			
10	1.796051	192.168.3			
11	2.004135	192.168.3			
12	2.007746	192.168.3			
13	2.152074	192.168.3			

Below the packet list, there is a packet details pane showing "Frame 1 (54 bytes on wire)". At the bottom, a file path is visible: "File: '/tmp/etherXXXXCBFIZT' 9690".

Overlaid on the bottom right is a terminal window titled "aag@stilo:~". It shows the command `tcpdump -i eth1` and its output, which includes the same network traffic events seen in the Wireshark packet list.

```
aag@stilo:~  
Файл Правка Вид Терминал Вкладки Справка  
stilo:/home/aag # tcpdump -i eth1  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes  
11:27:33.426627 IP stilo.asoiu > 224.0.0.22: igmp v3 report, 1 group record(  
11:27:33.430480 IP stilo.asoiu.mdns > 224.0.0.251.mdns: 0 [1n] ANY (Cache f  
)? stilo.local. (45)  
11:27:33.682544 IP stilo.asoiu.mdns > 224.0.0.251.mdns: 0 [1n] ANY? stilo.l  
. (45)  
11:27:33.786445 IP stilo.asoiu > 224.0.0.22: igmp v3 report, 1 group record(  
11:27:33.934930 IP stilo.asoiu.mdns > 224.0.0.251.mdns: 0 [1n] ANY (Cache f  
)? stilo.local. (45)  
11:27:34.142468 IP stilo.asoiu > 224.0.0.22: igmp v3 report, 1 group record(  
11:27:34.382534 IP stilo.asoiu.mdns > 224.0.0.251.mdns: 0 [1n] ANY (Cache f  
)? stilo.local. (45)  
11:27:34.502487 IP stilo.asoiu > 224.0.0.22: igmp v3 report, 1 group record(  
11:27:34.634629 IP stilo.asoiu.mdns > 224.0.0.251.mdns: 0 [1n] ANY (Cache f
```

План занятия:

1. Введение в анализ сетевого трафика
2. Wireshark. Основные функции и возможности
3. Захват трафика с помощью Wireshark
4. Анализ трафика с использованием Wireshark
5. tcpdump
6. Практические упражнения: захват и анализ сетевого трафика с использованием Wireshark.

1. Введение

Анализ сетевого трафика - это процесс мониторинга, записи, интерпретации и понимания данных, передаваемых через сеть между устройствами. Этот процесс включает в себя изучение различных аспектов сетевого взаимодействия, таких как протоколы, порты, адреса, типы данных и т. д. Целью анализа сетевого трафика является получение информации о работе сети, выявление проблем, обеспечение безопасности и оптимизация производительности.

Задачи анализа сетевого трафика:

Обнаружение и устранение сетевых проблем: Выявление и диагностика различных проблемы в сети, такие как перегрузки сети, потери пакетов, задержки, аномалии в сетевом взаимодействии и т. д.

Оптимизация производительности сети: Понимание того, какие типы данных передаются по сети, какие приложения и службы используют больше ресурсов сети, помогает оптимизировать конфигурацию сетевых устройств и ресурсов, повышая производительность и эффективность сети.

Обеспечение безопасности сети: Выявление и обнаружение потенциально вредоносной активности, атак, несанкционированных доступов и др. угроз для безопасности сети.

Планирование и мониторинг сетевых изменений: Анализ сетевого трафика помогает понять, как используются ресурсы сети, какие изменения могут быть необходимы для улучшения производительности или безопасности сети, и как эффективно внедрять эти изменения, минимизируя возможные негативные последствия.

В целом, анализ сетевого трафика играет ключевую роль в обеспечении эффективности, производительности и безопасности сети, а также в понимании её работы и выявлении проблем.

Некоторые инструменты анализа сетевого трафика:

Wireshark: Wireshark является одним из наиболее популярных и мощных инструментов для анализа сетевого трафика. Он обеспечивает возможность захвата и анализа трафика в реальном времени, а также анализ сохраненных файлов сетевого трафика.

tcpdump: Это утилита командной строки для захвата и анализа сетевого трафика в реальном времени.

tshark: Это консольная версия Wireshark, которая предоставляет те же функции, что и Wireshark, но без графического интерфейса.

NetworkMiner: Это инструмент с открытым исходным кодом для анализа сетевого трафика в операционной системе Windows.

Capsa: Коммерческое программное обеспечение для мониторинга и анализа сетевого трафика, которое обладает богатым набором функций и графическим интерфейсом.

2. Wireshark.

Основные функции и возможности

Wireshark – это широко распространённый инструмент для захвата и анализа сетевого трафика, который активно используется как для образовательных целей, так и для устранения неполадок на компьютере или в сети. Wireshark работает практически со всеми протоколами модели OSI, обладает понятным для обычного пользователя интерфейсом и удобной системой фильтрации данных. Помимо всего этого, программа является кроссплатформенной и поддерживает следующие операционные системы: Windows, Linux, Mac OS X, Solaris, FreeBSD, NetBSD, OpenBSD.

Установка Wireshark

Linux (Ubuntu/Debian):

Откройте терминал.

Установите Wireshark и необходимые пакеты с помощью команды:

```
sudo apt update  
sudo apt install wireshark
```

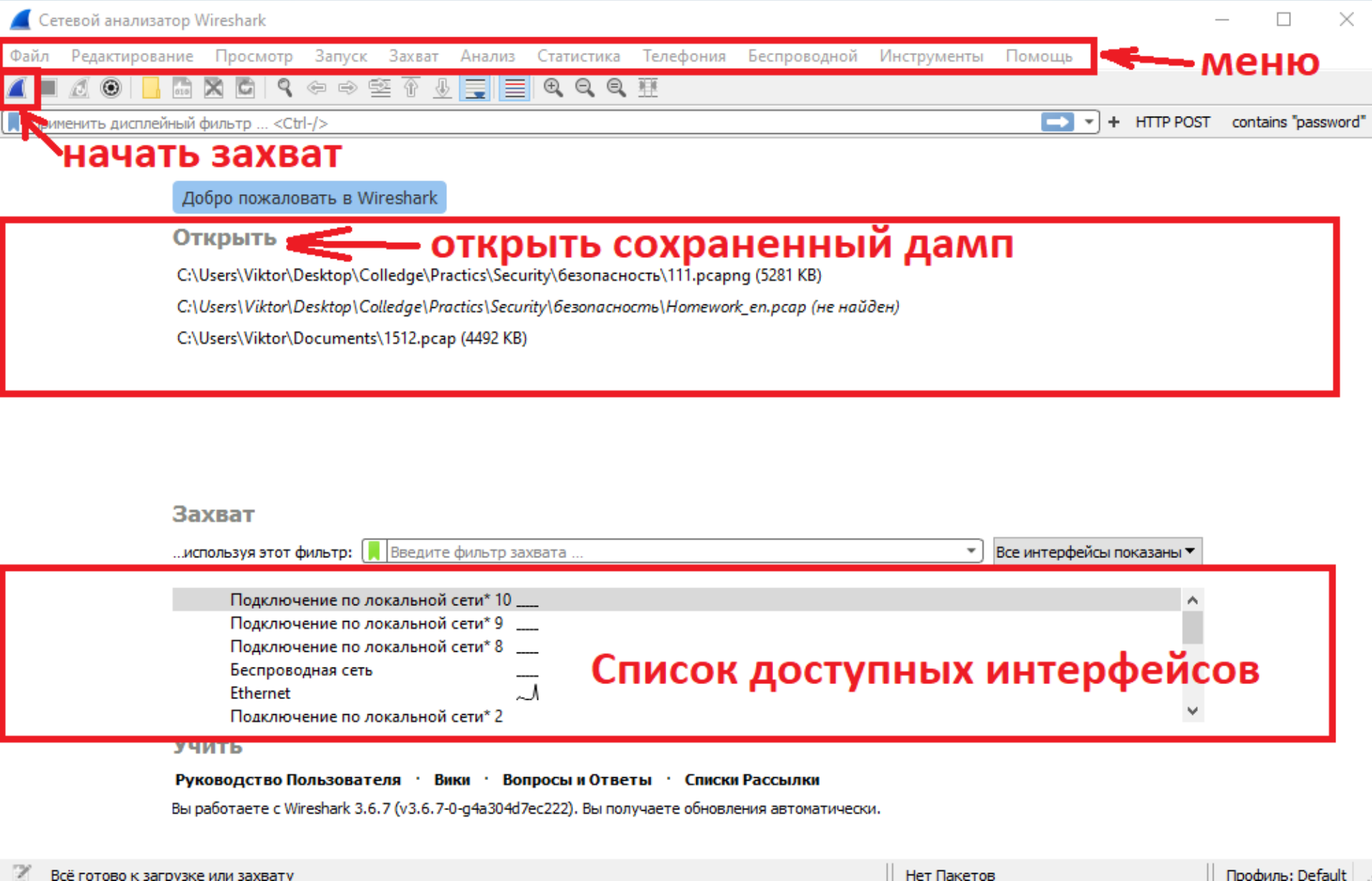
В процессе установки вам может быть предложено добавить пользователя в группу wireshark, чтобы иметь возможность запускать Wireshark без прав суперпользователя. Выберите "Да", если хотите использовать эту функцию.

После завершения установки Wireshark будет доступен из меню приложений или через команду **wireshark** в терминале.

Windows:

1. Перейдите на официальный сайт Wireshark:
<https://www.wireshark.org/>.
2. Нажмите на кнопку "Download" или "Download Wireshark".
3. Выберите версию Wireshark для Windows, соответствующую вашей архитектуре процессора (32-битная или 64-битная).
4. Скачайте установочный файл Wireshark.
5. Запустите установочный файл и следуйте инструкциям мастера установки.
6. После завершения установки Wireshark будет доступен в меню "Пуск" или в списке установленных программ.

Интерфейс Wireshark. Стартовое окно.



Интерфейс Wireshark. Окно захвата.

The screenshot shows the Wireshark network protocol analyzer interface. Red boxes and arrows highlight specific areas:

- главное меню и панель инструментов**: Points to the menu bar (File, Edit, View, etc.) and the toolbar.
- панель фильтров**: Points to the display filter field, which currently contains "HTTP POST contains 'password'".
- Список захваченных пакетов**: Points to the packet list pane, which displays a table of captured packets.
- Панель деталей пакета**: Points to the packet details pane, showing the structure of the selected packet (Frame 89).
- Панель байтов пакета**: Points to the packet bytes pane, showing the raw hexadecimal and ASCII data of the selected packet.

Список захваченных пакетов

No.	Time	Source	Destination	Protocol	Length	Info
328	56.663059	192.168.1.101	62.84.113.205	TCP	54	63705 → 443 [ACK] Seq=1 Ack=1141 Win=509 Len=0
329	57.643289	142.250.74.170	192.168.1.101	UDP	122	443 → 62539 Len=80
330	57.677456	192.168.1.101	142.250.74.170	UDP	75	62539 → 443 Len=33
331	58.111079	192.168.1.1	62.84.113.205	TCP	54	64700 → 443 [ACK] Seq=1 Ack=1 Win=510 Len=1
332	58.177139	5.255.255.255	192.168.1.101	TCP	54	443 → 64700 [ACK] Seq=1 Ack=2 Win=166 Len=0
333	59.087123	192.168.1.1	62.84.113.205	TCP	54	63223 → 443 [ACK] Seq=0 Ack=2 Win=513 Len=1
334	59.266661	52.70.125.1	192.168.1.101	TCP	54	443 → 63223 [ACK] Seq=2 Ack=1 Win=8 Len=0
335	59.619174	62.84.113.205	192.168.1.101	TLSv1.2	114	Application Data
336	59.661297	192.168.1.101	62.84.113.205	TCP	54	63705 → 443 [ACK] Seq=1 Ack=1201 Win=509 Len=0
337	60.236646	168.119.58.253	192.168.1.101	TLSv1.2	82	Application Data
338	60.237290	192.168.1.101	168.119.58.253	TLSv1.2	86	Application Data
339	60.336240	168.119.58.253	192.168.1.101	TCP	60	8443 → 64624 [ACK] Seq=113 Ack=129 Win=501 Len=0

Панель деталей пакета

```
> Frame 89: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface \Device\NPF_{05AA052B-313E-49A4-8D93-F67B92393DCD}, id 0
> Ethernet II, Src: CigShang_0e:60:30 (e4:8e:10:0e:60:30), Dst: CompalIn_30:28:6c (20:1a:06:30:28:6c)
> Internet Protocol Version 4, Src: 79.132.103.35, Dst: 192.168.1.101
> Internet Control Message Protocol
```

Панель байтов пакета

```
0000  20 1a 06 30 28 6c e4 8e 10 0e 60 30 08 00 45 20  ..0(1..  ..0..E
0010  00 44 41 9f 00 00 37 01 c9 45 4f 84 67 23 c0 a8  .DA...7.  .EO.g#..
0020  01 65 03 0a 75 c5 00 00 00 00 45 00 00 28 bd e8  .e..u...  .E..(..
0030  40 00 37 06 0d 33 c0 a8 01 65 4f 84 67 23 fc c0  @.7..3...  .eO.g#..
0040  01 bb 85 21 5d e8 f0 1b d7 4b 50 10 f8 04 96 2d  ...!]...  .KP.....
0050  00 00
```

3. Захват трафика с помощью Wireshark.

- 1. Запустите приложение Wireshark** на вашем компьютере.
- 2. Выбор сетевого интерфейса.** После запуска Wireshark откроется стартовое окно. В этом окне вы увидите список доступных сетевых интерфейсов. Выберите интерфейс, через который хотите осуществить захват трафика (например, Ethernet, Wi-Fi и т. д.).
- 3. Начало захвата.** После выбора сетевого интерфейса нажмите кнопку "Start" или "Capture" (в зависимости от версии Wireshark), чтобы начать захват трафика. Wireshark начнет запись пакетов, проходящих через выбранный сетевой интерфейс.
- 4. Мониторинг трафика.** После запуска захвата трафика вы увидите список захваченных пакетов в главном окне Wireshark. Здесь вы можете анализировать каждый пакет, просматривая его заголовок, содержимое и другие атрибуты.
- 5. Остановка захвата.** Чтобы остановить захват трафика, нажмите кнопку "Stop" или "Capture" в верхней панели инструментов. Это остановит запись новых пакетов, но сохранит уже захваченные.
- 6. Сохранение данных** (по желанию). Если вам нужно сохранить захваченные данные для дальнейшего анализа, выберите пункт меню "File" (Файл) -> "Save" (Сохранить) или "Save As" (Сохранить как), и выберите формат файла и место сохранения.

4. Анализ трафика с помощью Wireshark.

1. Применение фильтров. Вы можете использовать фильтры Wireshark, чтобы сосредоточиться на конкретных типах пакетов или протоколах. Например, вы можете применить фильтр "http" для отображения только HTTP-запросов и ответов.
2. Исследование статистики. Wireshark предоставляет различные статистические данные о перехваченном трафике, такие как статистика протоколов, временные диаграммы, распределение пакетов и другие. Используйте эти данные для получения дополнительной информации о сетевом трафике.
3. Изучение сессий. Выберите пакет: Щелкните правой кнопкой мыши на найденном пакете, чтобы открыть контекстное меню. Выберите "Follow": В контекстном меню выберите опцию "Follow TCP Stream" или "Follow UDP Stream", в зависимости от используемого протокола. Это откроет новое окно с представлением всей сессии для выбранного протокола.
4. Изучение файлов сессии. в Wireshark можно воспользоваться функцией "Export Objects", которая позволяет извлекать файлы, переданные в сети в рамках выбранной сессии.

Фильтрация трафика.

Wireshark предоставляет мощные возможности фильтрации для упрощения анализа трафика. Фильтры позволяют выбирать и отображать только нужные пакеты и информацию, исключая ненужные данные.

Вот некоторые из наиболее распространенных фильтров отображения, которые могут быть использованы в Wireshark:

1. Фильтры по адресу: Вы можете фильтровать пакеты на основе адреса источника или назначения. Например, вы можете использовать фильтр "`ip.src == 192.168.0.1`" для отображения только пакетов, исходящих от указанного IP-адреса.

2. Фильтры по протоколу: Вы можете фильтровать пакеты на основе протокола. Например, фильтр "`http`" отобразит только пакеты, относящиеся к протоколу HTTP.

3. Фильтры по содержимому: Вы можете фильтровать пакеты на основе содержимого полей или данных. Например, фильтр "`tcp.port == 80`" отобразит только пакеты с TCP-портом 80 (обычно используемым для HTTP).

4. Фильтры по типу пакета: Вы можете фильтровать пакеты на основе их типа или состояния. Например, фильтр `"tcp.flags.syn == 1"` отобразит только пакеты с установленным флагом SYN в TCP.

5. Фильтры по времени: Вы можете фильтровать пакеты на основе времени захвата. Например, вы можете использовать фильтр `"frame.time >= \"2021-01-01 00:00:00\""` для отображения пакетов, захваченных после указанной даты и времени.

6. Комбинированные фильтры: Вы также можете комбинировать несколько фильтров для более точной фильтрации. Например, вы можете использовать фильтр `"ip.src == 192.168.0.1 && tcp.port == 80"` для отображения только HTTP пакетов, исходящих от указанного IP-адреса

Пример использования фильтра

*Ethernet

Файл Редактирование Просмотр Запуск Захват Анализ Статистика Телефония Беспроводной Инструменты Помощь

http + HTTP POST contains "password"

No.	Time	Source	Destination	Protocol	Length	Info
16265	1347.390365	192.168.1.101	91.105.192.100	HTTP	94	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16309	1349.419603	192.168.1.101	91.105.192.100	HTTP	94	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16364	1355.717952	192.168.1.101	149.154.167.151	HTTP	306	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16411	1365.244198	91.105.192.100	192.168.1.101	HTTP	288	HTTP/1.1 200 OK
16414	1365.314845	192.168.1.101	91.105.192.100	HTTP	158	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16503	1388.605151	192.168.1.101	149.154.167.151	HTTP	242	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16529	1390.383316	91.105.192.100	192.168.1.101	HTTP	288	HTTP/1.1 200 OK
16532	1390.454827	192.168.1.101	91.105.192.100	HTTP	222	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16584	1415.523338	91.105.192.100	192.168.1.101	HTTP	288	HTTP/1.1 200 OK
16587	1415.595108	192.168.1.101	91.105.192.100	HTTP	334	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16605	1420.192817	192.168.1.101	149.154.167.151	HTTP	298	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16648	1432.417880	192.168.1.101	91.105.192.100	HTTP	306	POST /api HTTP/1.1 (application/x-www-form-urlencoded)
16682	1440.663220	91.105.192.100	192.168.1.101	HTTP	288	HTTP/1.1 200 OK
16685	1440.734322	192.168.1.101	91.105.192.100	HTTP	174	POST /api HTTP/1.1 (application/x-www-form-urlencoded)

> Frame 308: 288 bytes on wire (2304 bits), 288 bytes captured (2304 bits) on interface \Device\NPF_{05AA052B-313E-49A4-8D93-F67B92393DCD}, id
> Ethernet II, Src: CigShang_0e:60:30 (e4:8e:10:0e:60:30), Dst: CompalIn_30:28:6c (20:1a:06:30:28:6c)
> Internet Protocol Version 4, Src: 91.105.192.100, Dst: 192.168.1.101
> Transmission Control Protocol, Src Port: 80, Dst Port: 50341, Seq: 1, Ack: 1, Len: 234
> Hypertext Transfer Protocol
> Data (88 bytes)

0000 20 1a 06 30 28 6c e4 8e 10 0e 60 30 08 00 45 20 ..0(1.. ..`0..E
0010 01 12 e6 62 40 00 38 06 7d 88 5b 69 c0 64 c0 a8 ...b@.8. }.[i.d..
0020 01 65 00 50 c4 a5 72 f9 f3 62 c4 32 6d 38 50 18 ..P... ..h.2m8P..

Примеры фильтров:

1. Фильтр по IP-адресу источника: `ip.src == <IP-адрес>`. Например, `ip.src == 192.168.0.1` отобразит только пакеты, исходящие от указанного IP-адреса.
2. Фильтр по IP-адресу назначения: `ip.dst == <IP-адрес>`. Например, `ip.dst == 192.168.0.1` отобразит только пакеты, адресованные указанному IP-адресу.
3. Фильтр по протоколу: `proto == <протокол>`. Например, `proto == http` отобразит только пакеты, относящиеся к протоколу HTTP.
4. Фильтр по порту источника: `tcp.srcport == <порт>`. Например, `tcp.srcport == 80` отобразит только пакеты с TCP-портом 80.
5. Фильтр по порту назначения: `tcp.dstport == <порт>`. Например, `tcp.dstport == 443` отобразит только пакеты с TCP-портом 443.
6. Фильтр по содержимому данных: `data contains "<строка>"`. Например, `data contains "password"` отобразит только пакеты, содержащие строку "password" в поле данных.
7. Фильтр по типу пакета: `http, tcp, udp, icmp` и т. д. Можно использовать для отображения только пакетов, относящихся к указанному типу.
8. Фильтр по размеру пакета: `frame.len == <размер>`. Например, `frame.len > 1000` отобразит только пакеты размером больше 1000 байт.
9. Фильтр по времени захвата: `frame.time >= "<дата и время>"`. Например, `frame.time >= "2021-01-01 00:00:00"` отобразит только пакеты, захваченные после указанной даты и времени.
10. Фильтр по методу используемому в протоколе HTTP. `http.request.method==POST`
11. Комбинированные фильтры: Можно комбинировать несколько условий с помощью операторов логического И (`&&`) и логического ИЛИ (`||`). Например, `ip.src == 192.168.0.1 && tcp.dstport == 80` отобразит только пакеты, исходящие от указанного IP-адреса и имеющие TCP-порт 80.

Примеры фильтров:

1. Фильтр по IP-адресу источника: `ip.src == <IP-адрес>`. Например, `ip.src == 192.168.0.1` отобразит только пакеты, исходящие от указанного IP-адреса.
2. Фильтр по IP-адресу назначения: `ip.dst == <IP-адрес>`. Например, `ip.dst == 192.168.0.1` отобразит только пакеты, адресованные указанному IP-адресу.
3. Фильтр по протоколу: `proto == <протокол>`. Например, `proto == http` отобразит только пакеты, относящиеся к протоколу HTTP.
4. Фильтр по порту источника: `tcp.srcport == <порт>`. Например, `tcp.srcport == 80` отобразит только пакеты с TCP-портом 80.
5. Фильтр по порту назначения: `tcp.dstport == <порт>`. Например, `tcp.dstport == 443` отобразит только пакеты с TCP-портом 443.
6. Фильтр по содержимому данных: `data contains "<строка>"`. Например, `data contains "password"` отобразит только пакеты, содержащие строку "password" в поле данных.
7. Фильтр по типу пакета: `http`, `tcp`, `udp`, `icmp` и т. д. Можно использовать для отображения только пакетов, относящихся к указанному типу.
8. Фильтр по размеру пакета: `frame.len == <размер>`. Например, `frame.len > 1000` отобразит только пакеты размером больше 1000 байт.
9. Фильтр по времени захвата: `frame.time >= "<дата и время>"`. Например, `frame.time >= "2021-01-01 00:00:00"` отобразит только пакеты, захваченные после указанной даты и времени.
10. Фильтр по методу используемому в протоколе HTTP. `http.request.method==POST`
11. Комбинированные фильтры: Можно комбинировать несколько условий с помощью операторов логического И (`&&`) и логического ИЛИ (`||`). Например, `ip.src == 192.168.0.1 && tcp.dstport == 80` отобразит только пакеты, исходящие от указанного IP-адреса и имеющие TCP-порт 80.

Исследование статистики.

В Wireshark вы можете изучать различные статистические данные о захваченном сетевом трафике, например:

1. Статистика использования протоколов: Wireshark позволяет просматривать статистику использования различных сетевых протоколов. Вы можете узнать, какие протоколы наиболее активно используются в вашей сети, и проанализировать их распределение.
2. Статистика пакетов: Вы можете получить информацию о количестве захваченных пакетов, а также о типах пакетов (входящие, исходящие), размерах пакетов и других характеристиках.
3. Статистика времени: Wireshark позволяет анализировать временные характеристики сетевого трафика, такие как интервалы между пакетами, время ответа на запросы и другие временные параметры.
4. Статистика IP и MAC-адресов: Вы можете просматривать статистику использования IP-адресов и MAC-адресов в сети. Это позволяет выявить наиболее активные узлы сети и обнаружить возможные аномалии.
5. Статистика фильтров: Wireshark предоставляет возможность создавать и применять фильтры для анализа определенных аспектов сетевого трафика. Вы можете изучать статистику фильтров, чтобы понять, какие типы данных или событий наиболее часто встречаются в вашей сети.
6. Графики и диаграммы: Wireshark предоставляет графическое представление статистических данных в виде графиков и диаграмм, что упрощает визуализацию и анализ информации.

Статистика иерархии протоколов.

Wireshark · Статистика Иерархии Протоколов · 111.pcapng

Протокол	Процент Пакетов	Пакеты	Процент Байтов	Байты	Бит/с	Конт.
▼ Ethernet	100.0	7009	1.9	98126	4328	0
▼ Internet Protocol Version 6	0.1	5	0.0	200	8	0
Internet Control Message Protocol v6	0.1	5	0.0	176	7	5
▼ Internet Protocol Version 4	99.1	6946	2.7	138964	6130	0
▼ User Datagram Protocol	3.8	266	0.0	2128	93	0
Simple Service Discovery Protocol	0.2	13	0.0	2219	97	13
▼ QUIC IETF	2.9	204	1.6	82770	3651	183
Malformed Packet	0.0	1	0.0	0	0	1
▼ NetBIOS Datagram Service	0.0	1	0.0	201	8	0
▼ SMB (Server Message Block Protocol)	0.0	1	0.0	119	5	0
▼ SMB MailSlot Protocol	0.0	1	0.0	25	1	0
Microsoft Windows Browser Protocol	0.0	1	0.0	33	1	1
Domain Name System	1.0	68	0.1	4627	204	68
▼ Transmission Control Protocol	95.1	6669	93.6	4838118	213 k	465
Transport Layer Security	27.5	1925	65.9	3405429	150 k	189
Malformed Packet	0.3	20	0.0	0	0	20
▼ Hypertext Transfer Protocol	0.7	48	0.5	23659	1043	6
HTML Form URL Encoded	0.3	23	0.1	5174	228	23
Data	1.0	71	0.8	39672	1750	71
Internet Group Management Protocol	0.2	11	0.0	168	7	11
Data	0.5	32	0.0	2560	112	32
Address Resolution Protocol	0.4	26	0.0	1088	47	26

Нет дисплейного фильтра.

Заккрыть Копировать Справка

Статистика. Все IP-адреса.

Wireshark · All Addresses · good.pcapng

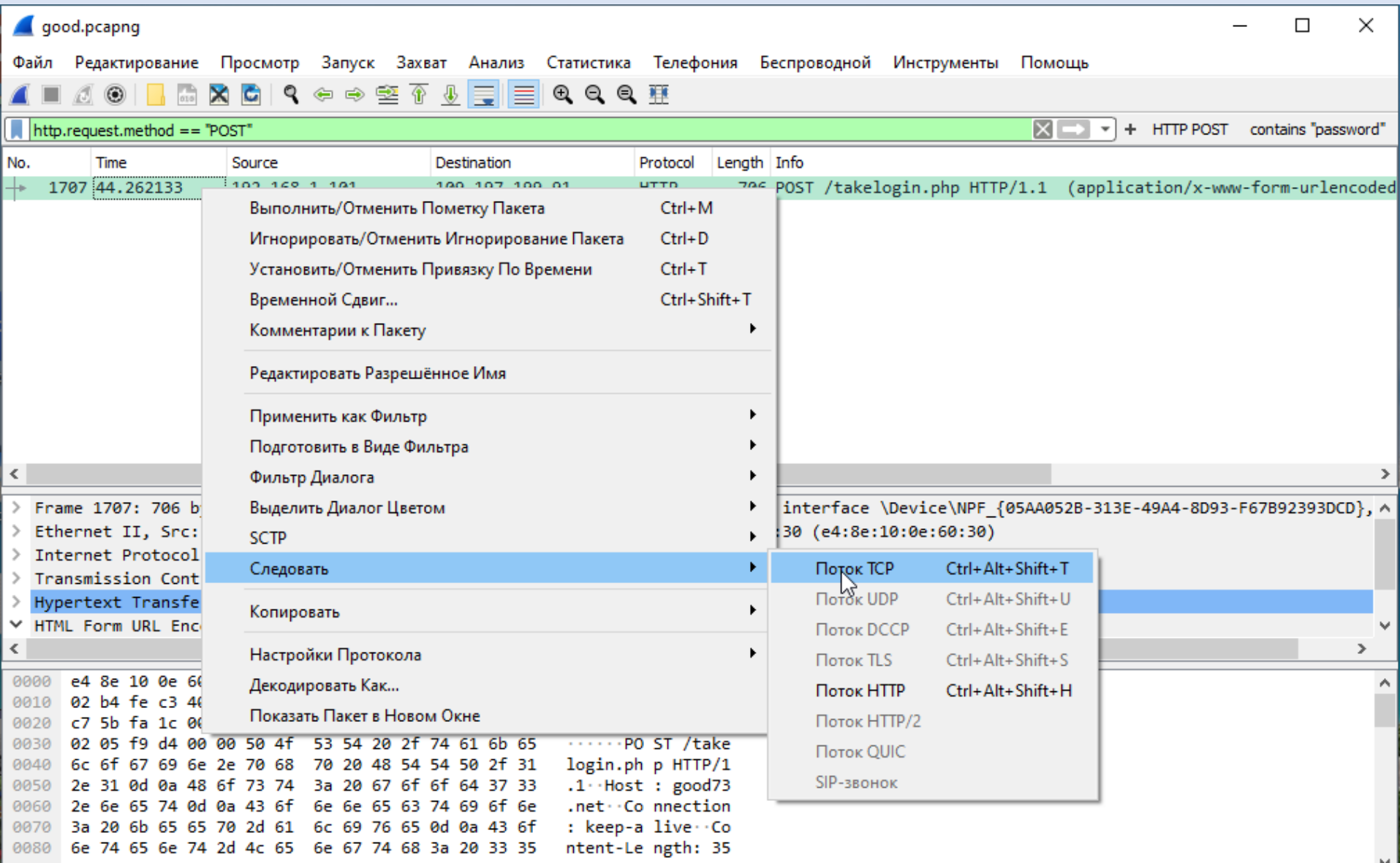
Topic / Item	Count	Average	Min Val	Max Val	Rate (ms)	Percent	Burst Rate	Burst Start
▼ All Addresses	1783				0.0362	100%	3.7400	24.530
54.77.143.119	2				0.0000	0.11%	0.0200	32.785
44.240.32.88	142				0.0029	7.96%	0.1300	36.557
44.236.158.41	39				0.0008	2.19%	0.1500	38.529
44.228.140.135	85				0.0017	4.77%	0.0800	38.432
35.83.55.181	41				0.0008	2.30%	0.1600	48.757
239.255.255.250	7				0.0001	0.39%	0.0100	6.160
213.180.199.9	39				0.0008	2.19%	0.2700	24.043
20.212.31.164	8				0.0002	0.45%	0.0300	0.000
20.199.120.182	3				0.0001	0.17%	0.0200	35.706
192.168.1.210	7				0.0001	0.39%	0.0100	6.160
192.168.1.101	1776				0.0360	99.61%	3.7400	24.530
192.168.1.1	37				0.0008	2.08%	0.0800	23.873
168.119.58.253	2				0.0000	0.11%	0.0200	14.704
162.159.134.234	6				0.0001	0.34%	0.0200	9.860
146.75.118.137	24				0.0005	1.35%	0.1100	36.429
109.197.199.91	1348				0.0273	75.60%	3.7400	24.530

Дисплейный фильтр:

Применить

Копировать Сохранить как... Заккрыть

Следовать за пакетом, для анализа всей сессии.



Объекты, полученные за время сессии.

good.pcapng

Файл Редактирование Просмотр Запуск Захват Анализ Статистика Телефония Беспроводной Инструменты

Открыть Ctrl+O

Открыть Недавние

Объединить...

Импортировать из Шестнадцатеричного Дампа...

Заккрыть Ctrl+W

Сохранить Ctrl+S

Сохранить Как... Ctrl+Shift+S

Набор Файлов

Экспортировать Указанные Пакеты...

Экспорт Результатов Анализа Пакетов

Экспортировать Байты Пакета... Ctrl+Shift+X

Экспортировать PDUхи в Файл...

Экспортировать Ключи Сессии TLS...

Экспортировать Объекты

Печать... Ctrl+P

Выход Ctrl+Q

Protocol Length Info

1	TCP	54	64028 → 80 [ACK] Seq=3439
1	TCP	1494	80 → 64028 [ACK] Seq=1629
1	HTTP	277	HTTP/1.1 200 OK (image/j
1	TCP	54	64028 → 80 [ACK] Seq=3439
1	HTTP	562	GET /pic/disabled.gif HTT
1	HTTP	329	HTTP/1.1 304 Not Modified
1	HTTP	598	GET /themes/TBDev/images/
1	HTTP	329	HTTP/1.1 304 Not Modified
1	TCP	54	64028 → 80 [ACK] Seq=4491
1	HTTP	494	GET /pic/smilies/laugh.gi
1	HTTP	1202	HTTP/1.1 200 OK (GIF89a)
1	TCP	54	64028 → 80 [ACK] Seq=4931

ntured (5648 bits) on interface \Device\NPF_{0...g_0e:60:30 (e4:8e:10:0e:60:30)

31, Ack: 166296, Len: 652

DICOM...

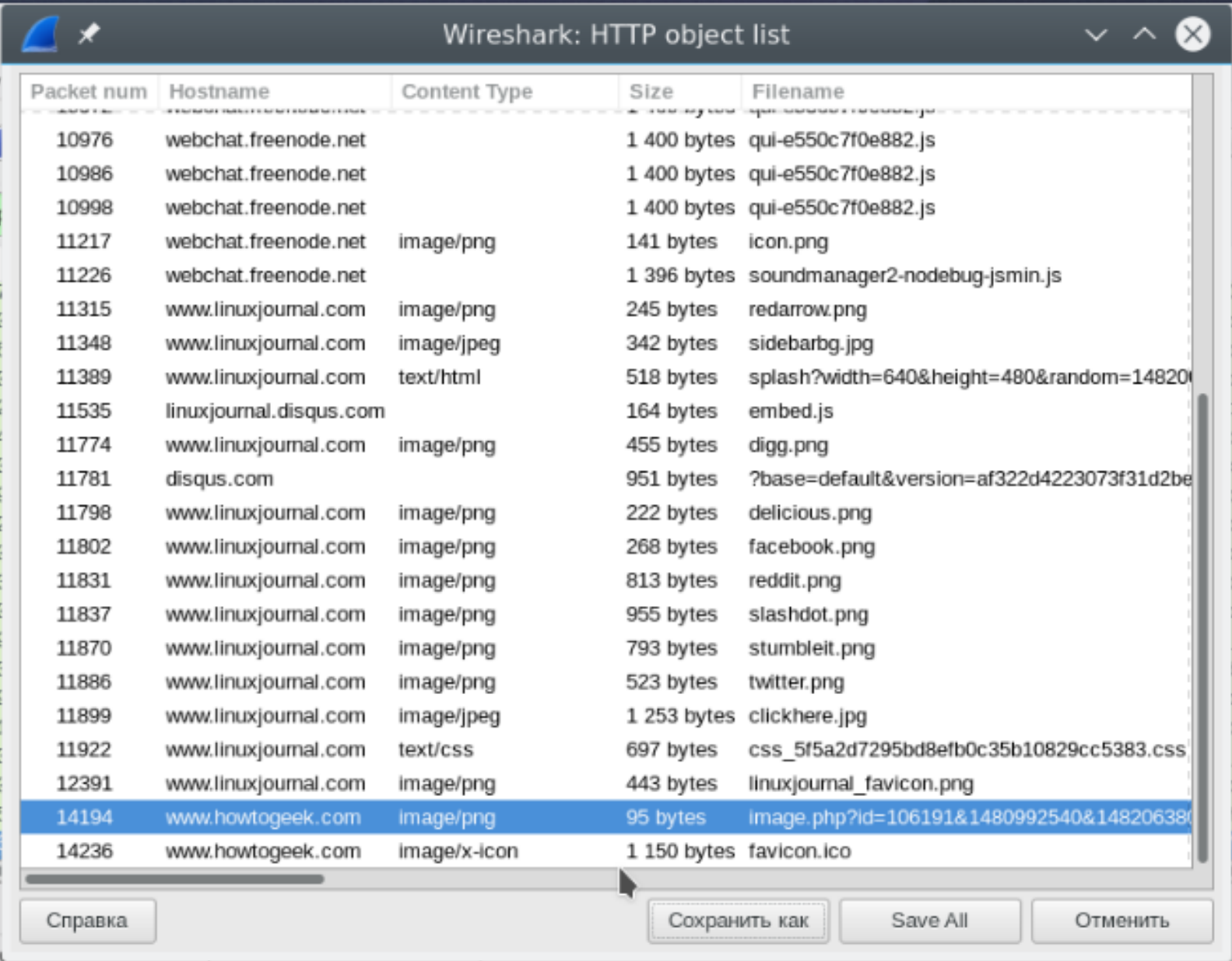
HTTP...

IMF...

SMB...

TFTP...

Объекты, полученные за время сессии.



The image shows a screenshot of the Wireshark network protocol analyzer. The main window displays the 'HTTP object list' pane, which lists objects downloaded during a session. The list includes columns for Packet num, Hostname, Content Type, Size, and Filename. The objects are sorted by packet number. The selected object is packet 14194, which is an image/png from www.howtogeek.com.

Packet num	Hostname	Content Type	Size	Filename
10976	webchat.freenode.net		1 400 bytes	qui-e550c7f0e882.js
10986	webchat.freenode.net		1 400 bytes	qui-e550c7f0e882.js
10998	webchat.freenode.net		1 400 bytes	qui-e550c7f0e882.js
11217	webchat.freenode.net	image/png	141 bytes	icon.png
11226	webchat.freenode.net		1 396 bytes	soundmanager2-nodebug-jsmin.js
11315	www.linuxjournal.com	image/png	245 bytes	redarrow.png
11348	www.linuxjournal.com	image/jpeg	342 bytes	sidebarbg.jpg
11389	www.linuxjournal.com	text/html	518 bytes	splash?width=640&height=480&random=14820
11535	linuxjournal.disqus.com		164 bytes	embed.js
11774	www.linuxjournal.com	image/png	455 bytes	digg.png
11781	disqus.com		951 bytes	?base=default&version=af322d4223073f31d2be
11798	www.linuxjournal.com	image/png	222 bytes	delicious.png
11802	www.linuxjournal.com	image/png	268 bytes	facebook.png
11831	www.linuxjournal.com	image/png	813 bytes	reddit.png
11837	www.linuxjournal.com	image/png	955 bytes	slashdot.png
11870	www.linuxjournal.com	image/png	793 bytes	stumbleit.png
11886	www.linuxjournal.com	image/png	523 bytes	twitter.png
11899	www.linuxjournal.com	image/jpeg	1 253 bytes	clickhere.jpg
11922	www.linuxjournal.com	text/css	697 bytes	css_5f5a2d7295bd8efb0c35b10829cc5383.css
12391	www.linuxjournal.com	image/png	443 bytes	linuxjournal_favicon.png
14194	www.howtogeek.com	image/png	95 bytes	image.php?id=106191&1480992540&148206380
14236	www.howtogeek.com	image/x-icon	1 150 bytes	favicon.ico

Frame 14194
Ethernet II
Internet Protocol
Transmission Control Protocol Src Port: 80 Dst Port: 56130 Seq: 813 Ack: 1743 Len: 102

Справка Сохранить как Save All Отменить

Список MIME-типов (**Internet Media Types**):

Application

application/json: JavaScript Object Notation JSON

application/javascript: JavaScript

application/octet-stream: двоичный файл без указания формата (нераспознанные двоичные данные)

application/pdf: Portable Document Format, PDF

application/soap+xml: SOAP

application/font-woff: Web Open Font Format

application/zip: ZIP

application/gzip: Gzipapplication/x-bittorrent : BitTorrent

application/xml: XMLapplication/msword: DOC

application/x-yaml: YAML

Message

message/httpmessage/imdn+xml: IMDN

message/partial: E-mail

message/rfc822: E-mail; EML-файлы, MIME-файлы, MHT-файлы, MHTML-файлы

Text

text/cmd: команды text/css: Cascading Style Sheets

text/csv: CSV text/html: HTML

text/javascript (Obsolete): JavaScript (RFC 4329)

text/plain: текстовые данные

text/php: Скрипт языка PHP text/xml: Extensible Markup Language

text/markdown: файл языка разметки Markdown

text/cache-manifest: файл манифеста

5. tcpdump

tcpdump - это утилита командной строки в Linux для захвата и анализа сетевого трафика. Она позволяет в режиме реального времени просматривать и записывать пакеты, проходящие через сетевой интерфейс.

Основные функции tcpdump

Захват пакетов: tcpdump может захватывать пакеты с различных сетевых интерфейсов, включая Ethernet, WiFi и даже loopback. Это делает его полезным инструментом для отладки сетевых проблем и мониторинга трафика.

Фильтрация трафика: С помощью мощного синтаксиса фильтров можно ограничивать захват только тем трафиком, который вас интересует. Например, можно захватывать только трафик, идущий на определенный IP-адрес или порт.

Анализ трафика: tcpdump предоставляет детальную информацию о каждом пакете, включая заголовки протоколов на разных уровнях. Это может помочь понять поведение сети и выявить потенциальные проблемы.

Сохранение захваченных данных: tcpdump может сохранять захваченные пакеты в файл для последующего анализа с помощью того же tcpdump или других инструментов, таких как Wireshark.

Примеры использования

Установка:

```
# yum install tcpdump -y
```

```
[root@localhost ~]# yum install tcpdump -y
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: mirror.neolabs.kz
```

Захват всех пакетов на интерфейсе enp0s3:

tcpdump -l enp0s3

```
[root@localhost ~]# tcpdump -i enp0s3  
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode  
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes  
23:02:01.733289 IP6 fe80::1c79:ff97:a499:dbcf > ff02::2: ICMP6, router solicitation, length 16  
23:02:01.737825 IP localhost.localdomain.40740 > gateway.domain: 60811+ PTR? 2.0.0.0.0.0.0.0.0.  
.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.f.ip6.arpa. (90)  
23:02:01.810035 IP gateway.domain > localhost.localdomain.40740: 60811 NXDomain 0/1/0 (154)
```

Захватить пакеты на указанном порту:

```
# tcpdump -l enp0s3 port 22
```

```
[root@localhost ~]# tcpdump -i enp0s3 port 22
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
23:04:50.334207 IP 10.10.10.101.59146 > localhost.localdomain.ssh: Flags [P.], seq 2788435933:2788435997, ack 856599239, win 1022, length 64
```

Захватить пакеты с указанным исходным или назначенным IP-адресом:

\$ tcpdump -i enp0s3 dst 8.8.8.8

```
[root@localhost ~]# tcpdump -i enp0s3 dst 8.8.8.8
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
23:12:05.227359 IP localhost.localdomain > dns.google: ICMP echo request, id 1626, seq 32, length 64
23:12:06.228616 IP localhost.localdomain > dns.google: ICMP echo request, id 1626, seq 33, length 64
```

\$ tcpdump -i enp0s3 src 10.10.10.1

```
[root@localhost ~]# tcpdump -i enp0s3 src 10.10.10.1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
23:13:21.008146 ARP, Request who-has 10.10.10.119 tell gateway, length 46
23:13:21.097485 IP gateway.domain > localhost.localdomain.41168: 6108 NXDomain 0/0/0 (43)
```

Захватить пакеты с указанным протоколом:

\$ sudo tcpdump -i enp0s3 icmp

```
[root@localhost ~]# tcpdump -i enp0s3 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
23:15:50.305174 IP localhost.localdomain > dns.google: ICMP echo request, id 1631, seq 1, length 64
23:15:50.390254 IP dns.google > localhost.localdomain: ICMP echo reply, id 1631, seq 1, length 64
```

Захват пакетов, идущих на определенный IP и порт:

\$ tcpdump -i enp0s3 dst 10.10.10.1 and port 53

```
[root@localhost ~]# tcpdump -i enp0s3 dst 10.10.10.1 and port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
23:18:56.119413 IP localhost.localdomain.48659 > gateway.domain: 17386+ A? mail.ru. (25)
```

Захват пакетов, идущих на определенный IP и порт, детальный вывод:

\$ tcpdump -i enp0s3 dst 10.10.10.1 and port 53 -vvv

```
[root@localhost ~]# tcpdump -i enp0s3 dst 10.10.10.1 and port 53 -vvv
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
23:23:01.069134 IP (tos 0x0, ttl 64, id 50760, offset 0, flags [DF], proto UDP (17), length 53)
    localhost.localdomain.38696 > gateway.domain: [bad udp cksum 0x28b1 -> 0x4961!] 44831+ A? mail.ru. (25)
23:23:01.069963 IP (tos 0x0, ttl 64, id 50763, offset 0, flags [DF], proto UDP (17), length 69)
```

Захват и сохранение пакетов в файл:

\$ tcpdump -i enp0s3 -w packets.cap

Чтение пакетов из файла:

\$ tcpdump -r packets.cap

Часто используемые параметры:

-i <интерфейс>: Указывает tcpdump захватывать пакеты только на определенном сетевом интерфейсе. Например, -i eth0 указывает захватывать пакеты на интерфейсе eth0.

-n: Отключает разрешение DNS-имен и выводит IP-адреса в числовом формате. Это может быть полезно для улучшения производительности и избежания задержек из-за разрешения имен.

-c <количество>: Ограничивает количество захватываемых пакетов. Например, -c 100 ограничивает захват первых 100 пакетов и затем tcpdump завершается.

-s <размер>: Устанавливает размер захватываемого пакета в байтах. Значение по умолчанию - 65535 байт. Вы можете использовать это для ограничения размера захватываемых данных и экономии ресурсов.

-w <файл>: Сохраняет захваченные пакеты в указанный файл. Например, -w capture.pcap сохраняет пакеты в файле capture.pcap, который может быть открыт и проанализирован позже с помощью других инструментов, таких как Wireshark.

-r <файл>: Загружает пакеты из указанного файла для анализа. Например, -r capture.pcap загружает пакеты из файла capture.pcap вместо захвата в реальном времени.

-v: Выводит более подробную информацию о захваченных пакетах. Это может включать расширенные заголовки протоколов и другие детали.

-q: Устанавливает "тихий" режим, в котором tcpdump выводит меньше информации. Это может быть полезно для автоматизированных сценариев или фильтрации вывода.

Домашнее задание:

1. Изучить дополнительные материалы.