# ???? ?? ?????? ? ????????? ? KICS for Networks

# ????????? ???????? ??? ?????? ? ?????????

Данная статья рассматривает основные утилиты и методы для локального проигрывания трафика, включая подготовку данных, их воспроизведение и анализ результатов.

## tcpreplay

**tcpreplay** - основной инструмент воспроизведения, который является наиболее популярным и функциональным инструментом для воспроизведения сетевого трафика из PCAP-файлов. Утилита позволяет точно воспроизводить ранее захваченный трафик с контролем скорости передачи и различными опциями редактирования пакетов.

Базовое использование tcpreplay:

```
# Простое воспроизведение файла
tcpreplay -i eth0 traffic.pcap

# Воспроизведение с максимальной скоростью
tcpreplay -i eth0 --topspeed --preload-pcap sample.pcap

# Воспроизведение с заданной скоростью (PPS)
tcpreplay -i eth0 -p 1000 sample.pcap

# Воспроизведение с заданной пропускной способностью
tcpreplay -i eth0 --mbps=100 sample.pcap
```

где:

- -i interface - указание сетевого интерфейса для вывода
- --topspeed - воспроизведение с максимальной скоростью

- --preload-pcap - предварительная загрузка файла в память (повышает производительность)
- -p speed - задание скорости в пакетах в секунду (PPS)
- --mbps=speed - задание скорости в мегабитах в секунду
- --loop=count - количество повторений воспроизведения
- -K - загрузка всего файла в память перед отправкой
- --unique-ip - изменение IP-адресов в каждой итерации

## tcpreplay-edit

Для адаптации трафика к тестовой среде KICS for Networks используется **tcpreplay-edit**:

```
# Изменение IP-адресов
tcpreplay-edit --srcipmap=192.168.1.0/24:10.0.1.0/24 \
               --dstipmap=192.168.2.0/24:10.0.2.0/24 \
               --infile=original.pcap --outfile=modified.pcap

# Изменение портов
tcpreplay-edit --portmap=80:8080,443:8443 \
               --infile=original.pcap --outfile=modified.pcap

# Рандомизация адресов
tcpreplay-edit --seed=12345 --randomize-ips \
               --infile=original.pcap --outfile=randomized.pcap
```

## tcpdump

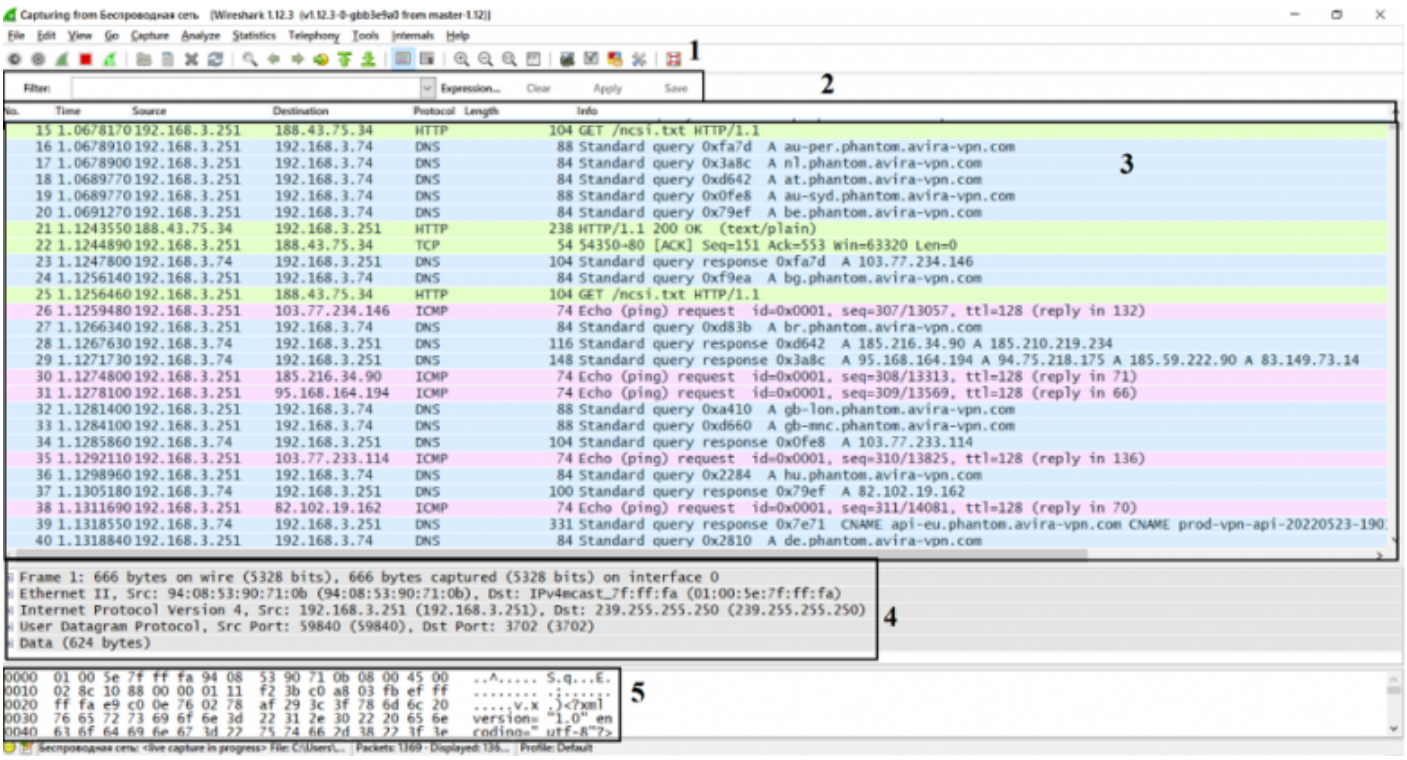
Для захвата трафика используется утилита **tcpdump**:

```
# Захват всего трафика на интерфейсе
tcpdump -i eth0 -w industrial_traffic.pcap

# Захват с фильтрацией по протоколу
tcpdump -i eth0 -w modbus_traffic.pcap 'port 502'

# Захват с ограничением размера файла
tcpdump -i eth0 -w traffic.pcap -C 100 # файлы по 100MB

# Захват с временным ограничением
timeout 300 tcpdump -i eth0 -w traffic_5min.pcap

# Modbus TCP (порт 502)
```

```
tcpdump -i eth0 'port 502' -w modbus.pcap

# DNP3 (порт 20000)
tcpdump -i eth0 'port 20000' -w dnp3.pcap

# IEC 61850 (порты 102, 843)
tcpdump -i eth0 'port 102 or port 843' -w iec61850.pcap

# PROFINET DCP
tcpdump -i eth0 'ether proto 0x8892' -w profinet.pcap

# Комбинированные фильтры
tcpdump -i eth0 '(port 502 or port 20000 or port 102) and not icmp' -w industrial.pcap
```

# ?????? ??????? ? ??????? Wireshark

Wireshark является незаменимым инструментом для анализа промышленного трафика перед его воспроизведением. При захвате трафика вы увидите что-то подобное:




Разберем более подробно это окно:

1. Панель фильтров, позволяющая найти необходимую информацию. Подробнее о неё рассказано в пятой главе руководства.
2. Панель наименований, разделяющая информацию из пункта 3 на номер, временя с начала захвата трафика, источник и адресат, а также используемый протокол,

размер пакета и небольшую информацию о сетевом пакете.
3. Панель пакетов, обновляющаяся в реальном времени. Здесь информация о пакетах разделена по столбцам, определённым на панели наименований.
4. Панель уровней, описывающая уровни модели OSI выбранного сетевого пакета.
5. Панель метаданных, представляющая данные в шестнадцатеричном коде и символах.

Вот некоторые полезные фильтры которые можно применять в поисковой строке wireshark:

```
# Modbus трафик
modbus

# DNP3 сообщения
dnp3

# IEC 61850 GOOSE
goose

# PROFINET Real-Time
pn_rt

# Ethernet/IP
enip

# Анализ ошибок в протоколах
expert.severity == "Error"

# Поиск по IP адресу отправителя
ip.src == x.x.x.x

# Поиск по IP адресу получателя
ip.dst == x.x.x.x

# Поиск по TCP порту 80
tpc.port == 80

# Также могут использоваться логические операторы, например:
И «and/&&»
ИЛИ «or/||»
НЕ «not/!»
```

Revision #3
Created 23 July 2025 09:54:50
Updated 17 November 2025 09:11:31 by Sergey Malyugin