Средства сетевой диагностики. Нет интернета. Пропал Интернет.

Для диагностики работы системы рекомендуем использовать следующие программы:

1. ping - доступность адреса в сети

2. traceroute -I <интерфейс> <узел> - трассировка до узла

3. ip addr - сетевые настройки

4. ip route - маршрутизация

5. ip rule - правила таблиц маршрутизации (PBR)

6. ip nei - просмотр таблицы arp записей

8. arping - аналог команды пинг на канальном уровне L2 (arp запросы)

9. telnet - соединение с адресом по произвольному tcp порту

10. tcpdump - перехват и анализ сетевого трафика

11. tshark - перехват и анализ сетевого трафика

12. iptables - настройки firewall, а также NAT правил

13. iperf - проверка ширины канала

14. ethtool - информация по режиму работы сетевого адаптера

Примеры:

ping

```
ping <ip>
```

При потери пакетов, а также для различной диагностики рекомендуется проверять канал и локальную сеть.

Для сети 100мб должно быть не более 1% потерь пакетов

```
ping -f -l 30 <ip>
```

Для сети 1000мб должно быть не более 2-3% потерь пакетов

```
ping -f -l 300 <ip>
```

p.s. При этом если пингуется аппаратный маршрутизатор, данные могут сильно зависеть от его загруженности и проверять лучше связь до ПК расположенного за маршрутизатором.

Проверка наличия ip-адреса(абонента) в сети (широковещательном сегменте)

```
arp -n | grep ip_
```

Проверка подмены адреса

```
arping -D -I <  > <ip>
```

Если будет получен ответ - значит в сети произошла подмена адреса сервера. Необходимо искать виновника.

Также подмену можно обнаружить с тестового ПК:

```
ping <ip>
arp
```

Будет выведена arp-таблица. Необходимо сравнить mac-адрес в этой таблице с адресом сетевой карты (локальной) сервера Carbon Billing.

Соединение с адресом <ip> и портом <port>

```
telnet <ip> <port>
```

netstat

Данная команда показывает содержимое различных структур данных, связанных с сетью, в различных форматах в зависимости от указанных опций.

Пример 1.

Задача - проверить какая служба/демон "слушает" на порту 1443 и слушают ли вообще

```
netstat -antp | grep 1443
```

Данная команда выведет информацию какая служба/демон "слушает" на порту 1443. Если не будет никакого вывода, значит данный порт никто не "слушает"

tcpdump

Для примеров примем следующие значения:

- локальный интерфейс eth0 ip 10.0.0.1/24
- внешний интерфейс eth1 ip 1.1.1.1/30 шлюз 1.1.1.2
- подсеть серых адресов абонентов 10.0.0.0/24 шлюз 10.0.0.1 dns 10.0.0.1
- тестовый абонент 10.0.0.101 и его мак-адрес 01:23:45:67:89:0a

Пример 1.

Задача - просмотреть icmp (ping) пакеты на локальный интерфейс Carbon Billing.

```
tcpdump -i eth0 -nn proto ICMP
```

Эта команда выдаст все ICMP пакеты, проходящие через локальный интерфейс.

```
tcpdump -i eth0 -nn proto ICMP and host 10.0.0.1
```

Выдает информацию о ICMP пакетах, приходящих и исходящих на локальный IP адрес.

```
tcpdump -i eth0 -nn proto ICMP and net 10.0.0.0/24
```

Выдает информацию о ICMP пакетах, приходящих и исходящих из локальной сети.

```
tcpdump -i eth0 -nn proto ICMP and not host 10.0.0.1
```

Выдаст все ICMP пакеты, проходящие через локальный интерфейс, кроме пакетов, которые относятся к хосту 10.0.0.1

Пример 2.
Нам необходимо увидеть, кто в локальной сети создает большое количество новых сессий.

```
tcpdump -i eth0 -nn tcp[13] == 2
```

Команда выдаст все TCP пакеты с флагом SYN (начало сессии).

```
tcpdump -i eth0 -nn tcp[13] == 2 and src net 10.0.0.0/24
```

Выдаст все SYN пакеты, где ip источника будет ip-адрес локальной сети.

Пример 3.
Нам необходимо проверить, что от пользователя приходят запросы к DNS серверу.

```
tcpdump -i eth0 -nn port 53 and host 10.0.0.101
```

Команда выдает все dns запросы и ответы для заданного адреса.

Пример 4.
Анализ ARP пакетов в локальной сети.

```
tcpdump arp -i eth0 -nn
```

Команда выдает все ARP пакеты на интерфейсе.

```
tcpdump arp -i eth0 -nn -e | grep 01:23:45:67:89:0a
```

Команда выдает все arp запросы тестового абонента с заданным MAC-адресом.

Пример 5.

Проверить работу dhcp relay

```
tcpdump -nv -i eth1 'udp and port 67 and port 68'
```

Пример 6.

Проверить работу PPPoE, нет ли посторонних PPPoE-серверов в сети

```
tcpdump -e -i eth1 -nv 'ether proto 0x8863 or ether proto 0x8864'
```

Пример 7.
Посмотреть входящие/исходящие пакеты на порту 1443

```
tcpdump -nvi any port 1443
```

Зная определенный хост, например отправителя 1.1.1.1

```
tcpdump -nvi any port 1443 and host 1.1.1.1
```

Для более детального анализа различных сетевых протоколов есть очень удобная утилита Wireshark. Программа бесплатна и может быть установлена как на ОС Windows, так и на ОС Linux. Чтобы провести анализ сетевого трафика с помощью этой программы, нужно сделать на сервере дамп трафика за интересующий интервал времени и с нужными фильтрами:

```
tcpdump arp -i eth0 -nn -s 0 -w /root/eth0_dump
```

Пакеты, проходящие через локальный интерфейс, целиком запишутся в файл /root/eth0_dump. После этого нужно скачать этот файл на локальную машину и с помощью программы Wireshark провести детальный анализ.

Пример 8. Получить от DNS сервера запись

```
dig vk.com  @8.8.8.8
```

Пример 9. Найти кто-шлет broadcast в сети:

Команда выведет пакеты с указанием мак-адресов посылающих широковещательные запросы.

```
tcpdump -nveei <interface> host 255.255.255.255
```

Ethtool

Пример 1. Общая информация по сетевому контроллеру

```
[root@www-example-com root]# ethtool eth1
Settings for Leth1:
 Supported ports: [ TP ]
 Supported link modes:   10baseT/Half 10baseT/Full
                         100baseT/Half 100baseT/Full
                         1000baseT/Full
 Supports auto-negotiation: Yes
 Advertised link modes:  10baseT/Half 10baseT/Full
                         100baseT/Half 100baseT/Full
                         1000baseT/Full
 Advertised auto-negotiation: Yes
 Speed: 1000Mb/s
 Duplex: Full
 Port: Twisted Pair
 PHYAD: 1
 Transceiver: internal
 Auto-negotiation: on
 Supports Wake-on: umbg
 Wake-on: g
 Current message level: 0x00000007 (7)
 Link detected: yes
```

Пример 2. Для отладки различных ситуаций с проблемами rx/tx нужна подробная статистика, она выводится с помощью следующей команды:

```
[root@www-example-com root]$ ethtool -S eth1
NIC statistics:
     rx_packets: 9244765078
     tx_packets: 12132738524
     rx_bytes: 2844075321346
     tx_bytes: 12330719323485
     rx_broadcast: 3982759
     tx_broadcast: 487538
     rx_multicast: 2972
     tx_multicast: 0
     rx_errors: 18
     tx_errors: 0
     tx_dropped: 0
     multicast: 2972
     collisions: 0
     rx_length_errors: 0
     rx_over_errors: 0
     rx_crc_errors: 9
     rx_frame_errors: 0
     rx_no_buffer_count: 3
     rx_missed_errors: 2475
     tx_aborted_errors: 0
     tx_carrier_errors: 0
     tx_fifo_errors: 0
     tx_heartbeat_errors: 0
     tx_window_errors: 0
     tx_abort_late_coll: 0
     tx_deferred_ok: 0
     tx_single_coll_ok: 0
     tx_multi_coll_ok: 0
     tx_timeout_count: 0
     tx_restart_queue: 409
     rx_long_length_errors: 0
     rx_short_length_errors: 0
     rx_align_errors: 0
     tx_tcp_seg_good: 0
     tx_tcp_seg_failed: 0
     rx_flow_control_xon: 0
     rx_flow_control_xoff: 0
     tx_flow_control_xon: 0
     tx_flow_control_xoff: 0
     rx_long_byte_count: 2844075321346
     rx_csum_offload_good: 8824741871
     rx_csum_offload_errors: 17871
     alloc_rx_buff_failed: 0
     tx_smbus: 0
     rx_smbus: 0
     dropped_smbus: 0
```

iperf

Пример 1.
Чтобы проверить ширину канала утилитой, с одной стороны запускаем ее в режиме сервера:

```
iperf -s
------------------------------------------------------------
Server listening on TCP port 5001
TCP window size: 85.3 KByte (default)
------------------------------------------------------------
```

На второй машине в режиме клиента с указанием адреса ПК, на котором включен режим сервера.

```
iperf -c 192.168.1.1
------------------------------------------------------------
Client connecting to 192.168.1.1, TCP port 5001
TCP window size: 49.7 KByte (default)
------------------------------------------------------------
[  3] local 192.168.1.2 port 54593 connected with 192.168.1.1 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-10.0 sec  95.31 GBytes  96.28 Gbits/sec
```

Аналогичный вывод будет на стороне сервера.