

- Netflow
 -
 -
- sFlow
 -
 -
- SNMP
 -
 -
- Zabbix-agent/proxy
 -
 - zabbix-agent
 - zabbix-server
- Syslog
 -
 -
- -
 -
- -
 -

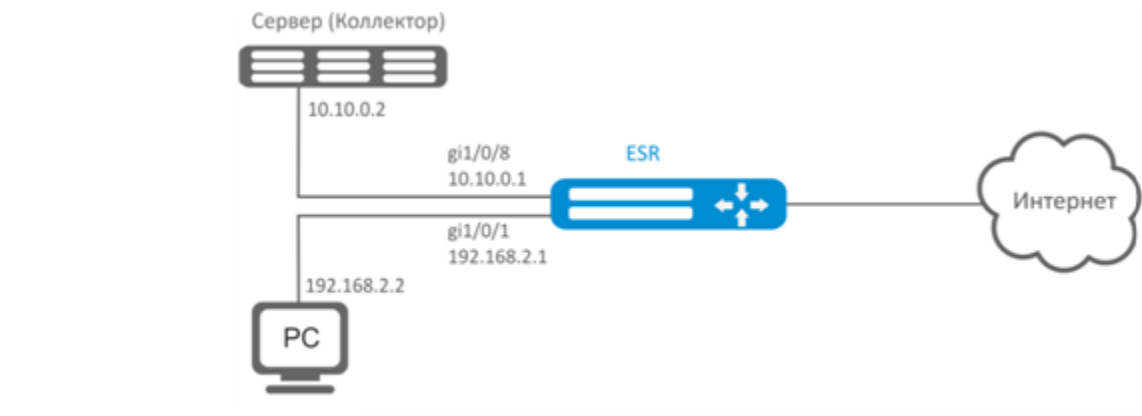
Netflow

Netflow — , . Netflow (, , .) () . .

1	Netflow-.	esr(config)# netflow version <VERSION>	<VERSION> – Netflow-: 5, 9 10.
2	.	esr(config)# netflow max-flows <COUNT>	<COUNT> – , [10000..2000000]. : 512000.
3	, .	esr(config)# netflow inactive-timeout <TIMEOUT>	<TIMEOUT> – , , [0..240]. : 15 .
4	Netflow-.	esr(config)# netflow refresh-rate <RATE>	<RATE> – , , [1..10000]. : 10.
5	Netflow .	esr(config)# netflow enable	
6	Netflow .	esr(config)# netflow collector <ADDR>	<ADDR> – IP-, AAA.BBB.CCC.DDD, [0..255].
7	Netflow- .	esr(config-netflow-host)# port <PORT>	<PORT> – UDP-, [1..65535]. : 2055.
8	Netflow- // .	esr(config-if-gi)# ip netflow export	

:

gi1/0/1 gi1/0/8 .



:

:

- gi1/0/1, gi1/0/8 firewall «ip firewall disable».
- IP- .

:

IP- :

```
esr(config)# netflow collector 10.10.0.2
```

```
netflow gi1/0/1:

esr(config)# interface gigabitethernet 1/0/1
esr(config-if-gi)# ip netflow export
```

netflow :

```
sr(config)# netflow enable
```

Netflow :

```
esr# show netflow statistics
```

Netflow sFlow, sFlow.

sFlow

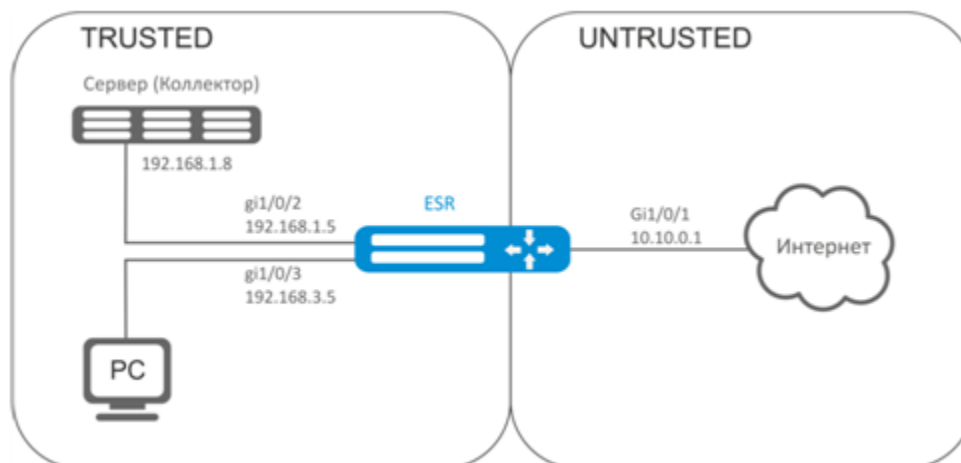
Sflow — , , .

1	sFlow-	esr(config)# sflow sampling-rate <RATE>	<RATE>- , [1..10000000]. 10 . :1000.
2	,	esr(config)# sflow poll-interval <TIMEOUT>	<TIMEOUT>- , , [1..10000]. :10 .
3	sFlow .	esr(config)# sflow enable	
4	sFlow .	esr(config)# sflow collector <ADDR>	<ADDR>- IP-, AAA.BBB.CCC.DDD, [0..255].

5	sFlow- // .	esr(config-if-gi)# ip sflow export	
---	-------------	------------------------------------	--

:

trusted untrusted.



:

ESR :

```

esr# configure
esr(config)# security zone TRUSTED
esr(config-zone)# exit
esr(config)# security zone UNTRUSTED
esr(config-zone)# exit
  
```

:

```

esr(config)# interface gi1/0/1
esr(config-if-gi)# security-zone UNTRUSTED
esr(config-if-gi)# ip address 10.10.0.1/24
esr(config-if-gi)# exit
esr(config)# interface gi1/0/2-3
esr(config-if-gi)# security-zone TRUSTED
esr(config-if-gi)# exit
esr(config)# interface gi1/0/2
esr(config-if-gi)# ip address 192.168.1.5/24
esr(config-if-gi)# exit
esr(config)# interface gi1/0/3
esr(config-if-gi)# ip address 192.168.3.5/24
esr(config-if-gi)# exit
  
```

IP- :

```

esr(config)# sflow collector 192.168.1.8
  
```

sFlow «rule1» TRUSTED-UNTRUSTED:

```

esr(config)# security zone-pair TRUSTED UNTRUSTED
esr(config-zone-pair)# rule 1
esr(config-zone-pair-rule)# action sflow-sample
esr(config-zone-pair-rule)# match protocol any
esr(config-zone-pair-rule)# match source-address any
esr(config-zone-pair-rule)# match destination-address any
esr(config-zone-pair-rule)# enable

```

sFlow :

```

sr(config)# sflow enable

```

sFlow [Netflow](#).

SNMP

SNMP (. Simple Network Management Protocol —) —, IP- TCP/UDP. SNMP , .

1	SNMP-	esr(config)# snmp-server	
2	community SNMPv2c.	esr(config)# snmp-server community <COMMUNITY> [<TYPE>] [{ <IP-ADDR> <IPV6-ADDR> }] [client-list <OBJ-GROUP- NETWORK-NAME>] [<VERSION>] [view <VIEW- NAME>] [vrf <VRF>]	<COMMUNITY> – SNMP; <TYPE> – : • ro – ; • rw – . <IP-ADDR> – IP-, , AAA.BBB.CCC.DDD, [0..255]; <IPV6-ADDR> – IPv6-, X:X:X:X::X, [0..FFFF]; <OBJ-GROUP-NETWORK-NAME> – IP-, snmp-, 31; <VERSION> – snmp, community, v1 v2c; <VIEW-NAME> – SNMP view, 31; <VRF> – VRF, , 31.
3	SNMP,	esr(config)# snmp-server contact <CONTACT>	<CONTACT> – , 255.
4	DSCP IP- SNMP-().	esr(config)# snmp-server dscp <DSCP>	<DSCP> – DSCP, [0..63]. : 63.
5	snmp-()	esr(config)# snmp-server system- shutdown	
6	C SNMPv3-.	esr(config)# snmp-server user <NAME>	<NAME> – , 31.
7	SNMP,	esr(config)# snmp-server location <LOCATION>	<LOCATION> – , 255.
8	SNMPv3.	esr(config-snmp-user)# access <TYPE>	<TYPE> – : • ro – ; • rw – .
9	SNMPv3.	esr(config-snmp-user)# authentication access <TYPE>	<TYPE> – : • auth – ; • priv – .

10	SNMPv3-	esr(config-snmp-user)# authentication algorithm <ALGORITHM>	<ALGORITHM> – : • md 5 – md5; • sha 1 – sha1.
11	SNMPv3-	esr(config-snmp-user)# authentication key ascii-text { <CLEAR-TEXT> encrypted <ENCRYPTED-TEXT> }	<CLEAR-TEXT> – , 8 16; • encrypted – : <ENCRYPTED-TEXT> – 8 16 (16 32) (0xYYYY...) (YYYY...).
12	IP-, SNMPv3 SNMPv3.	esr(config-snmp-user)# client-list <NAME>	<NAME> – object-group, 31 .
13	IPv4/IPv6-, SNMPv3-	esr(config-snmp-user)# ip address <ADDR>	<ADDR> – IP-, , AAA.BBB.CCC.DDD, [0..255].
		esr(config-snmp-user)# ipv6 address <ADDR>	<IPV6-ADDR> – IPv6-, X:X:X:X::X, [0..FFFF].
14	SNMPv3-	esr(config-snmp-user)# enable	: .
15	.	esr(config-snmp-user)# privacy algorithm <ALGORITHM>	<ALGORITHM> – : • aes 128 – AES-128; • des – DES.
16	.	esr(config-snmp-user)# privacy key ascii-text { <CLEAR-TEXT> encrypted <ENCRYPTED-TEXT> }	<CLEAR-TEXT> – , 8 16; <ENCRYPTED-TEXT> – 8 16 (16 32) (0xYYYY...) (YYYY...).
	snmp view, OID user.	esr(config-snmp-user)# view <VIEW-NAME>	<VIEW-NAME> – SNMP view, OID, 31 .
17	SNMP IP- SNMP .	esr(config)# snmp-server host { <IP-ADDR> <IPV6-ADDR> } [vrf <VRF>]	<IP-ADDR> – IP-, AAA.BBB.CCC.DDD, [0..255]. <IPV6-ADDR> – IPv6-, X:X:X:X::X, [0..FFFF]; <VRF> – VRF, SNMP-, 31 .
18	SNMP ()).	esr(config-snmp-host)# port <PORT>	<PORT> – UDP-, [1..65535]. : 162.
19	SNMP-	esr(config)# snmp-server enable traps <TYPE>	<TYPE> – . : config, entry, entry-sensor, environment, envmon, files-operations, flash, flash-operations, interfaces, links, ports, screens, snmp, syslog. .. ESR-Series . CLI . 1.11.0 .
20	snmp view, OID community (SNMPv2) user (SNMPv3).	esr(config)# snmp-server enable traps <TYPE>	<VIEW-NAME> – SNMP view, 31 .

:

SNMPv3 admin. IP- esr – 192.168.52.41, IP- – 192.168.52.8.



:

:

- gi1/0/1;
- IP- gi1/0/1.

:

SNMP-:

```
esr(config)# snmp-server
```

SNMPv3:

```
esr(config)# snmp-server user admin
```

:

```
esr(snmp-user)# authentication access priv
```

SNMPv3-:

```
esr(snmp-user)# authentication algorithm md5
```

SNMPv3-:

```
esr(snmp-user)# authentication key ascii-text 123456789
```

:

```
esr(snmp-user)# privacy algorithm aes128
```

:

```
esr(snmp-user)# privacy key ascii-text 123456789
```

SNMPv3-:

```
esr(snmp-user)# enable
```

- Trap-PDU :

```
esr(config)# snmp-server host 192.168.52.41
```

Zabbix-agent/proxy

Zabbix-agent -, , Zabbix . : . , , firewall - tcp, 10050. - tcp, 10051.

Zabbix- - , Zabbix-.

1	/proxy.	esr(config)# zabbix-agent esr(config)# zabbix-proxy	
2	() zabbix .	esr(config-zabbix)# hostname <WORD> esr(config-zabbix-proxy)# hostname <WORD>	<WORD> - , 255 .
3	zabbix .	esr(config-zabbix)# server <ADDR> esr(config-zabbix-proxy)# server <ADDR>	<ADDR> - IP- , AAA.BBB.CCC.DDD, [0..255].

4	().	esr(config-zabbix)# active-server <ADDR> <PORT> esr(config-zabbix-proxy)# active-server <ADDR> <PORT>	<ADDR> – IP-, AAA.BBB.CCC.DDD, [0..255]. <PORT> – , [1..65535]. 10051.
5	, /()	esr(config-zabbix)# port <PORT> esr(config-zabbix-proxy)# port <PORT>	<PORT> – , zabbix/, [1..65535]. :10050.
6	zabbix/().	esr(config-zabbix)# remote-commands esr(config-zabbix-proxy)# remote-commands	
7	, ().	esr(config-zabbix)# source-address <ADDR> esr(config-zabbix-proxy)# source-address <ADDR>	<ADDR> – IP-, AAA.BBB.CCC.DDD, [0..255]. : .
8	().	esr(config-zabbix)# timeout <TIME> esr(config-zabbix-proxy)# timeout <TIME>	<TIME> – , [1..30]. 3. , .. , ..
9	/	esr(config-zabbix)# enable esr(config-zabbix-proxy)# enable	
10	firewall (self) TCP 10050,10051.. Firewall		

zabbix-agent



:

.

:

zabbix , :

```
esr(config-zabbix)# server 192.168.32.101
esr(config-zabbix)# source-address 192.168.39.170
```

hostname, active-server, .

```
esr(config-zabbix)# hostname ESR-agent
esr(config-zabbix)# active-server 192.168.32.101
esr(config-zabbix)# remote-commands
```

, .

```
esr(config-zabbix)# timeout 30
esr(config-zabbix)# enable
```

zabbix-server

:

Узлы сети

Все узлы сети / TEST Активировано ZBX SNMP JMX IPMI Группы элементов данных 10 Элементы данных 94 Триггеры 15 Графики 36 Правила обнаружения 2 Веб-сценарии

Узел сети Шаблоны IPMI Макросы Инвентарные данные узла сети Шифрование

* Имя узла сети ESR-agent

Видимое имя

* Группы SLA ✕ Выбрать
начните печатать для поиска

* Должен существовать по крайней мере один интерфейс.

Интерфейсы агента	IP адрес	DNS имя	Подключаться через	Порт	По умолчанию
	192.168.39.170		IP DNS	10050	☒ Удалить

[Добавить](#)

Интерфейсы SNMP [Добавить](#)

Интерфейсы JMX [Добавить](#)

Интерфейсы IPMI [Добавить](#)

Описание

Наблюдение через прокси (без прокси) ▾

Активировано ☒

Обновить Клонировать Полное клонирование Удалить Отмена

(-> ->)

Общие
Прокси
Аутентификация
Группы пользователей
Пользователи
Способы оповещений
Скрипты
Очередь

Скрипты

Имя
ping_agent

Тип
IPMI
Скрипт

Выполнять на
Zabbix агент
Zabbix сервер (прокси)
Zabbix сервер

Команды
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[sudo ping -c 3 192.168.32.101]"

Описание

Группа пользователей
Все

Группа узлов сети
Все

Требуемые права доступа к узлам сети
Чтение
Запись

Включить подтверждение
☐

Текст подтверждения

Тест подтверждения

Обновить
Клонировать
Удалить
Отмена

ESR :

- **Ping:**

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[ sudo ping -c 3 192.168.32.101]"
```

(ESR), , ping (192.168.32.101), .

"-c" - , ping .

- **Ping VRF:**

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[sudo netns -exec -n backup sudo ping 192.168.32.101 -c 5 -W 2 ]"
```

VRF backup.

- **Fping**

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[ sudo fping 192.168.32.101]"
```

(ESR), , fping (192.168.32.101), .

- **Fping VRF**

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[sudo netns-exec -n backup sudo fping 192.168.32.101 ]"
```

- **Traceroute**

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[ sudo traceroute 192.168.32.101]"
```

(ESR), , traceroute (192.168.32.101), .

• Traceroute VRF

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[ sudo netns-exec -n backup sudo traceroute 192.168.32.179]"
```

• Iperf

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[ sudo iperf -c 192.168.32.101 -u -b 100K -i 1 -t 600]"
```

(ESR), , iperf (192.168.32.101), .

• Iperf VRF

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[ sudo netns-exec -n backup sudo iperf -c 192.168.32.101 -u -b 100K -i 1 -t 600]"
```

• Nslookup

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[sudo nslookup ya.ru ]"
```

(ESR), , nslookup , .

• Nslookup VRF

```
zabbix_get -s {HOST.CONN} -p 10050 -k "system.run[sudo netns-exec sudo nslookup ya.ru ]"
```

Iperf:

iperf_agent

```
zabbix_get -s 192.168.39.170 -p 10050 -k "system.run[ sudo iperf -c 192.168.32.101]"

-----
Client connecting to 192.168.32.101, TCP port 5001
TCP window size: 49.5 KByte (default)
-----
[ 3] local 192.168.39.170 port 52815 connected with 192.168.32.101 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3]  0.0-10.0 sec  1.01 GBytes  864 Mbits/sec
```

Отмена

Syslog

Syslog (. system log –) – , , IP.

1	syslog-, snmp- snmp-trap.	esr(config)# syslog snmp <SEVERITY>	<SEVERITY>– , (); • emerg – , ; • alert – , ; • crit – , ; • error – ; • warning – , ; • notice – ; • info – ;

2	syslog-, (Telnet, SSH) ()	esr(config)# syslog monitor <SEVERITY>	• debug – , , ; • none – syslog-.
3	syslog- ()	esr(config)# syslog cli-commands	
4	syslog	esr(config)# syslog file <NAME> <SEVERITY>	<NAME> – , , 31; <SEVERITY> syslog snmp.
5	()	esr(config)# syslog file-size <SIZE>	<SIZE> – , [10..10000000]
6	, ()	esr(config)# syslog max-files <NUM>	<NUM> – , [1.. 1000]
7	syslog syslog-	esr(config)#syslog host <HOSTNAME> <ADDR> <SEVERITY> <TRANSPORT> <PORT>	<HOSTNAME> – syslog-, 31 . . «all» no syslog host syslog-; <ADDR> – IP-, AAA.BBB.CCC.DDD, [0..255]; <SEVERITY> – , , Syslog ; <TRANSPORT> – , , : • TCP – TCP; • UDP – UDP; <PORT> – TCP/UDP-, , [1..65535], 514
8	()	esr(config)#syslog reload debugging	
9	()	esr(config)#syslog sequence-numbers	
10	().	esr(config)#syslog timestamp msec	
11	().	esr(config)#logging login on-failure	
12	().	esr(config)#logging syslog configuration	
13	().	esr(config)#logging userinfo	

:

:

- ;
- ;
- / ;
- .

IP- ESR – 192.168.52.8, ip- Syslog – 192.168.52.41. – UDP 514.



:

:

- gi1/0/1;
- IP- gi1/0/1.

:

, – info:

```
esr(config)# syslog file ESR info
```

IP Syslog-:

```
esr(config)# syslog host SERVER 192.168.17.30 info udp 514
```

:

```
esr(config)# logging login on-failure
```

syslog:

```
esr(config)# logging syslog configuration
```

/ :

```
esr(config)# logging service start-stop
```

:

```
esr(config)# logging userinfo
```

:

```
esr# commit
Configuration has been successfully committed
esr# confirm
Configuration has been successfully confirmed
```

:

```
esr# show syslog configuration
```

:

```
esr# show syslog ESR
```

.

1		esr# verify filesystem <detailed>	detailed – .

:

:

:

:

```
esr# verify filesystem
Filesystem Successfully Verified
```

ESR / .

1		esr(config)# archive	
2	()	esr(config-ahchive)# type <TYPE>	<TYPE> - . : • local; • remote; • both. : remote
3	()	esr(config-ahchive)# auto	
4	()	esr(config-ahchive)# by-commit	
5	(remote both)	esr(config-ahchive)# path <PATH>	<PATH> - , ,
6	(, auto)	esr(config-ahchive)# time-period <TIME>	<TIME> - , [1..35791394]. : 720
7	(, local both)	esr(config-ahchive)# count-backup <NUM>	<NUM> - . [1..100]. : 1

```
:
1 . tftp- 172.16.252.77 esr-example. - 30.
:
, IP-, tftp- .
:
:
```

```
esr# configure
esr(config)# archive
```

:

```
esr(config)# type both
```

:

```
esr(config-archive)# path tftp://172.16.252.77:/esr-example/esr-example.cfg
esr(config-archive)# count-backup 30
```

:

```
esr(config-archive)# time-period 1440
```

:

```
esr(config-archive)# auto  
esr(config-archive)# by-commit
```

```
1      tftp-      "esr-exampleYYYYMMDD_HHMMSS.cfg".,      flash:backup/      "config_YYYYMMDD_HHMMSS".      flash:backup/ 30 ,      .
```