SNMP Items

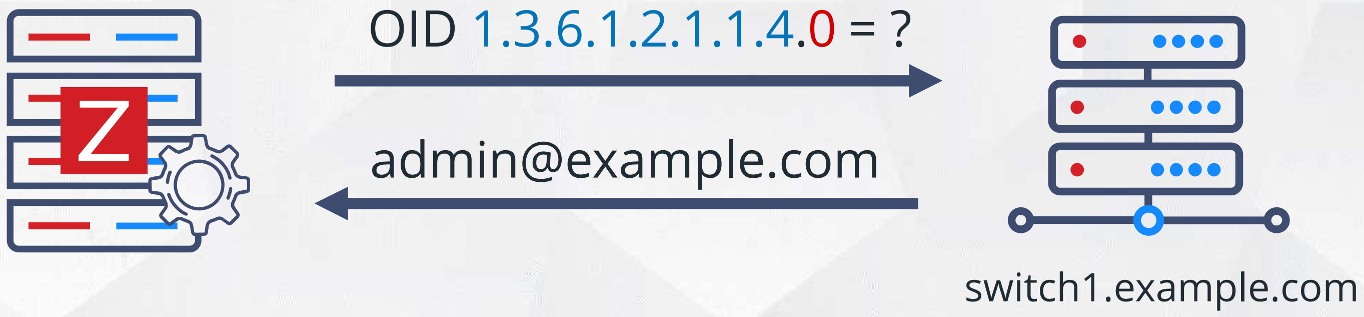
SNMP PROTOCOL

SNMP stands for Simple Network Management Protocol

- Zabbix sends SNMP GET request to a device
- Device sends back requested value or error message
- UDP protocol on port 161 is used for communication by default

Each device supports a list of OIDs (Object Identifiers)

- Each OID reports some metric
- The index with a value of 0 is required for non-indexed object



OID 1.3.6.1.2.1.1.4.0 = ?

admin@example.com

switch1.example.com

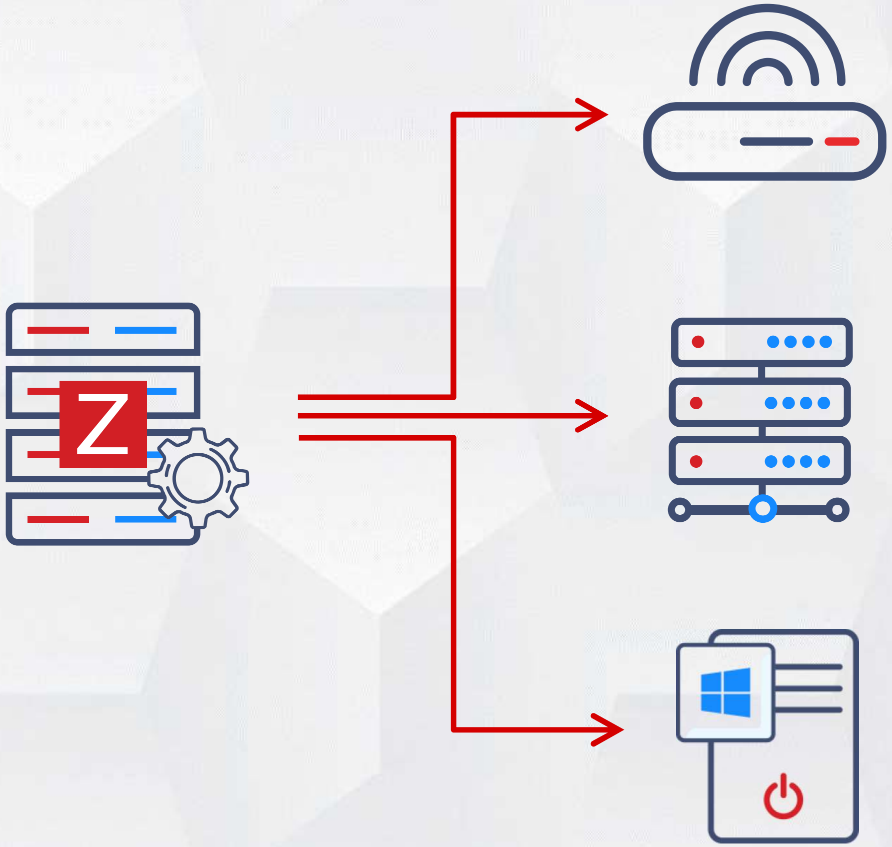
OID	Name
1.3.6.1.2.1.1.1	sysDescr
1.3.6.1.2.1.1.2	sysObjectID
1.3.6.1.2.1.1.3	sysUpTime
1.3.6.1.2.1.1.4	sysContact
1.3.6.1.2.1.1.5	sysName
1.3.6.1.2.1.1.6	sysLocation

SNMP monitoring is performed directly from Zabbix server or proxy:

- Works out-of-box when installed from packages
- net-snmp library is required
- Performed by poller process
- Timeout settings affect SNMP timeout

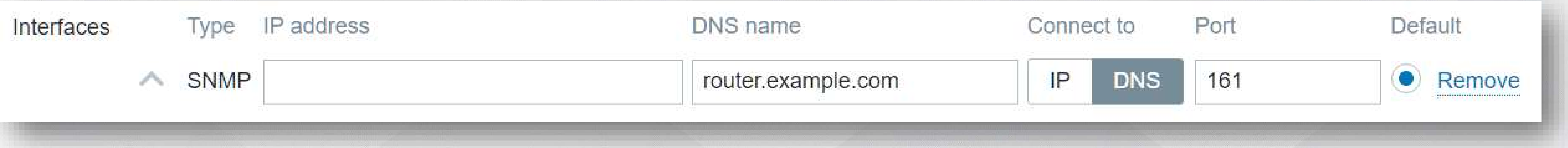
Zabbix can use SNMP to monitor:

- Network devices (switches, routers, storages, etc.)
- Regular computers and servers
- Applications
- Anything that supports the SNMP protocol



To use SNMP checks an interface with the type SNMP must be created first

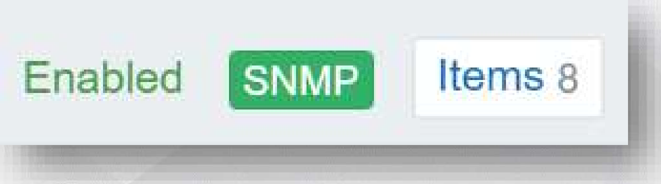
- Specify IP address or DNS name of the monitored device
- Fill in the required SNMP parameters
- Check or uncheck the "use bulk requests" option
  - ✓ Multiple values are requested simultaneously if "Use bulk request" is checked
  - ✓ Bulk request mode may not work properly on some devices



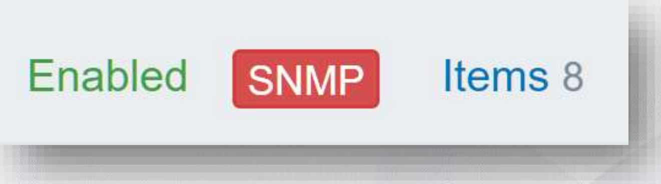
Interfaces	Type	IP address	DNS name	Connect to	Port	Default	
	SNMP		router.example.com	IP [✓ DNS]	161	●	Remove

The SNMP interface will become available (or unavailable if some problem exists)

- at least one SNMP item must be created on the interface



Enabled SNMP Items 8

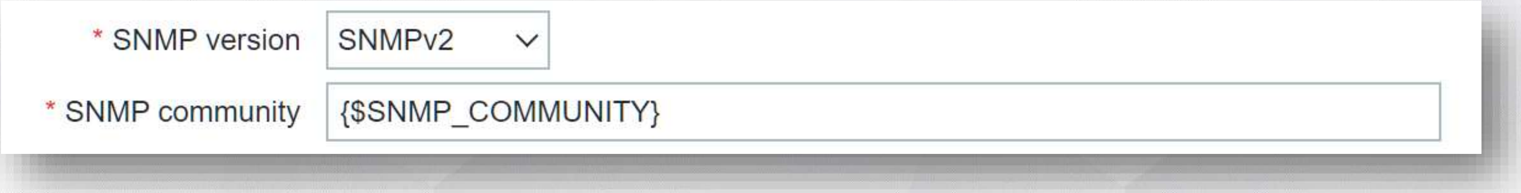


Enabled SNMP Items 8

SNMP VERSIONS

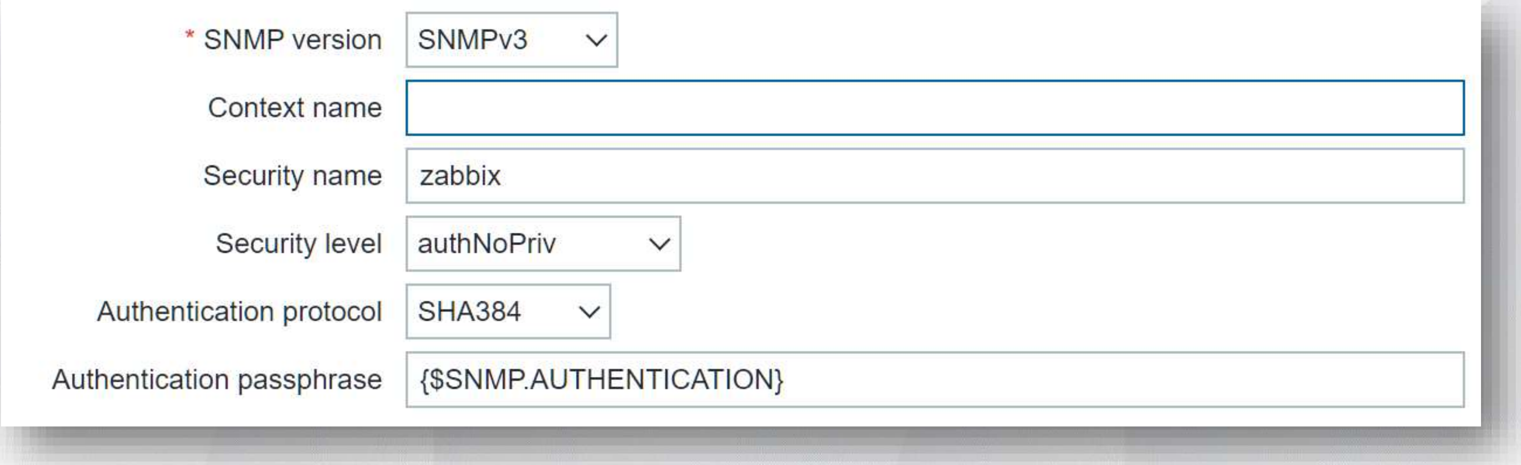
Zabbix supports SNMP versions v1, 2c and 3

- SNMPv1 and v2:
  - ✓ uses community names for read/write permissions



* SNMP version	SNMPv2
* SNMP community	{$SNMP_COMMUNITY}

- SNMPv3:
  - ✓ uses username and passphrases
  - ✓ provides authentication and encryption (modern algorithms are supported)
  - ✓ SNMP engine ID must be unique per device and SNMP engine boots must be persistent
  - ✓ The time must be synchronized on the SNMP device



* SNMP version	SNMPv3
Context name	
Security name	zabbix
Security level	authNoPriv
Authentication protocol	SHA384
Authentication passphrase	{$SNMP.AUTHENTICATION}

SNMP OID AND MIB

Zabbix supports both OID and MIB (Management Information Base) formats

- MIB is a formatted text file organized into a hierarchical format
  - ✓ MIB files contain details about the monitored objects
  - ✓ It is required to have MIB files on Zabbix server or proxy if MIB format is used

- An OID is an address that is used to differentiate between devices within the MIB hierarchy
  - ✓ Represented as a long sequence of numbers, coding the nodes, separated by the dots
  - ✓ No additional setup is required, OID format works out-of-box

.1	iso
.1.3	org
.1.3.6	dod
.1.3.6.1	internet
.1.3.6.1.2	mgmt
.1.3.6.1.2.1	mib-2
.1.3.6.1.2.1.1	system
.1.3.6.1.2.1.1.1	sysDescr
.1.3.6.1.2.1.1.2	sysObjectID
.1.3.6.1.2.1.1.3	sysUpTime

1.3.6.1.2.1.1.2
=
iso.org.dod.internet.mgmt.mib-2.system.sysObjectID

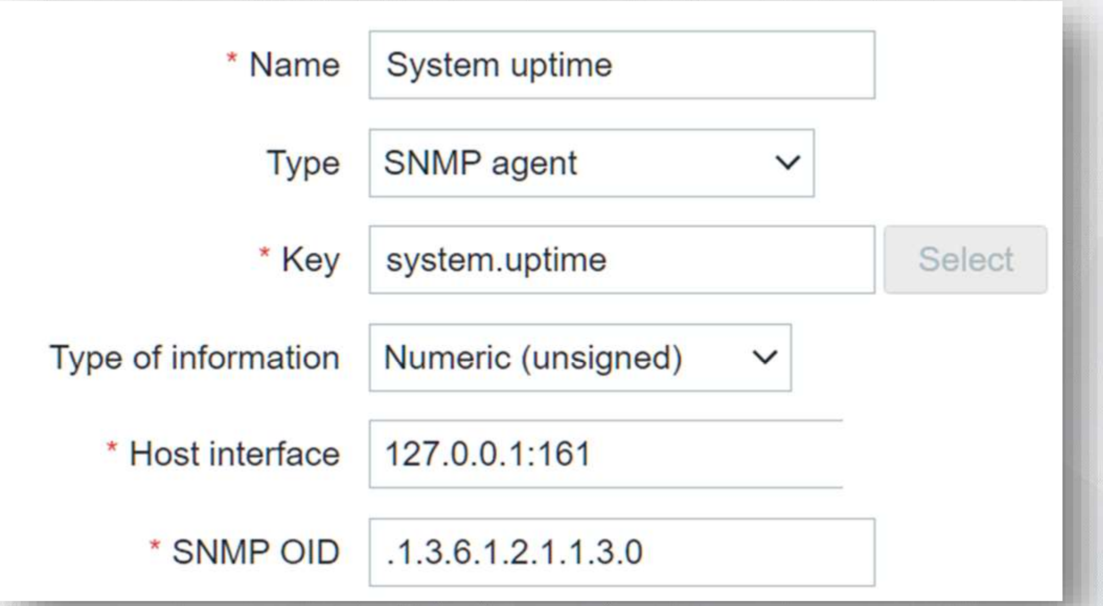
1.3.6.1.2.1.1.3
=
iso.org.dod.internet.mgmt.mib-2.system.sysUpTime

SNMP OID FIELD

OID or MIB information is entered into the SNMP OID field

- SNMP item key has a free format (must be unique per host or template)

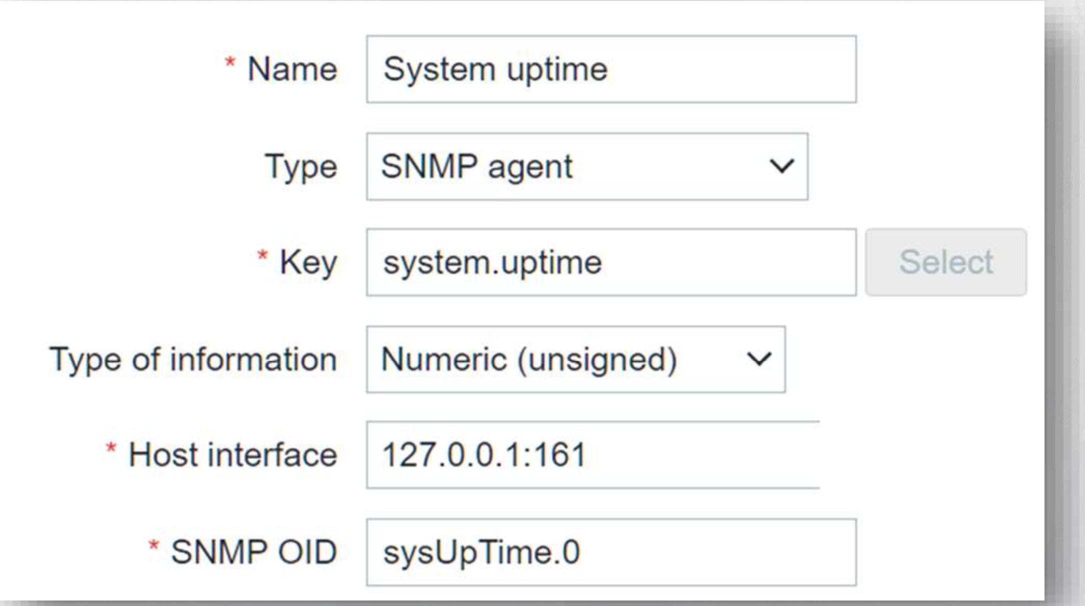
OID format



* Name	System uptime
Type	SNMP agent
* Key	system.uptime
Type of information	Numeric (unsigned)
* Host interface	127.0.0.1:161
* SNMP OID	.1.3.6.1.2.1.1.3.0

.1.3.6.1.2.1.1.3.0
or
iso(1).org(3).dod(6).internet(1).mgmt(2).mib-2(1)
.system(1).sysUpTime(3)

MIB format



* Name	System uptime
Type	SNMP agent
* Key	system.uptime
Type of information	Numeric (unsigned)
* Host interface	127.0.0.1:161
* SNMP OID	sysUpTime.0

sysUpTime
OBJECT-TYPE
SYNTAX TimeTicks ACCESS read-only
STATUS mandatory DESCRIPTION
"The time (in hundredths of a second) since the network management portion of the system was last re-initialized."

SNMP CLI UTILITIES

To get the CLI SNMP utilities, install the "net-snmp-utils" package:

- snmpget -c<community> -v<version> <IP ADDRESS or DNS> <OID>
  - ✓ Retrieves a single value from SNMP agent

```
# snmpget -c public -v2c router.example.com .1.3.6.1.2.1.1.3.0
SNMPv2-MIB::sysUpTime.0 = Timeticks: (1536925142) 14 days, 20:11:35.95
```

- snmpwalk -c<community> -v<version> <IP ADDRESS or DNS> <start of OID tree>
  - ✓ Retrieves multiple OIDs and values

```
# snmpwalk -c public -v2c router.example.com .1
SNMPv2-MIB::sysDescr.0 = HP-UX net-snmp B.10.20 A 9000/715
SNMPv2-MIB::sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.hpux10
SNMPv2-MIB::sysUpTime.0 = Timeticks: 1536925142) 14 days, 20:11:35.95
```

  - ✓ Output format can be specified by adding -On flag

```
# snmpwalk -c public -v2c -On router.example.com .1
.1.3.6.1.2.1.1.1.0 = HP-UX net-snmp B.10.20 A 9000/715
.1.3.6.1.2.1.1.2.0 = OID: enterprises.ucdavis.ucdSnmpAgent.hpux10
.1.3.6.1.2.1.1.3.0 = Timeticks: 1536925142) 14 days, 20:11:35.95
```

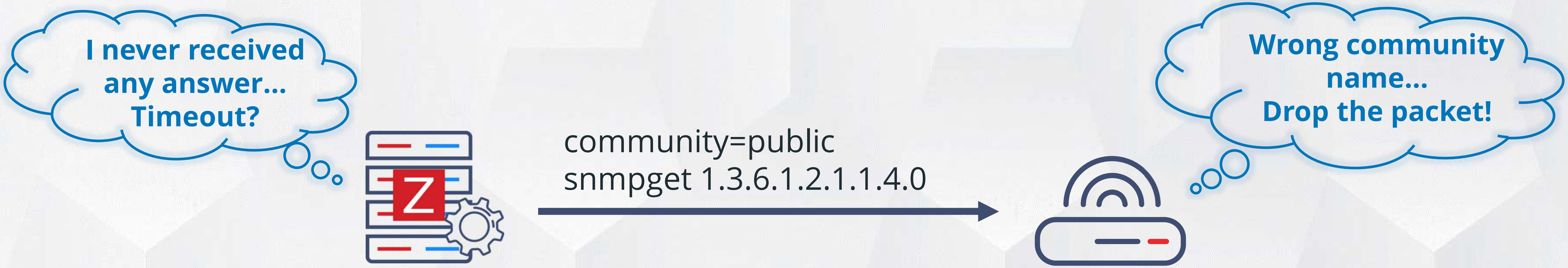
SNMP TROUBLESHOOTING

Common reasons, why SNMP requests may not work:

- Wrong credentials (community or username/password)
- UDP port 161 is closed by a local or remote firewall
- Zabbix server is not in the ACL (access control list) on the remote SNMP device
- Timeout is too short for Zabbix server or proxy
- Requested OID is not known by the monitored device

SNMP timeout message does not always mean a communication timeout

- The UDP packet may be just dropped, and no response received back

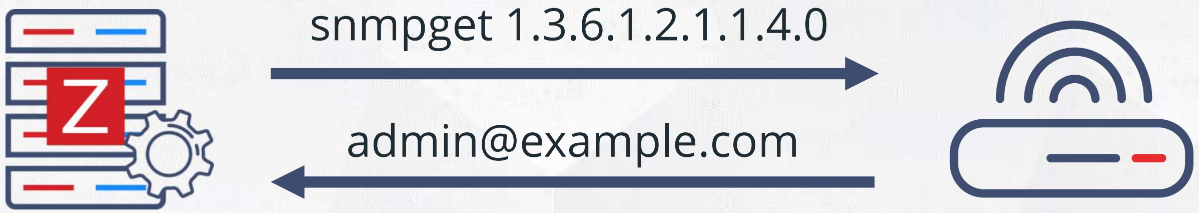

I never received any answer… Timeout?
community=public
snmpget 1.3.6.1.2.1.1.4.0
Wrong community name… Drop the packet!

SNMP trap items

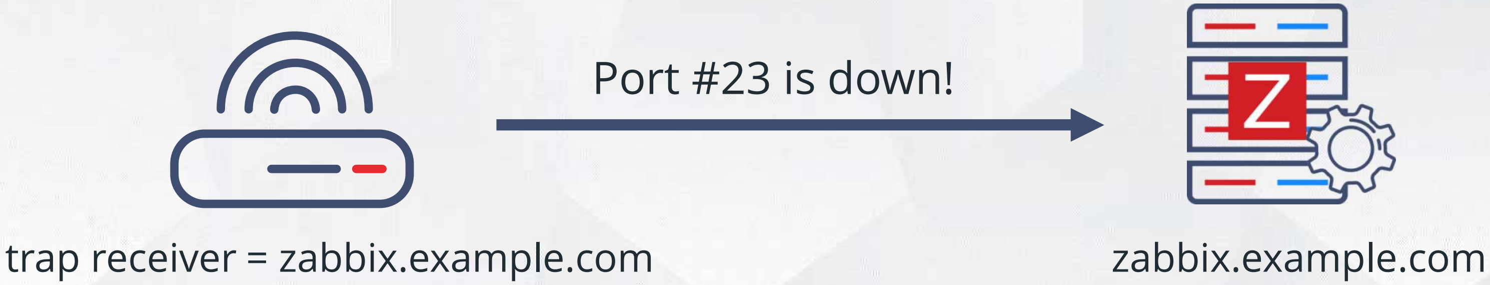
15 minutes

SNMP traps work differently from SNMP items:

- SNMP items request information from the device (polling)



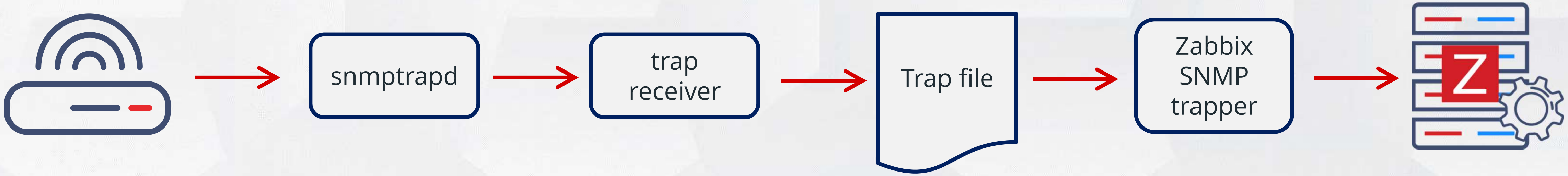
- SNMP trap items receive messages generated by the SNMP device itself (trapping)
  - ✓ Problem situations or thresholds are defined on the device
  - ✓ Each device type has its own unique trap items (read the documentation or MIB files)
  - ✓ When a problem is detected, the device will send SNMP messages to all trap recipients



RECEIVING SNMP TRAPS

Receiving SNMP traps in Zabbix is designed to work with snmptrapd

- UDP protocol on port 162 is used for communication by default
- snmptrapd receives a trap and passes the trap to the trap receiver
  - ✓ snmptrapd must be installed and started
- Trap receiver parses, formats and writes the trap to a file
  - ✓ Any trap receiver can be used (by example zabbix_trap_receiver.pl or SNMPTT)
- Zabbix SNMP trapper process reads and parses the trap file
  - ✓ SNMP trapper must be started on Zabbix server or proxy
- Zabbix checks all SNMP trap items with SNMP interface address matching the trap address
  - ✓ If the address cannot be matched with any host, the trap is logged in Zabbix server log file



SNMP TRAP CONFIGURATION

Zabbix configuration file has two settings for SNMP traps

- SNMP trapper process must be started

```
### Option: StartSNMPTrapper
#       If 1, SNMP trapper process is started.
StartSNMPTrapper=1
```

- Correct SNMP trap file location must be specified
  - ✓ The file location must match the location specified in the trap receiver

```
### Option: SNMPTrapperFile
#       Temporary file used for passing data from SNMP trap daemon to the server.
#       Must be the same as in zabbix_trap_receiver.pl or SNMPTT configuration file.
SNMPTrapperFile=/tmp/zabbix_traps.tmp
```

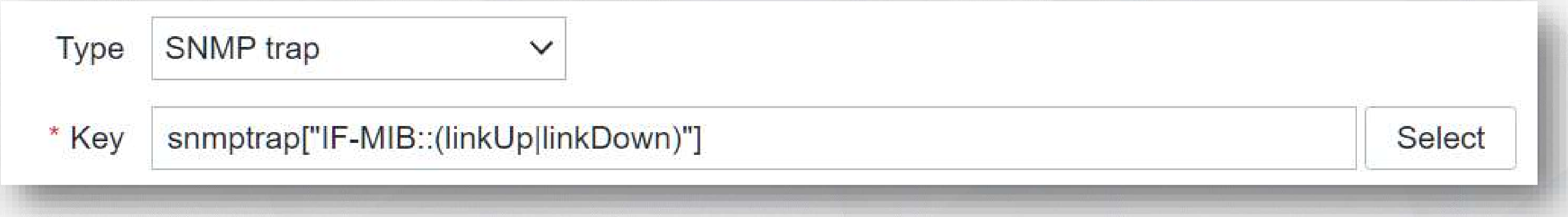
Zabbix does not provide any log rotation system for the trap file

- Use logrotate or other method to rotate the trap file

SNMP TRAP ITEM TYPES

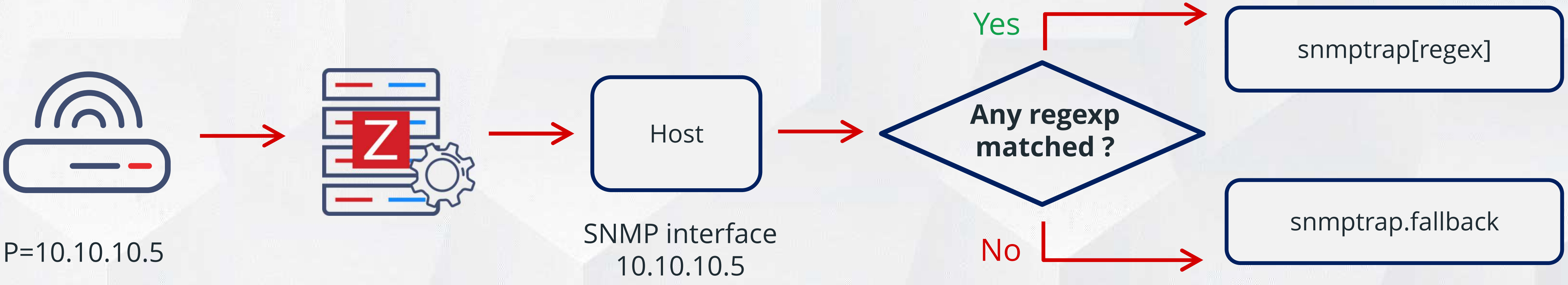
Two types of SNMP trap items can be created (only for SNMP interfaces):

- snmptrap[regexp]
  - ✓ Catches all SNMP traps on the host that match the regular expression specified in the parameter
  - ✓ Any part of the trap can be used as regular expression
  - ✓ User macros are supported in the parameter



Type	SNMP trap
* Key	snmptrap["IF-MIB::(linkUp|linkDown)"]

- snmptrap.fallback
  - ✓ Catches all SNMP traps that were not caught by any of the "snmptrap[regexp]" items



IP=10.10.10.5
Host
SNMP interface
10.10.10.5
Any regexp matched ?
Yes
snmptrap[regex]
No
snmptrap.fallback