

SNMP (Simple Network Management protocol)

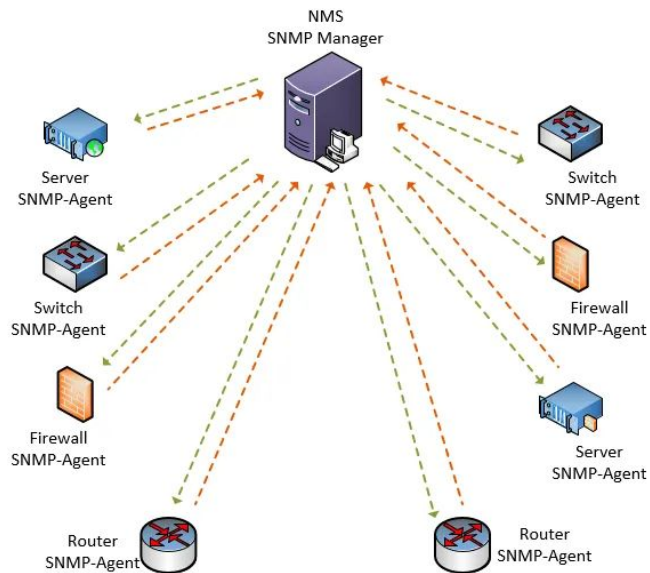
André Arthur
Bartolomeu Nunes Filho
Cristiliano Negreiros
João Nogueira





SNMP

- Monitoramento e gerenciamento de redes.
- Núcleo com um conjunto de simples operações
- Geralmente associado a gerenciamento de roteadores.





Comandos

GetRequest

Usado para retirar um pedaço de informação de gerenciamento.

SetRequest

Usado para fazer uma mudança no subsistema gerido.

GetNextRequest

Usado interativamente para retirar sequências de informação de gerenciamento.

GetBulkRequest

Usado para retirar informações de um grupo de objetos.



Comandos

Response

Retorna associações de variáveis e reconhecimento do agente para o gerente para GetRequest, SetRequest, GetNextRequest, GetBulkRequest e InformRequest.

Trap

Usado para reportar uma notificação ou para outros eventos assíncronos sobre o subsistema gerido.

InformRequest

Notificação assíncrona reconhecida com confirmação do recebimento da PDU de trap.



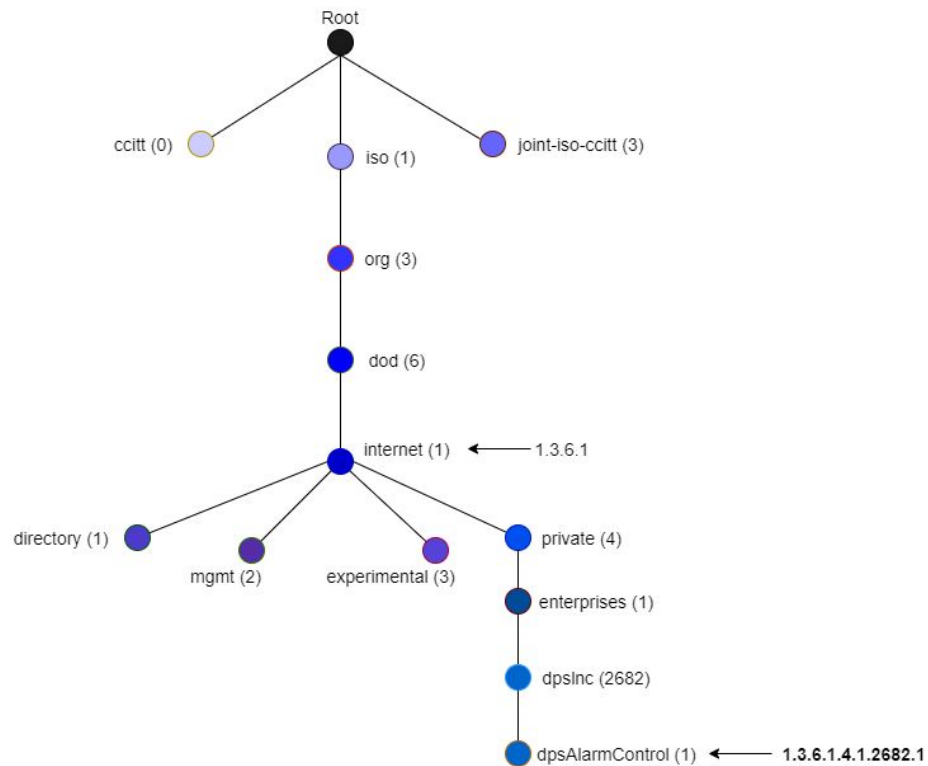
MIB's e OID's

MIB: MIB (Management Information Base) é uma coleção de informações organizadas hierarquicamente. Eles são acessados usando um protocolo como SNMP.

OID:OIDs significa identificadores de objetos. Os OIDs identificam exclusivamente objetos gerenciados em uma hierarquia MIB.



Árvore de MIB





Código

```
from struct import pack
import socket

### Definindo parâmetros do SNMP

OID = "1.2.1.1.1.0" # descrição do sistema
COMM = b'public' #nome da comunidade

### Definindo parâmetros do gerente

ipDest = b'127.0.0.1' #local host
portDest = 161

## Criando conexão no socket
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

#verifica criação do socket
if(s == -1):
    print('\n\nNão foi possível criar o socket')
else:
    print('\n\nsocket foi criado em ', s, ('\n'))
```

Código

```
### Montando mensagem SNMP de trás pra frente

# Value field
val = b'mensagem de teste do snmp'
snmpVal = pack("bb{s}".format(len(val)),4,len(val),val) #byte 0x05 - data type nulo

# Object Field
OID = OID.split(".") #System Description
snmpOid = pack("2b",0x2b, 0x06) + bytearray(int(x) for x in OID ) # adicionando iso.3 ao OID

# Sequence / Varbind Type Field
snmpVarbind = snmpOid + snmpVal
snmpVarbindSeq = pack('b',6) + len(snmpOid).to_bytes(1,'little') + snmpVarbind

# Sequence / Varbind List Field
snmpVarbindList = pack('b',0x30) + len(snmpVarbindSeq).to_bytes(1,"little") + snmpVarbindSeq
snmpVarbindList = pack('b',0x30) + len(snmpVarbindList).to_bytes(1,"little") + snmpVarbindList

# Error Index
ErrIndex = pack("b",0x0)
pdu = pack("b",0x02) + len(ErrIndex).to_bytes(1,"little") + ErrIndex + snmpVarbindList

# Error
Err = pack("b",0x0)
pdu = pack('b',2) + len(Err).to_bytes(1,"little") + Err + pdu
```




```
# Request ID
```

```
RqID = pack("b",0x2)
```

```
pdu = pack('b',2) + len(RqID).to_bytes(1,"little") + RqID + pdu
```

```
pduType = pack('B',0xa0) # 0xa0 = getRequest, 0xa2 = getResponse
```

```
pdu = pduType + len(pdu).to_bytes(1,"little") + pdu
```

```
# Community
```

```
lenComm = len(COMM)
```

```
snmpComm = pack("{}s".format(lenComm),COMM)
```

```
snmpProtocolHeader = pack('b',0x04) + lenComm.to_bytes(1,"little") + snmpComm + pdu
```

```
# Versão
```

```
snmpVersion = pack('b',0x0)
```

```
snmpProtocolHeader = pack('b',0x2) + len(snmpVersion).to_bytes(1,"little") + snmpVersion + snmpProtocolHeader
```

```
# Mensagem SNMP Final
```

```
snmpMessage = pack('b',0x30) + len(snmpProtocolHeader).to_bytes(1,"little") + snmpProtocolHeader
```

```
print(f'Mensagem SNMP raw bytes:\n{snmpMessage}')
```



```
## Transmitindo mensagem
```

```
print("\n\nTransmitindo mensagem\n\n")  
s.sendto(snmpMessage,(ipDest,portDest))
```

```
try:
```

```
    Rxbuf = s.recv(2000)#tamanho dos dados pro buffer
```

```
    print (f'Resposta Recebida!\n\nString Resposta:\n\n{Rxbuf.decode("utf-8", errors="ignore")}')  

```

```
except socket.timeout:
```

```
    print ('time out!!!')
```

```
s.close()
```

```
print ('\n\nFim da Operacao. Socket fechado.')
```



Resposta

```
socket foi criado em <socket.socket fd=1428, family=AddressFamily.AF_INET, type=SocketKind.SOCK_DGRAM, proto=0>
```

Mensagem SNMP raw bytes:

```
b"0?\x02\x01\x00\x04\x06public\xa02\x02\x01\x02\x02\x01\x00\x02\x01\x000'0%\x06\x08+\x06\x01\x02\x01\x01\x01\x00\x04\x19mensagem de teste do snmp"
```

Transmitindo mensagem

Resposta Recebida!

String Resposta:

```
0000public00000000+00000000Hardware: Intel64 Family 6 Model 142 Stepping 9 AT/AT COMPATIBLE - Software: Windows Version 6.3  
(Build 19043 Multiprocessor Free)
```



Versão com pysnmp

```
from pysnmp.entity.rfc3413.oneliner import cmdgen

def snmpget(oid,ip):
    cmdGen = cmdgen.CommandGenerator()

    errorIndication, errorStatus, errorIndex, varBinds = cmdGen.getCmd(
cmdgen.CommunityData('public'),
cmdgen.UdpTransportTarget((ip,161)),
    oid
    )
    if errorIndication:
        print(errorIndication)
    else:
        if errorStatus:
            print('%s at %s' % {
                errorStatus.prettyPrint(),
                errorIndex and varBinds[int(errorIndex)-1] or '?'
            })
        else:
            for name,val in varBinds:
                print('%s = %s' % (name.prettyPrint(), val.prettyPrint()))
```



Resposta

```
snmpget('1.3.6.1.2.1.1.1.0', 'localhost') #SystemDescr  
snmpget('1.3.6.1.2.1.1.5.0', 'localhost') #SystemName  
snmpget('1.3.6.1.2.1.1.6.0', 'localhost') #SystemLocation
```

```
SNMPv2-MIB::sysDescr.0 = Hardware: Intel64 Family 6 Model 142 Stepping 9 AT/AT COMPATIBLE - Software: Windows Version 6.3 (Build 19043 Multiprocessor Free)  
SNMPv2-MIB::sysName.0 = LAPTOP-TISLCRU2  
SNMPv2-MIB::sysLocation.0 = Brasil
```



Resultado no FrontEnd

The image shows a web application interface for querying SNMPv2 MIBs. It features a dark blue background with a bokeh effect of orange and yellow light spots. In the top-left corner, there is a black box containing the names 'Cristiliano', 'André', 'João Vitor', and 'Bartolomeu' in white text. The main content area is a dark blue rounded rectangle. Inside this rectangle, there are two input fields: the first is labeled 'OID' and contains the value '1.3.6.1.2.1.1.6.0'; the second is labeled 'IP' and contains the value 'localhost'. Below these fields is a yellow button with the text 'Enviar'. Underneath the button, a white box displays the result: 'SNMPv2-MIB::sysLocation.0 = recife'.

Cristiliano
André
João Vitor
Bartolomeu

OID
1.3.6.1.2.1.1.6.0

IP
localhost

Enviar

SNMPv2-MIB::sysLocation.0 =
recife

Obrigado!

