Monitor AireOS WLC via SNMP with OIDs

Contents

Introduction

This document describes how to configure and monitor SNMP on Cisco Wireless LAN Controller (WLC).

Prerequisites

Requirements

Cisco recommends that you have a default Simple Network Management Protocol (SNMP) tool on your operating system or knowledge to install one.

Components Used

This document is not restricted to specific software and hardware versions. All tests were performed on a 3504 WLC running image version 8.9 and MacOS 10.14. OIDs in this article are also valid on earlier AireOS releases and other AireOS-based wireless controllers (8540/5508/5520/2504).

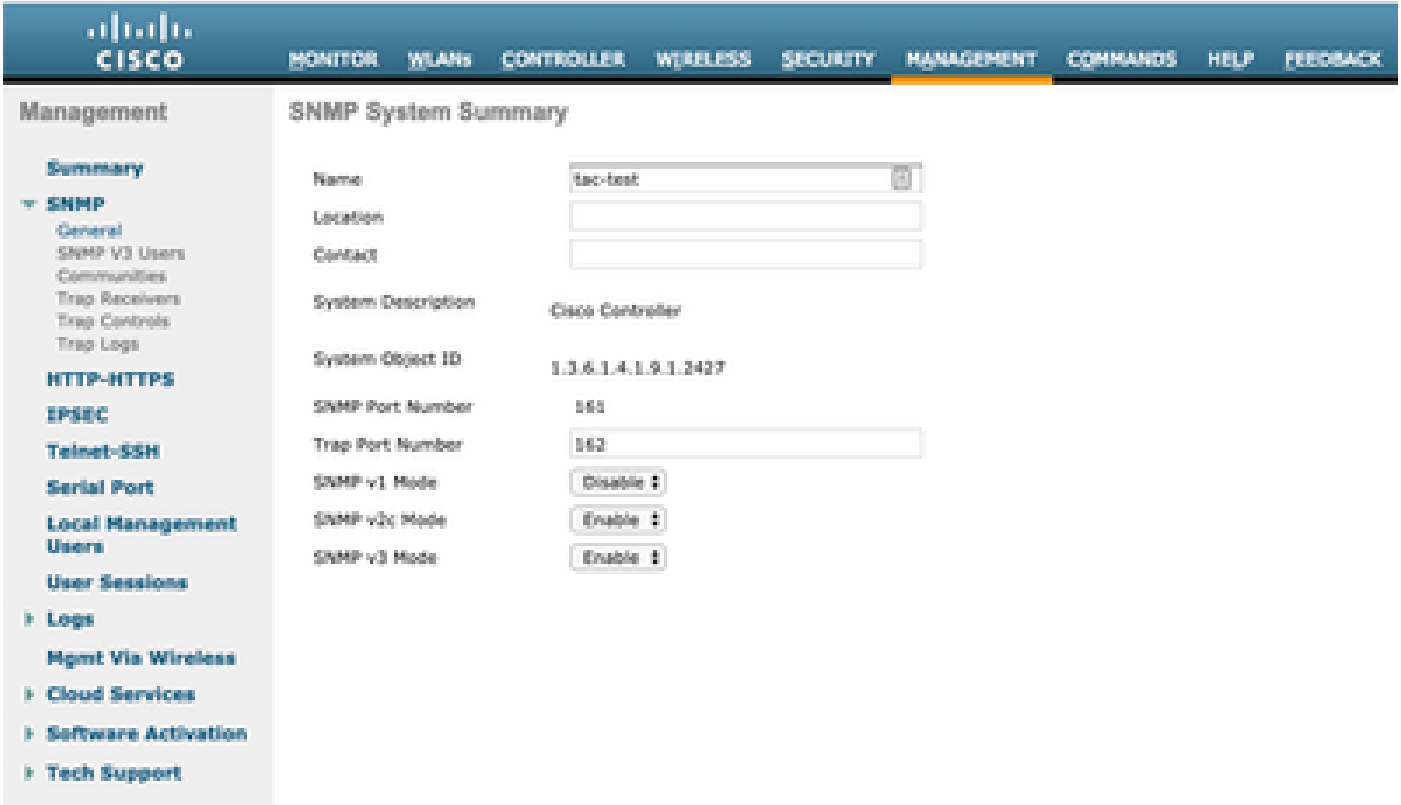
The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Configure SNMP Settings on WLC

SNMPv2c is a community-based version of SNMP and all communication between the devices is in clear text. SNMPv3 is the most secure version which offers message integrity checks, authentication, and encryption of the packets. SNMPv1 is extremely outdated but still exists to provide legacy software compatibility.

Note: SNMPv2c is enabled by default with community private that has read and write privileges, and community public that has read-only privileges. It is recommended to remove them and create a new community with a different name.

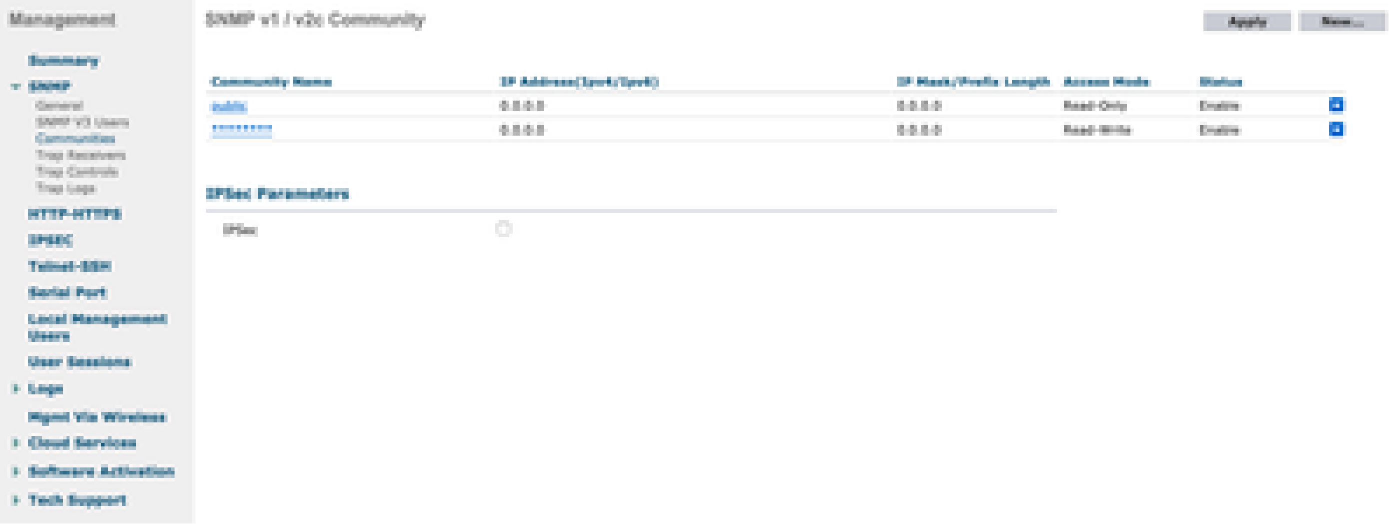
In this article, only SNMPv2c and SNMPv3 are used. Log into the web interface of the controller. Under Management > SNMP > General ensure to enable the desired version of the protocol.



SNMP System Summary

Name	tac-test
Location	
Contact	
System Description	Cisco Controller
System Object ID	1.3.6.1.4.1.9.1.2427
SNMP Port Number	161
Trap Port Number	162
SNMP v1 Mode	Disable
SNMP v2c Mode	Enable
SNMP v3 Mode	Enable

Under the communities menu, all currently created communities are displayed.



SNMP v1 / v2c Community

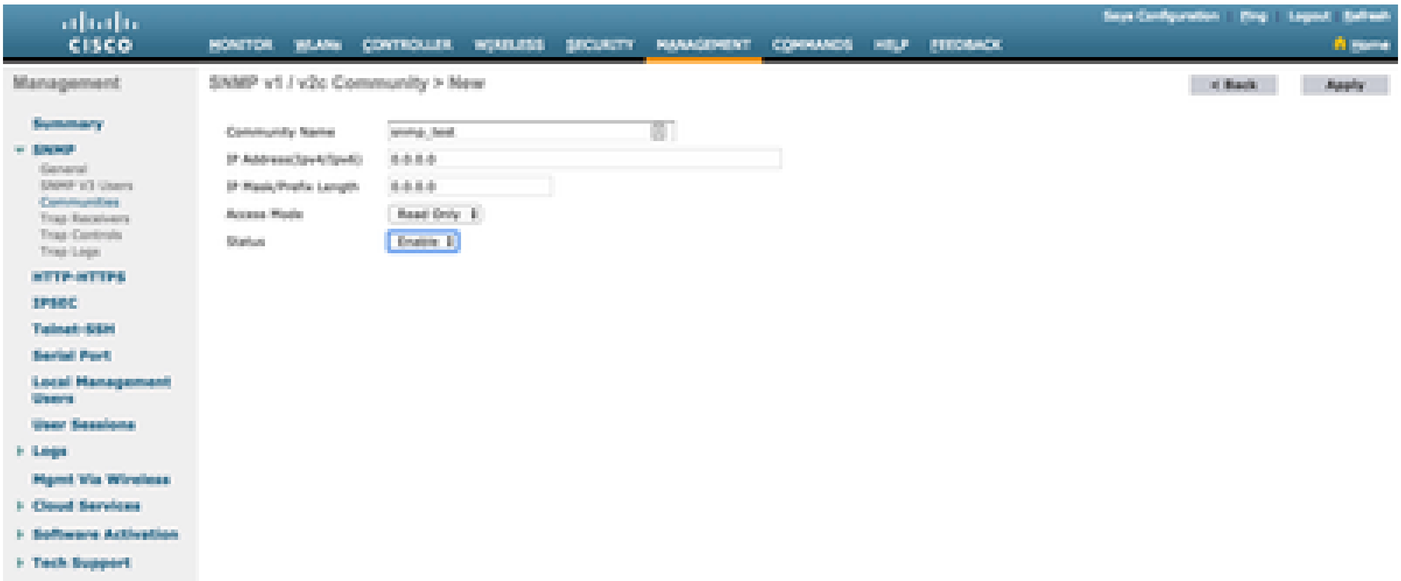
Community Name	IP Address(Ipv4/Ipv6)	IP Mask/Prefix Length	Access Mode	Status
public	0.0.0.0	0.0.0.0	Read-Only	Enable
*********	0.0.0.0	0.0.0.0	Read-Write	Enable

IPSec Parameters

IPSec ○

It is best practice to remove default pre-configured communities and create a new one. IP address and netmask behave like an access list. By default, both are set to 0.0.0.0, which means that all IP addresses are allowed to make SNMP queries for this community. The access mode field is left as Read Only as this community is to be used only to monitor, and not for the configuration of the WLC.

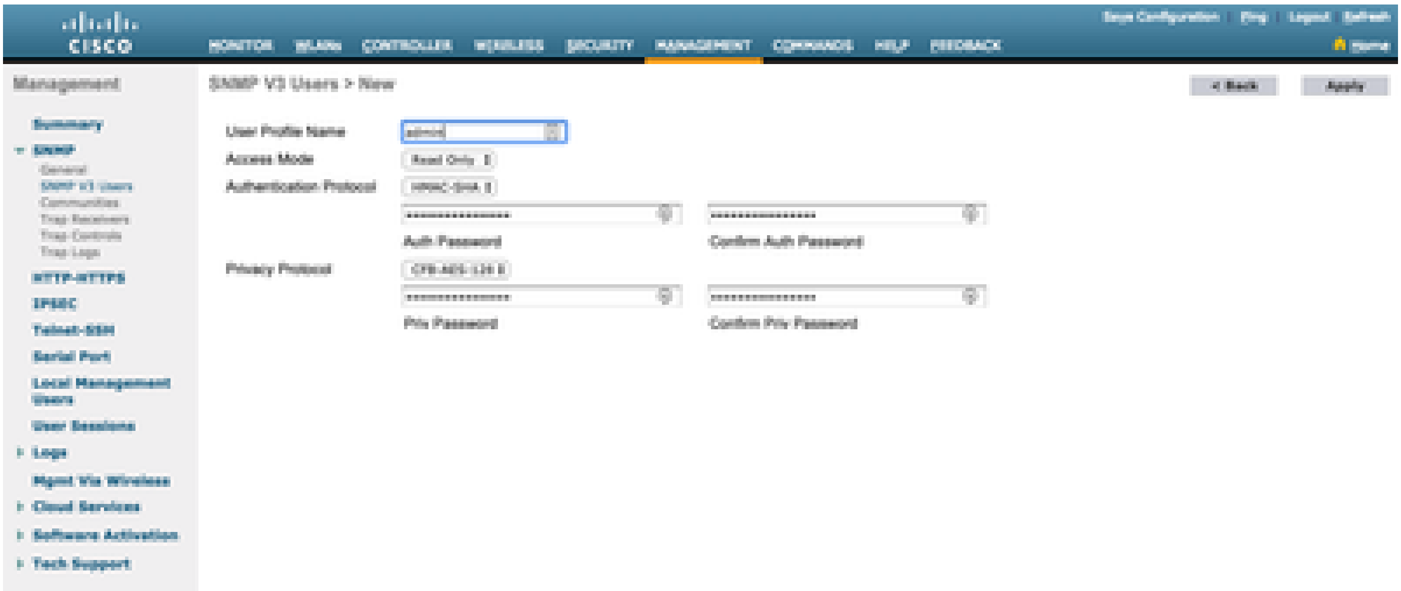
Note: All versions earlier than 8.7.1.135 are affected by a Cisco bug ID CSCvg61933 where the netmask cannot be set to 255.255.255.255. Either upgrade the controller to the latest recommended release later than 8.7.1.135 or use this command in the CLI to create a new community `config snmp community ipaddr <ip_address> <netmask> <community_name>`.



SNMP v1 / v2c Community > New

Community Name: snmp_test
IP Address(Ipv4/Ipv6): 0.0.0.0
IP Mask/Prefix Length: 0.0.0.0
Access Mode: Read Only
Status: Enable

Under the SNMP V3 Users menu, you can see all the configured users, their privileges, and protocols used for authentication and encryption. The New button allows the creation of a new user. It is recommended to choose HMAC-SHA as an authentication protocol, and CFB-AES-128 as a privacy protocol. Create a user named **admin** with authentication and privacy password set to Cisco123Cisco123.



SNMP V3 Users > New

User Profile Name: admin
Access Mode: Read Only
Authentication Protocol: HMAC-SHA
Auth Password: ••••••••••••••••
Confirm Auth Password: ••••••••••••••••
Privacy Protocol: CFB-AES-128
Priv Password: ••••••••••••••••
Confirm Priv Password: ••••••••••••••••

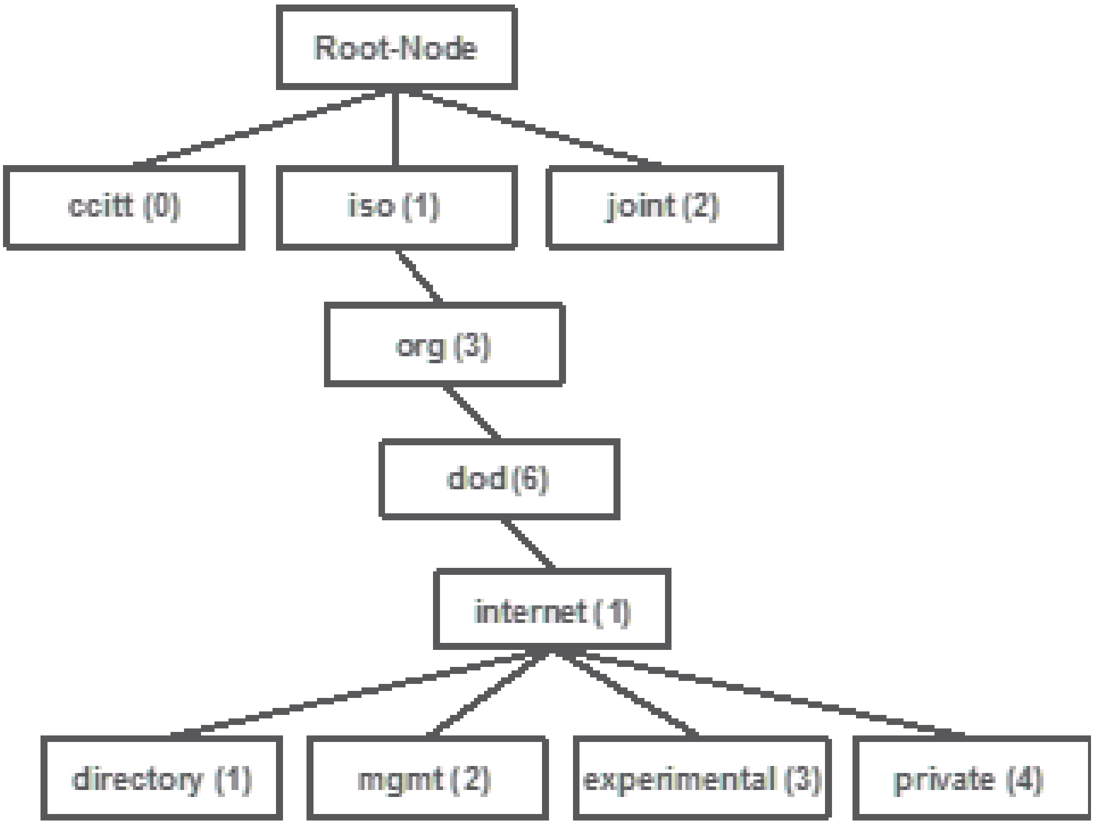
Object Names and Object IDs (OIDs)

What are Object Names and OIDs

OIDs are unique identifiers that represent a certain variable or object. For example, the current CPU usage is considered a variable whose values can be retrieved when you call upon their object ID. Each OID is unique and no two must be the same across the world, quite similar to a MAC address. These identifiers are in a tree hierarchy, and each OID can be tracked down back to its root. Each vendor has its own branch after a common root.

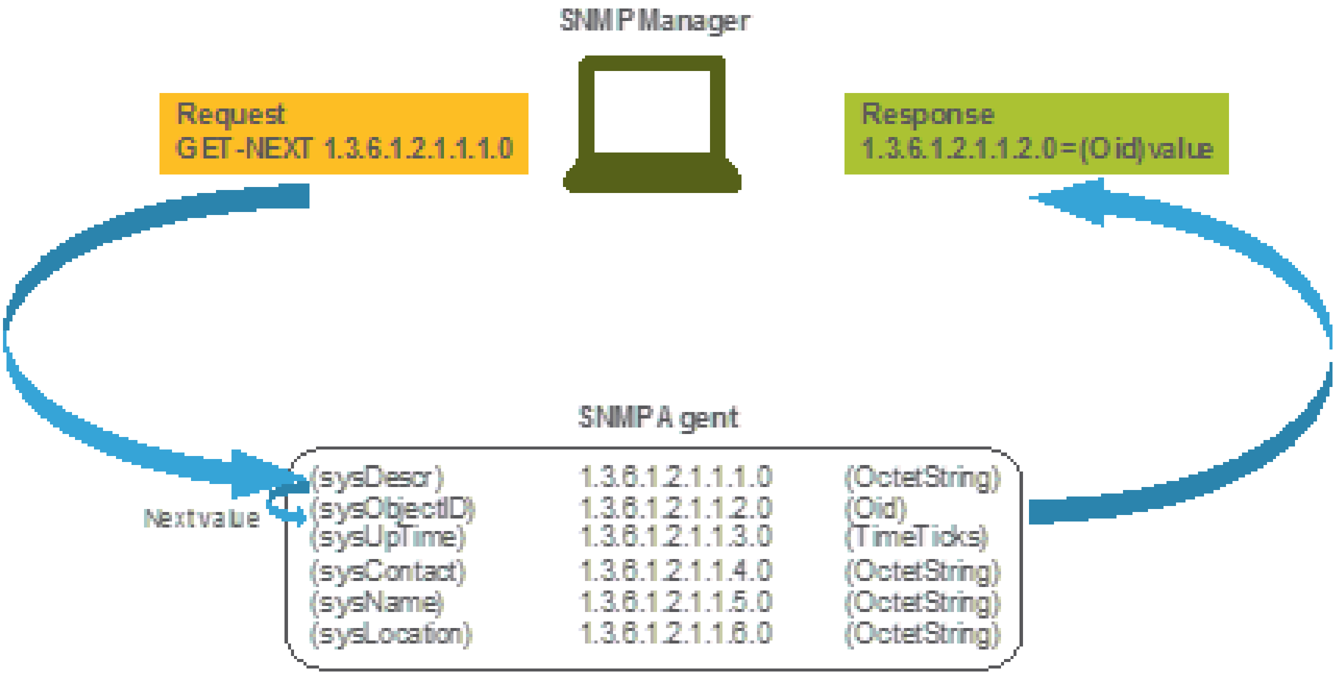
An analogy can be a home address, where the root is the country or state, then a city zip code, the street, and finally the home number.

The numbers followed by a dot represent each step it takes to get to a certain point in that tree or branch.



All these values are stored in a Management Information Base (MIB) in each network device. Each identifier has a name and definition (range of possible values, types, and so on).

You do not have to load MIBs on your SNMP tool in order to use SNMP and to query a device, as long as a valid OID is known. The device responds with the value that is stored in the variable that the OID represents. For example, in the image shown, the SNMP manager queries the SNMP agent of a device for its system description with the use of the OID 1.3.6.1.2.1.1.1.0.



(sysDescr)	1.3.6.1.2.1.1.1.0	(OctetString)
(sysObjectID)	1.3.6.1.2.1.1.2.0	(Oid)
(sysUpTime)	1.3.6.1.2.1.1.3.0	(TimeTicks)
(sysContact)	1.3.6.1.2.1.1.4.0	(OctetString)
(sysName)	1.3.6.1.2.1.1.5.0	(OctetString)
(sysLocation)	1.3.6.1.2.1.1.6.0	(OctetString)

If you load the MIB into your query tool, you can use it to translate OID numbers into names and discover their definitions.

MIBs and List of all Object Names and IDs on Cisco WLCs

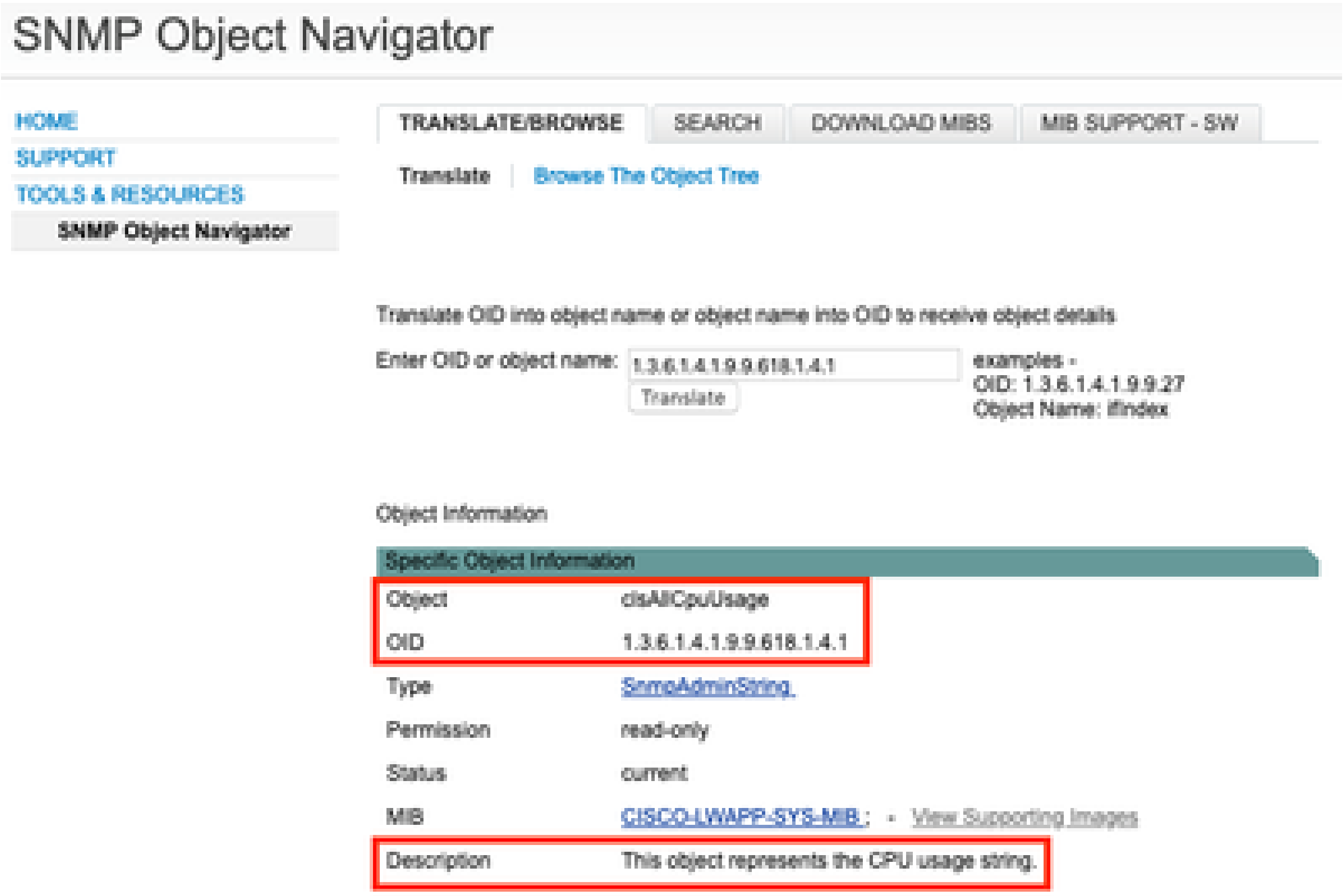
As of May 2019, a simple, user-friendly table that contains every single available object name and their respective OIDs for Wireless LAN Controllers does not exist. As an alternative, Cisco offers Management Information Base (MIB), which cannot be easily readable but contains all available object names and their description. Cisco 3504 WLC MIB can be downloaded here.

The downloaded archive file contains multiple .my text files that can either be imported into any third-party SNMP monitoring server or simply opened with a regular text editor. In order to find the OID of a specific object name, you first need to locate the exact file that contains it.

For example, all objects related to monitoring the physical state of the device (like temperature and fan speed) are located inside a MIB called CISCO-ENVMON-MIB.my. Here, ciscoEnvMonFanState is the object name which is used to provide the state of the WLC fan. MIB files have the syntax shown. Information about the fan state object looks like this:

```
ciscoEnvMonFanState OBJECT-TYPE
        SYNTAX     CiscoEnvMonState
        MAX-ACCESS read-only
        STATUS     current
        DESCRIPTION
                "The current state of the fan being instrumented."
        ::= { ciscoEnvMonFanStatusEntry 3 }
```

Most third-party monitoring software relies on OIDs, and not object names. Translation between the object name and object ID can be done with the use of the Cisco SNMP Object Navigator Tool. Enter the object name into the search bar. The output provides the OID and a short description. Additionally, the same tool can be used to find the corresponding object name of the OID.



Translate OID into object name or object name into OID to receive object details

Enter OID or object name: 1.3.6.1.4.1.9.9.618.1.4.1

examples -
OID: 1.3.6.1.4.1.9.9.27
Object Name: ifIndex

Object Information

Specific Object Information	
Object	clsAllCpuUsage
OID	1.3.6.1.4.1.9.9.618.1.4.1
Type	SnmpAdminString
Permission	read-only
Status	current
MIB	CISCO-LWAPP-SYS-MIB ; - View Supporting Images
Description	This object represents the CPU usage string.

Use of OIDs to Monitor the State of WLC

After you acquire the OID of the object that needs to be monitored, the first SNMP query can be executed. These examples showcase how to acquire a WLC CPU usage per core (OID = 1.3.6.1.4.1.9.9.618.1.4.1) for the SNMPv2 community snmp_test and SNMPv3 user admin with SHA Auth password Cisco123Cisco123 and AES Privacy password set to Cisco123Cisco123 . The controller management interface is located on 10.48.39.164.

Monitor via SNMPwalk

SNMPwalk is an SNMP application that uses SNMP GETNEXT requests to query a network entity for a tree of information. It is present by default on MacOS and most Linux distributions. For SNMPv2c, the command has the syntax:

```
snmpwalk -v2c -c <community_name> <WLC_management_interface_ip> <OID>
```

For example:

```
VAPEROVI-M-H1YM:~ vaperovi$ snmpwalk -v2c -c snmp_test 10.48.39.164 1.3.6.1.4.1.9.9.618.1.4.1

SNMPv2-SMI::enterprises.9.9.618.1.4.1.0 = STRING: "0%/1%, 0%/1%, 0%/1%, 0%/1%"
```

If SNMPv3 is used, the command has the syntax:

```
snmpwalk -v3  -l authPriv -u <username> -a [MD5|SHA] -A <auth_password> -x [AES|DES] -X <priv_password>
```

Choose MD5/SHA and AES/DES based on how you created the SNMPv3 user on the controller.

For example:

```
VAPEROVI-M-H1YM:~ vaperovi$ snmpwalk -v3  -l authPriv -u admin -a SHA -A Cisco123Cisco123 -x AES -X Cis

SNMPv2-SMI::enterprises.9.9.618.1.4.1.0 = STRING: "0%/1%, 0%/1%, 0%/0%, 0%/1%"
```

Monitoring via Python 3 and pysmnp Library

These code snippets are written in Python 3.7 and utilize the `pysnmp` module (`pip install pysnmp`) to make SNMP queries for CPU utilization of Cisco 3504 WLC. These examples use the same SNMPv2 community and SNMPv3 user created in one of the previous chapters. Simply replace the variable values and integrate the code with your own custom scripts.

SNMPv2c example:

```
from pysnmp.hlapi import *

communityName = 'snmp_test'
ipAddress = '10.48.39.164'
OID = '1.3.6.1.4.1.14179.2.3.1.13.0'

errorIndication, errorStatus, errorIndex, varBinds = next(
    getCmd(SnmpEngine(),
           CommunityData(communityName),
           UdpTransportTarget((ipAddress, 161)),
           ContextData(),
           ObjectType(ObjectIdentity(OID)))
)

if errorIndication:
    print(errorIndication)
elif errorStatus:
    print('%s at %s' % (errorStatus.prettyPrint(),
                        errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
else:
    for varBind in varBinds:
        print(' = '.join([x.prettyPrint() for x in varBind]))
```

Output:

```
SNMPv2-SMI::enterprises.14179.2.3.1.13.0 = 73
```

SNMPv3 example:

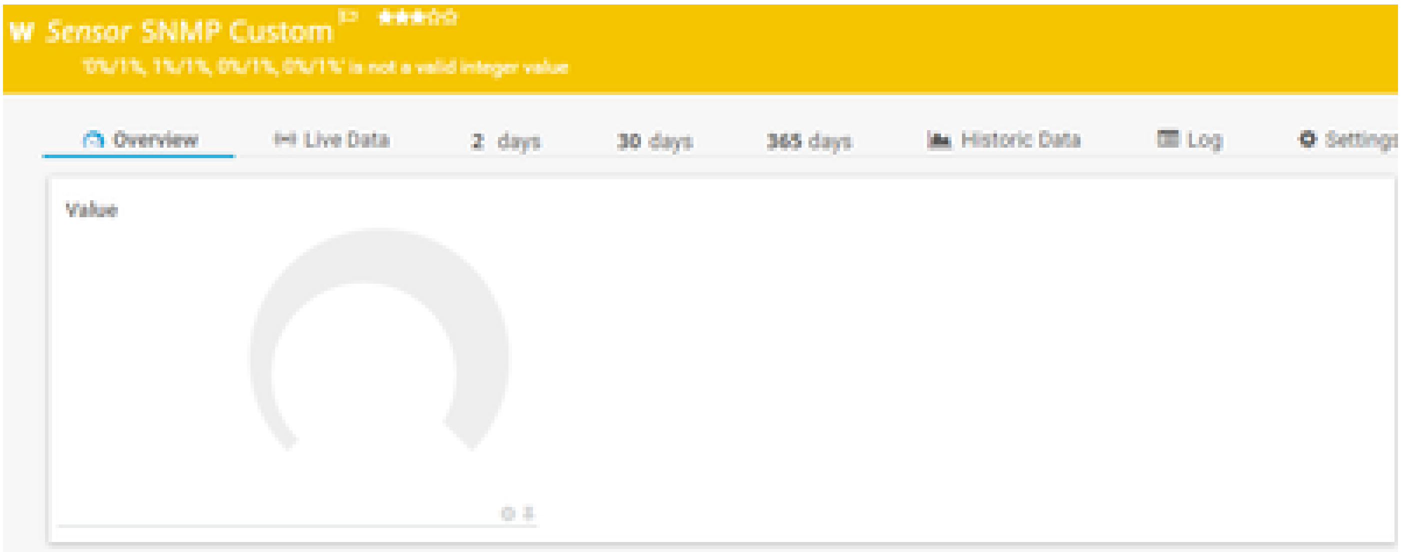
```
from pysnmp.hlapi import *

username = 'admin'
ipAddress = '10.48.39.164'
OID = '1.3.6.1.4.1.14179.2.3.1.13.0'
authKey = 'Cisco123Cisco123'
privKey = 'Cisco123Cisco123'

errorIndication, errorStatus, errorIndex, varBinds = next(
    getCmd(SnmpEngine(),
           UsmUserData(username, authKey, privKey,
                       authProtocol=usmHMACSHAAuthProtocol,
                       privProtocol=usmAesCfb128Protocol),
           UdpTransportTarget((ipAddress, 161)),
           ContextData(),
           ObjectType(ObjectIdentity(OID)))
)

if errorIndication:
    print(errorIndication)
elif errorStatus:
    print('%s at %s' % (errorStatus.prettyPrint(),
                        errorIndex and varBinds[int(errorIndex) - 1][0] or '?'))
else:
    for varBind in varBinds:
        print(' = '.join([x.prettyPrint() for x in varBind]))
```

Integration with Third-Party Software (Grafana/PRTG Network Monitor/SolarWinds)

Cisco Prime Infrastructure offers the ability to easily monitor and configure multiple network devices, that include wireless controllers. Prime Infrastructure comes preloaded with all the OIDs and the integration with WLC simply consists of the addition of the WLC credentials to Prime. After the sync, it is possible to set alarms and push configuration templates for multiple wireless controllers at once.

On the other hand, Cisco WLC can also be integrated with multiple third-party monitoring solutions, as long as the OIDs are known. Programs like Grafana, PRTG Network Monitor, and SolarWinds server allow the MIBs or OIDs to be imported and values to be displayed in a user-friendly graph.

Monitoring servers can need to be tweaked to accommodate this integration. In the example shown in the image, the PRTG monitoring server is provided with the per-core CPU utilization OID which returns the string 0%/1%, 1%/1%, 0%/1%, 0%/1%. PRTG expects an integer value and raises an error.



'0%/1%, 1%/1%, 0%/1%, 0%/1%' is not a valid integer value

Table of Most Commonly Monitored OIDs

If you consider that MIBs present the data in non-user-friendly syntax, this table includes some of the most common object names and their OIDs the Cisco customers use.

Description	Object Name	OID	Expected Response
Overall CPU usage in %	agentCurrentCPUUtilization	1.3.6.1.4.1.14179.1.1.5.1.0	INTEGER: 0
Per core CPU usage	clsAllCpuUsage	1.3.6.1.4.1.9.9.618.1.4.1.0	STRING: 0%/1%, 0%/1%, 0%/1%, 0%/1%
RAM usage in %	clsSysCurrentMemoryUsage	1.3.6.1.4.1.9.9.618.1.8.6.0	Gauge32: 33
CPU temperature in °C	bsnSensorTemperature	1.3.6.1.4.1.14179.2.3.1.13.0	INTEGER: 76
Number of joined AP	clsSysApConnectCount	1.3.6.1.4.1.9.9.618.1.8.4.0	Gauge32: 2
Number of clients	clsMaxClientsCount	1.3.6.1.4.1.9.9.618.1.8.12.0	Gauge32: 0
Number of clients per WLAN	bsnDot11EssNumberOfMobileStations	1.3.6.1.4.1.14179.2.1.1.1.38.0	Counter32: 3 Counter32: 2