knowledge14®

April 27–May 1, 2014 • Moscone Center • San Francisco, CA

Discovery 201

Aleck Lin
Discovery Tech Lead
ServiceNow

Doug Schulze
Expert Discovery Guru
ServiceNow

servicenow®

© 2014 ServiceNow All Rights Reserved

Lab Prerequisites

- Skills needed for this lab are a core understanding of the ServiceNow Discovery technology, comprehension of TCP/IP protocols, IP networking and base understanding of the ServiceNow core platform
- JavaScript Knowledge

© 2014 ServiceNow All Rights Reserved

Agenda

Discovery Basics

SNMP Basics

Discovery of a device using SNMP – classify, custom probes and sensors, System OID

Discovery identifiers

Q&A

© 2014 ServiceNow All Rights Reserved

Disco Stu

Before becoming a fan of disco, Stu was known as Nautical Stu due to his career as a Sea Captain, before giving it up for his disco.

In 2002 Disco Stu revealed that he's aware disco is dead, and that he does not even like it anymore. He expresses worry that he has let disco define him as a person and fears becoming a "one note guy". – simpsons.wikia.com

That was until… SeviceNow gave him new purpose



© 2014 ServiceNow All Rights Reserved

Deploying a MID Server

- MID Servers (Linux based) have been pre-deployed on your instance for the convenience of this lab.

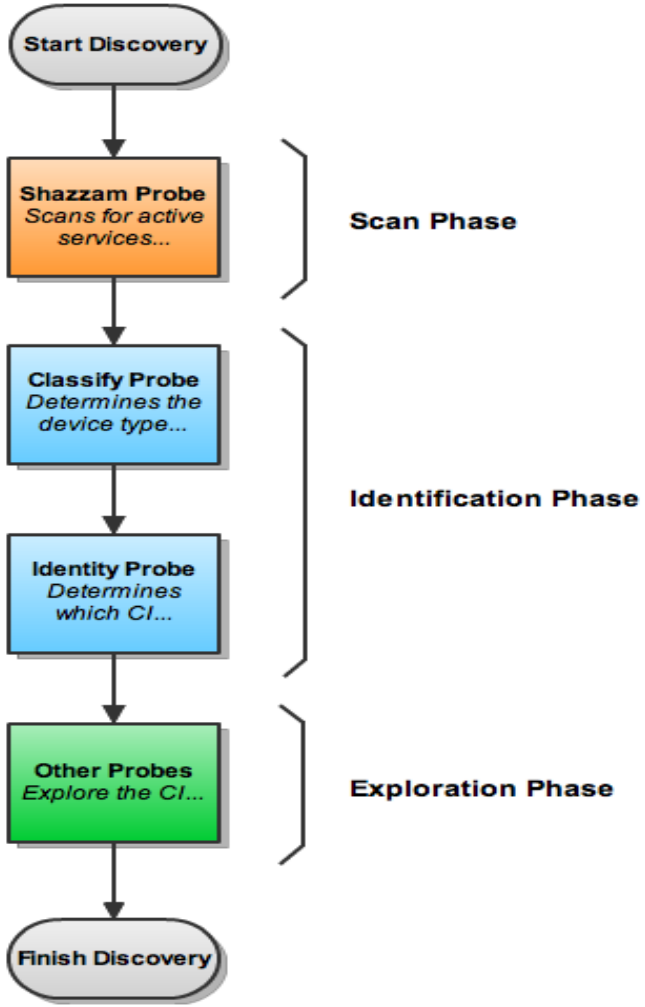


© 2014 ServiceNow All Rights Reserved

How Discovery Works

- Four stages of Discovery
  - Port Scanning (What's out there?)
  - Classifying (What is it?)
  - Identifying (Do we have it already?)
  - Exploring (Update it!)





© 2014 ServiceNow All Rights Reserved

SNMP Basics

- SNMP stands for Simple Network Management Protocol
- Typically used to for network devices, such as routers, switches, printers, UPS, servers…
- SNMP devices expose management data as variables (identified by OIDs), which are organized in hierarchies. These hierarchies and other metadata (such as type and description of the variable) are described by MIBs (Management Information Bases)

© 2014 ServiceNow All Rights Reserved

SNMP Basics

- MIB (Management Information Base) typically consists of many managed objects (aka variables).
- There are two types of managed object
  - Scalar object, which defines single object instance
  - Tabular object, which defines multiple related object instances that are grouped in MIB tables
- OIDs, or Object Identifiers uniquely identify managed objects in a MIB hierarchy
- www.oid-info.com

© 2014 ServiceNow All Rights Reserved

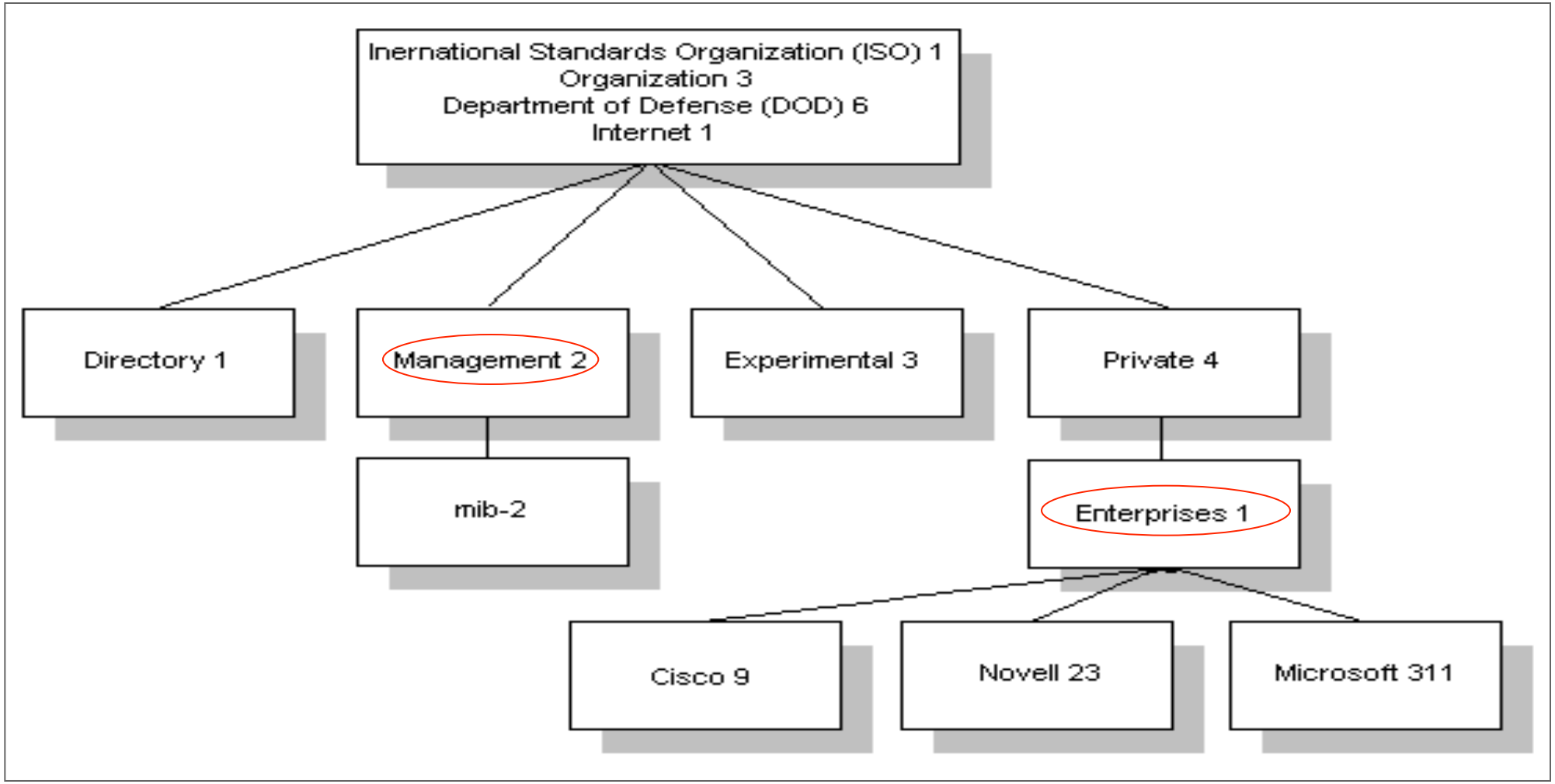
Understanding OID Structure

- Typically assume ISO(1).Identified-organization(3).dod(6).internet(1)
  - Directory (1)
  - Mgmt(2)
  - Experimental(3)
  - Private(4)
  - Security(5)
  - snmpV2(6)
  - Mail(7)
  - Features(8)



© 2014 ServiceNow All Rights Reserved

Understanding OID Structure



© 2014 ServiceNow All Rights Reserved

What kind of device is it?

- "SNMP MIB-2" MIB (RFC 1213)
  ISO(1).Identified-organization(3).dod(6).internet(1).Mgmt(2).mib-2(1).system(1)
  - sysDescr(1) – Scalar object
  - sysObjectID(2)
  - sysUpTime(3)
  - sysContact(4)
  - sysName(5)
  - sysLocation(6)
  - sysServices(7)
  - sysORLastChange(8)
  - sysORTable(9) – Tabular object

© 2014 ServiceNow All Rights Reserved

Understanding OID Structure

- To access an object, for example, the sysDescr object. The OID would be “1.3.6.1.2.1.1.1”
- Using command line tools to retrieve the object value such as snmpwalk, snmpget
- For example,
  snmpwalk -v 2c -c public 50.18.243.130 1.3.6.1.2.1.1.1
  would return result such as
  “.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 = STRING: Cisco PIX Firewall 6.0”

© 2014 ServiceNow All Rights Reserved

Example of a Cisco PIX 520 Firewall

```
monster-MacBook:~ aleck.lin$ snmpwalk -Of -v 2c -c public 10.0.100.240 1.3.6.1.2.1.1
.iso.org.dod.internet.mgmt.mib-2.system.sysDescr.0 = STRING: Cisco PIX Firewall 6.0
.iso.org.dod.internet.mgmt.mib-2.system.sysObjectID.0 = OID: .iso.org.dod.internet.private.enterprises.9.1.392
.iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.sysUpTimeInstance = Timeticks: (129600000) 15 days, 0:00:00.00
.iso.org.dod.internet.mgmt.mib-2.system.sysContact.0 = STRING: Jalasoft
.iso.org.dod.internet.mgmt.mib-2.system.sysName.0 = STRING: JalaFWPix525 - 10.0.100.240 - V2*
.iso.org.dod.internet.mgmt.mib-2.system.sysLocation.0 = STRING: Undisclosed Location
.iso.org.dod.internet.mgmt.mib-2.system.sysServices.0 = INTEGER: 64
.iso.org.dod.internet.mgmt.mib-2.system.sysORLastChange.0 = Timeticks: (253) 0:00:02.53
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.1 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpMIB
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.2 = OID: .iso.org.dod.internet.mgmt.mib-2.ifMIB
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.3 = OID: .iso.org.dod.internet.mgmt.mib-2.tcpMIB
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.4 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpFrameworkMIB
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.5 = OID: .iso.org.dod.internet.private.enterprises.9
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.6 = OID: .iso.org.dod.internet.private.enterprises.9.9.48
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.7 = OID: .iso.org.dod.internet.private.enterprises.9.9.147
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.8 = OID: .iso.org.dod.internet.mgmt.mib-2.ianaifType
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.9 = OID: .iso.org.dod.internet.private.enterprises.9.18.2.1
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.10 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpTargetMIB
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.11 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpCommunityMIB
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORID.12 = OID: .iso.org.dod.internet.snmpV2.snmpModules.snmpNotificationMIB
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.1 = STRING: The MIB module for SNMPv2 entities.
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.2 = STRING: The MIB module to describe generic objects for network i
of MIBIIs ifTable and incorporates the extensions defined in RFC 1229.
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.3 = STRING: The MIB module for managing TCP implementations.
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.4 = STRING: The SNMP Management Architecture MIB
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.5 = STRING: The Structure of Management Information for theCisco ent
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.6 = STRING: MIB module for monitoring memory pools
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.7 = STRING: MIB module for monitoring Cisco Firewalls.
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.8 = STRING: The MIB module which defines the IANAifType textual conv
 object defined in MIBIIs ifTable.
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.9 = STRING: The Cisco QOS Policy PIB for provisioning QOS policy.
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.10 = STRING: This MIB module defines MIB objects which provide mecha
an SNMP entity for the generation of SNMP messages.
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.11 = STRING: This MIB module defines objects to help support coexist
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORDescr.12 = STRING: This MIB module defines MIB objects which provide mecha
an SNMP entity for the generation of notifications.
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORUpTime.1 = Timeticks: (249) 0:00:02.49
.iso.org.dod.internet.mgmt.mib-2.system.sysORTable.sysOREntry.sysORUpTime.2 = Timeticks: (252) 0:00:02.52
```

© 2014 ServiceNow All Rights Reserved

Lab Exercise 1 - Discover the Firewall!

- In this lab, we will learn how to classify a SNMP device that Discovery does not find out of the box
- A firewall device is available for the purpose of this lab



© 2014 ServiceNow All Rights Reserved

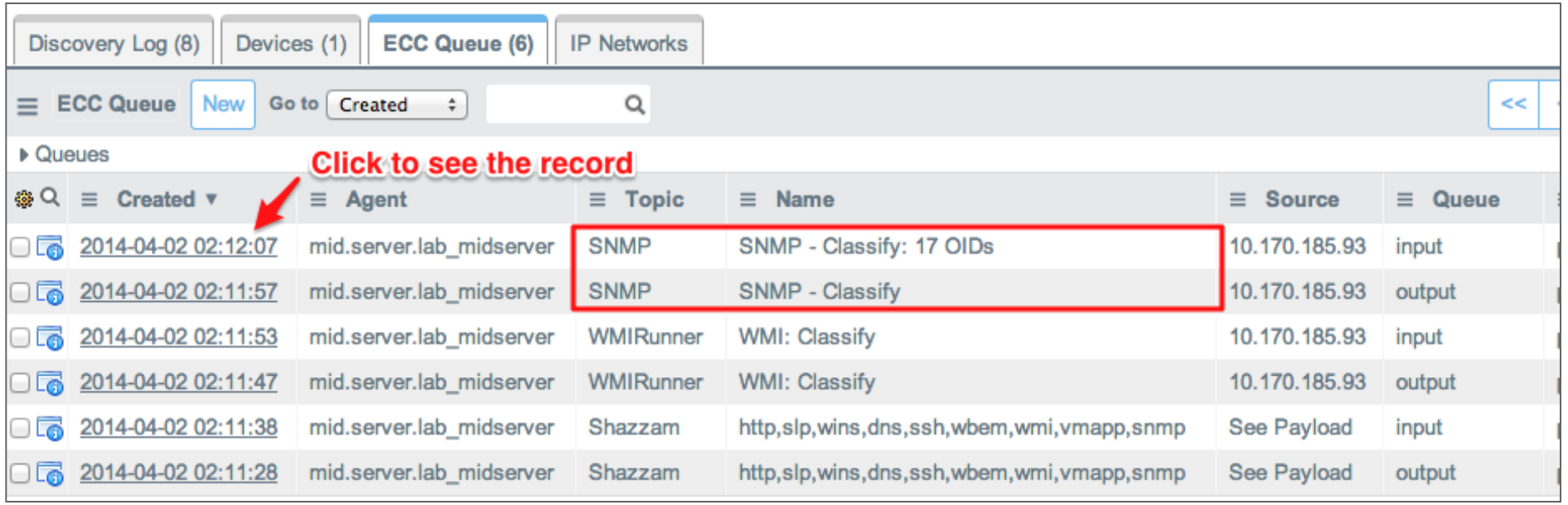
Discover the Firewall!

- Run a Discovery against the IP Firewall device
- Identify how to classify against the device
- Create a table type for IP Firewall device

© 2014 ServiceNow All Rights Reserved

Discover the Firewall!

- Discovery found SNMP port and launched SNMP classify probe, but stopped afterwards because it couldn't find any match in the SNMP classifications.
- Let's take a look at what we found by drilling into the SNMP – Classify input ECC queue record.

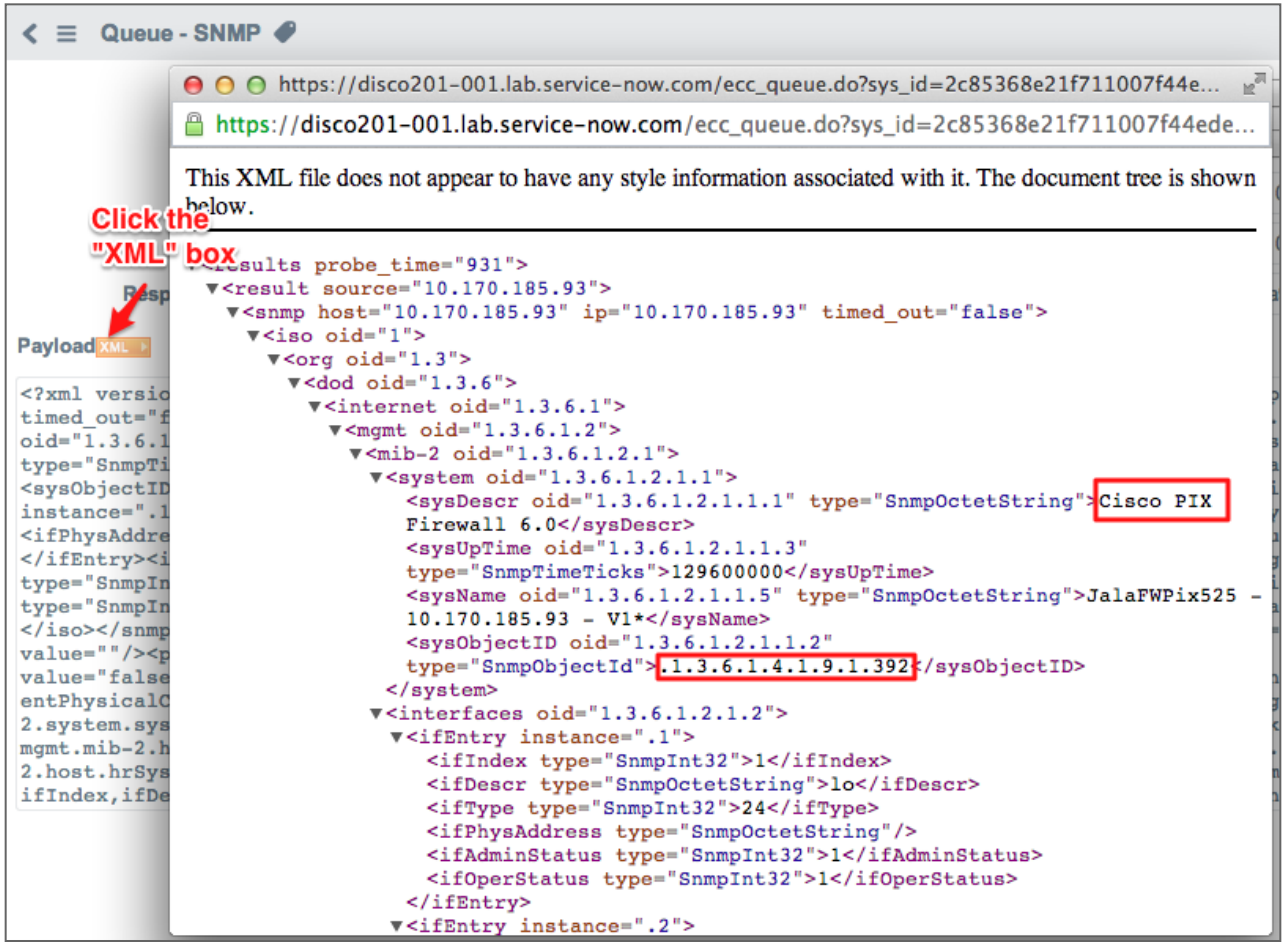


Created	Agent	Topic	Name	Source	Queue
2014-04-02 02:12:07	mid.server.lab_midserver	SNMP	SNMP - Classify: 17 OIDs	10.170.185.93	input
2014-04-02 02:11:57	mid.server.lab_midserver	SNMP	SNMP - Classify	10.170.185.93	output
2014-04-02 02:11:53	mid.server.lab_midserver	WMIRunner	WMI: Classify	10.170.185.93	input
2014-04-02 02:11:47	mid.server.lab_midserver	WMIRunner	WMI: Classify	10.170.185.93	output
2014-04-02 02:11:38	mid.server.lab_midserver	Shazzam	http,slp,wins,dns,ssh,wbem,wmi,vmapp,snmp	See Payload	input
2014-04-02 02:11:28	mid.server.lab_midserver	Shazzam	http,slp,wins,dns,ssh,wbem,wmi,vmapp,snmp	See Payload	output

© 2014 ServiceNow All Rights Reserved

Discover the Firewall!

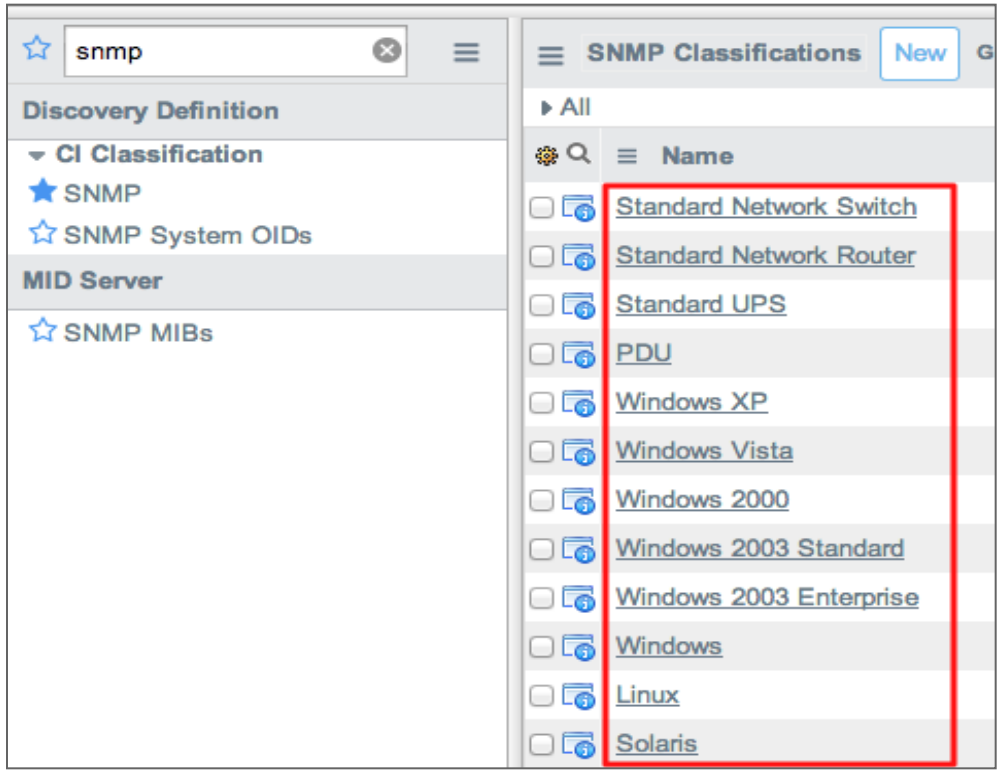
- We can see basic info from the sysDescr that it is a “Cisco PIX PIX Firewall 6.0”



© 2014 ServiceNow All Rights Reserved

Discover the Firewall!

- We need a SNMP classifier for the firewall device.
- If Click into “SNMP” module under the “Discovery Definition” application, we’ll notice that firewall is not on the list at all.





© 2014 ServiceNow All Rights Reserved

Discover the Firewall!

- Before creating a new SNMP classification for the firewall, we'll need to create a firewall table to properly store the CI record.
- We'll create a "u_cmdb_ci_ip_firewall" that will extend the cmdb_ci_netgear table.
- The "u_" is advised so that your table will never clash with the ones that ServiceNow create (or will create).
- Let's switch gear to create and the table first and we'll come back to create the new classification for firewalls.

© 2014 ServiceNow All Rights Reserved

Discover the Firewall!

- Create an IP Firewall SNMP classifier
- Run Discovery
- See what we have discovered
- Personalize the form layout



© 2014 ServiceNow All Rights Reserved

Exercise 2 - Custom Probes and Sensors

- In the following section we'll talk about how to create custom SNMP probes and sensors
- The goal is to help you figure out how you can leverage the Discovery tool to gather data that is meaningful to your organization
- Even though SNMP is the primary focus in this exercise, the general principle and techniques can be applied to writing other probes and sensors

© 2014 ServiceNow All Rights Reserved

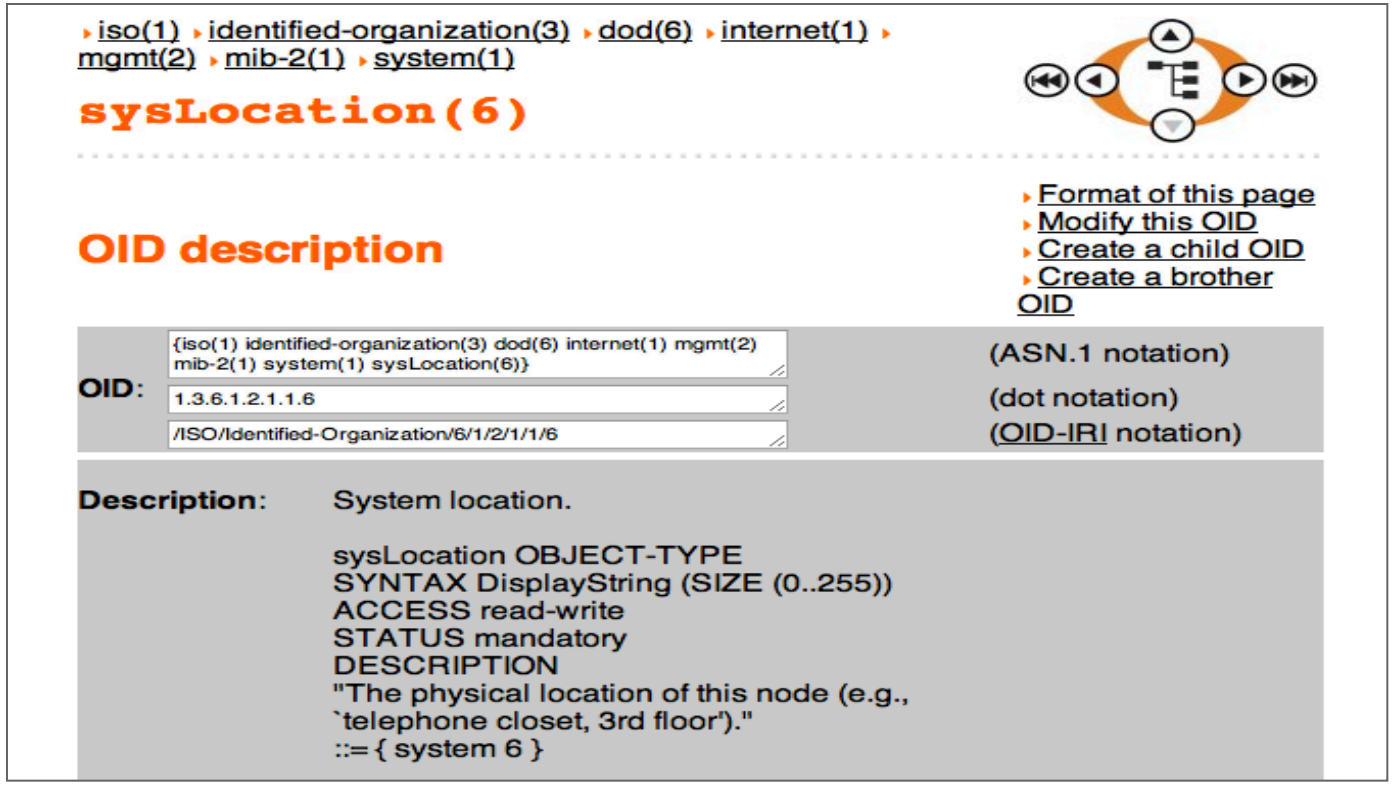
Exercise 2.1 - Gather Scalar OID Objects

- Create a SNMP probe
  - iso.org.dod.internet.mgmt.mib-2.system.sysLocation
- Test probe
  - Verify data from the OID
- Create a corresponding SNMP sensor
- Add it to the IP Firewall SNMP classifier
- Run Discovery!

© 2014 ServiceNow All Rights Reserved

Exercise 2.1 - Gather Scalar OID Objects

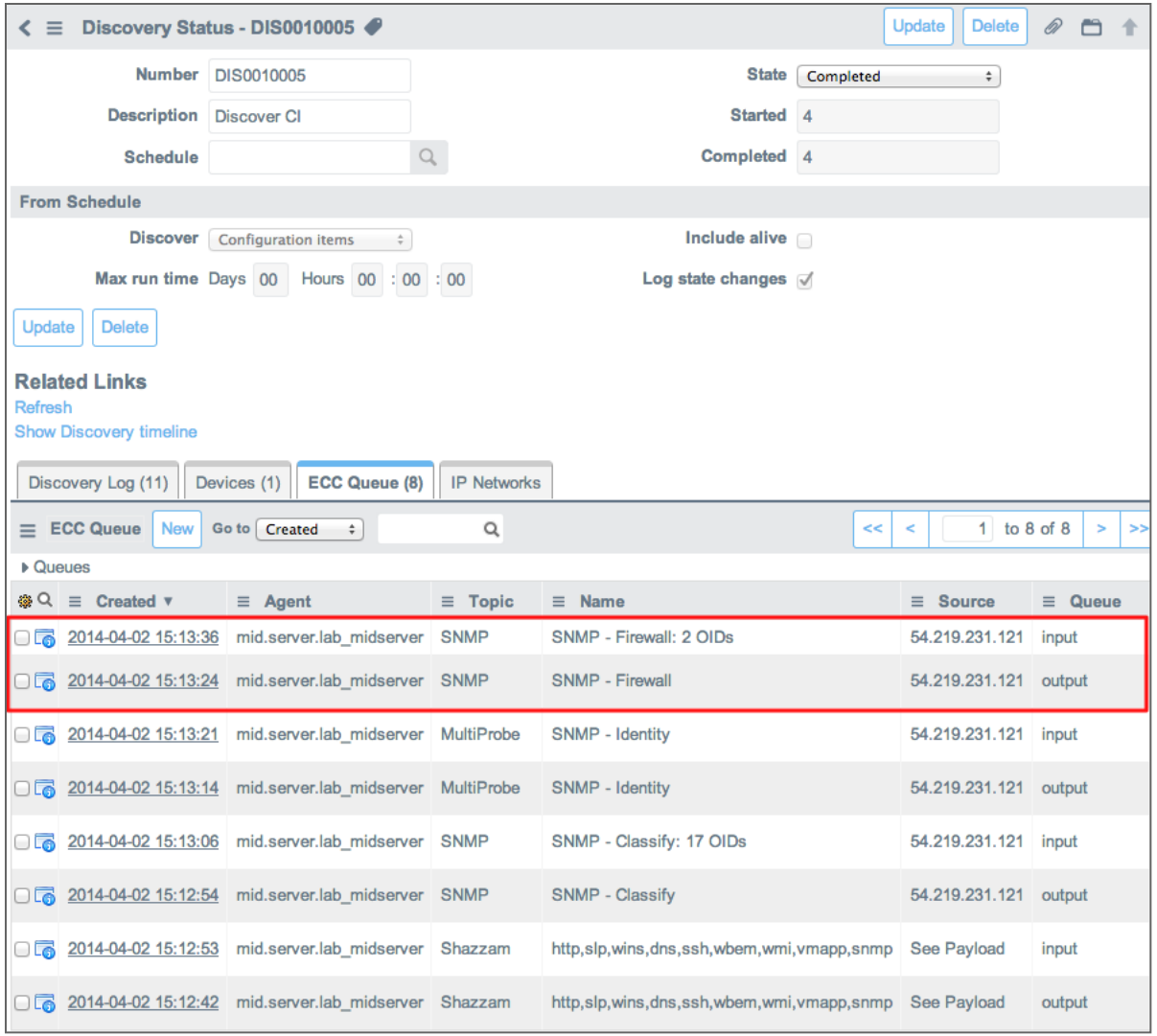
- This screen shot from www.oid-info.com describes the particular OID we have entered for our new probe.



© 2014 ServiceNow All Rights Reserved

Exercise 2.1 - Gather Scalar OID Objects

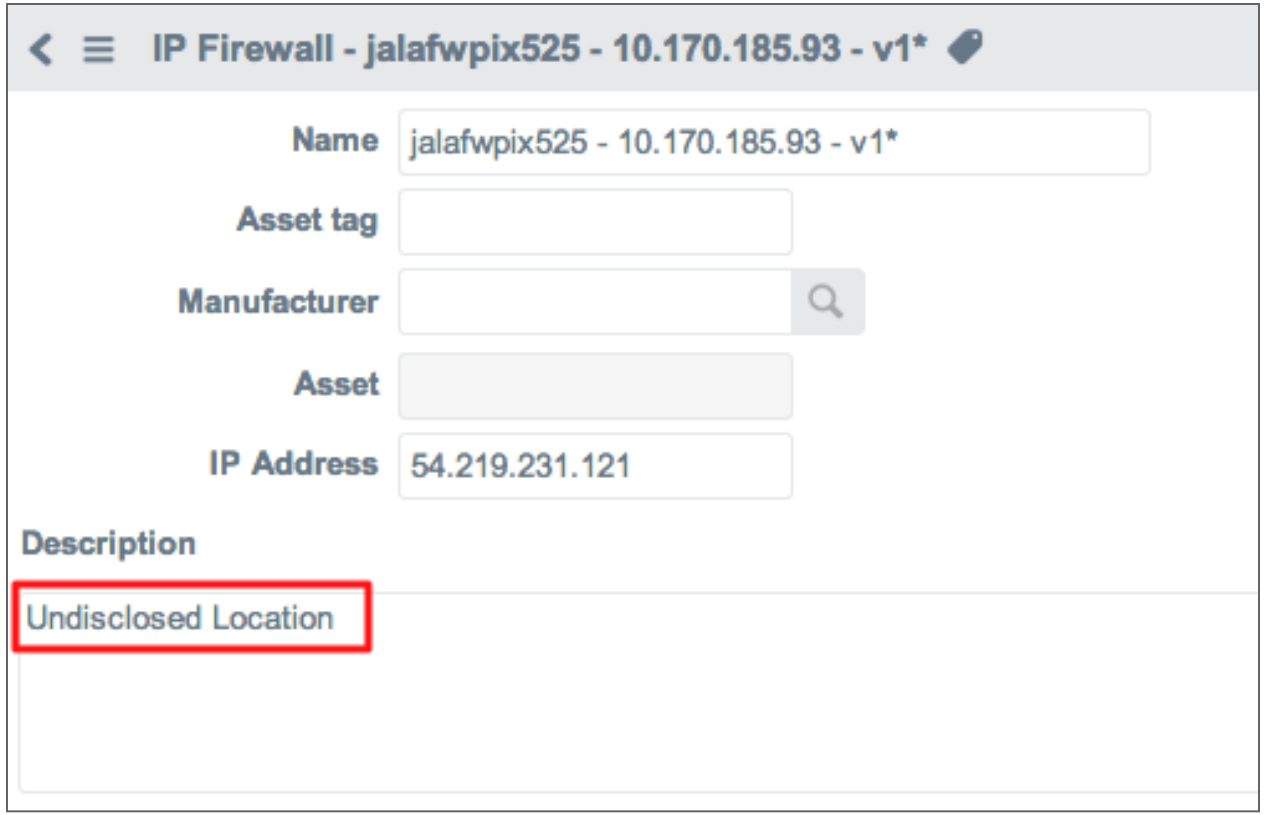
- Observe that the Discovery run indeed has launched the new probe
- Go to the “Devices” related list and drill into the CI



© 2014 ServiceNow All Rights Reserved

Exercise 2.1 - Gather Scalar OID Objects

- Observe that the value of the OID has indeed been added to the “Short description” field



© 2014 ServiceNow All Rights Reserved

Lab Challenge – Add Additional Info From Other OIDs

- Add “iso.org.dod.internet.mgmt.mib-2.system.sysContact” to the “asset_tag” field.
  - Add the new OID to the probe we had created
  - Modify the existing sensor to process the information
  - Re-run discovery and see the result!



© 2014 ServiceNow All Rights Reserved

Custom Probes and Sensors – Part 2

- For discovering SNMP MIB objects, we've covered how to get values from scalar objects. In part 2, we'll learn about how to discover tabular objects.

- From Discovery's perspective, we'll explore how to launch a probe for getting OID table and its values and how to process the results in the sensor

© 2014 ServiceNow All Rights Reserved

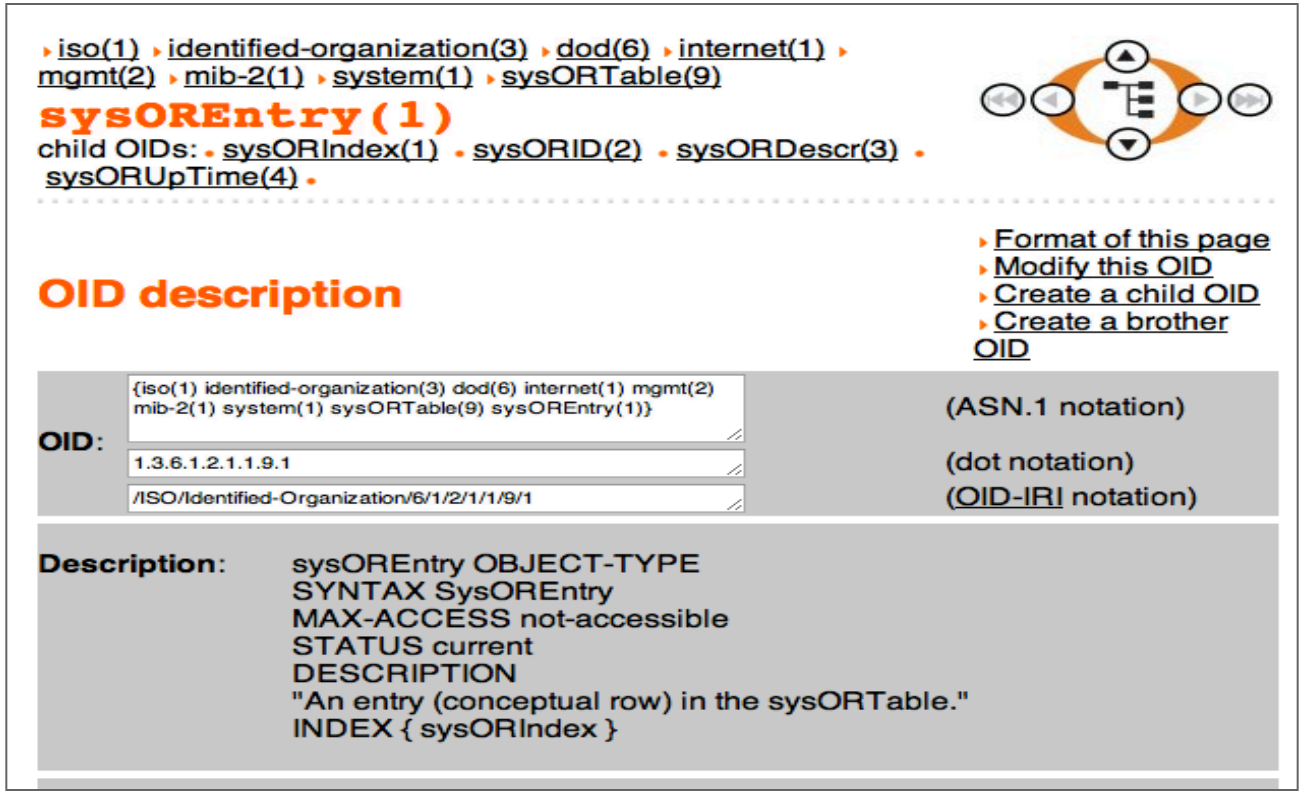
Exercise 2.2 - Gather Tabular OID objects

- Add the new tabular OID to the probe
  - iso.org.dod.internet.mgmt.mib-2.system.sysORTable
  - sysORID
  - sysORDescr
- Test probe
  - Verify data from the OID
- Modify the corresponding SNMP sensor
- Run Discovery!

© 2014 ServiceNow All Rights Reserved

Exercise 2.2 - Gather Tabular OID objects

- This screen shot from www.oid-info.com describes the particular OID Table we have entered for our new probe.



© 2014 ServiceNow All Rights Reserved

Exercise 2.2 - Gather Tabular OID objects

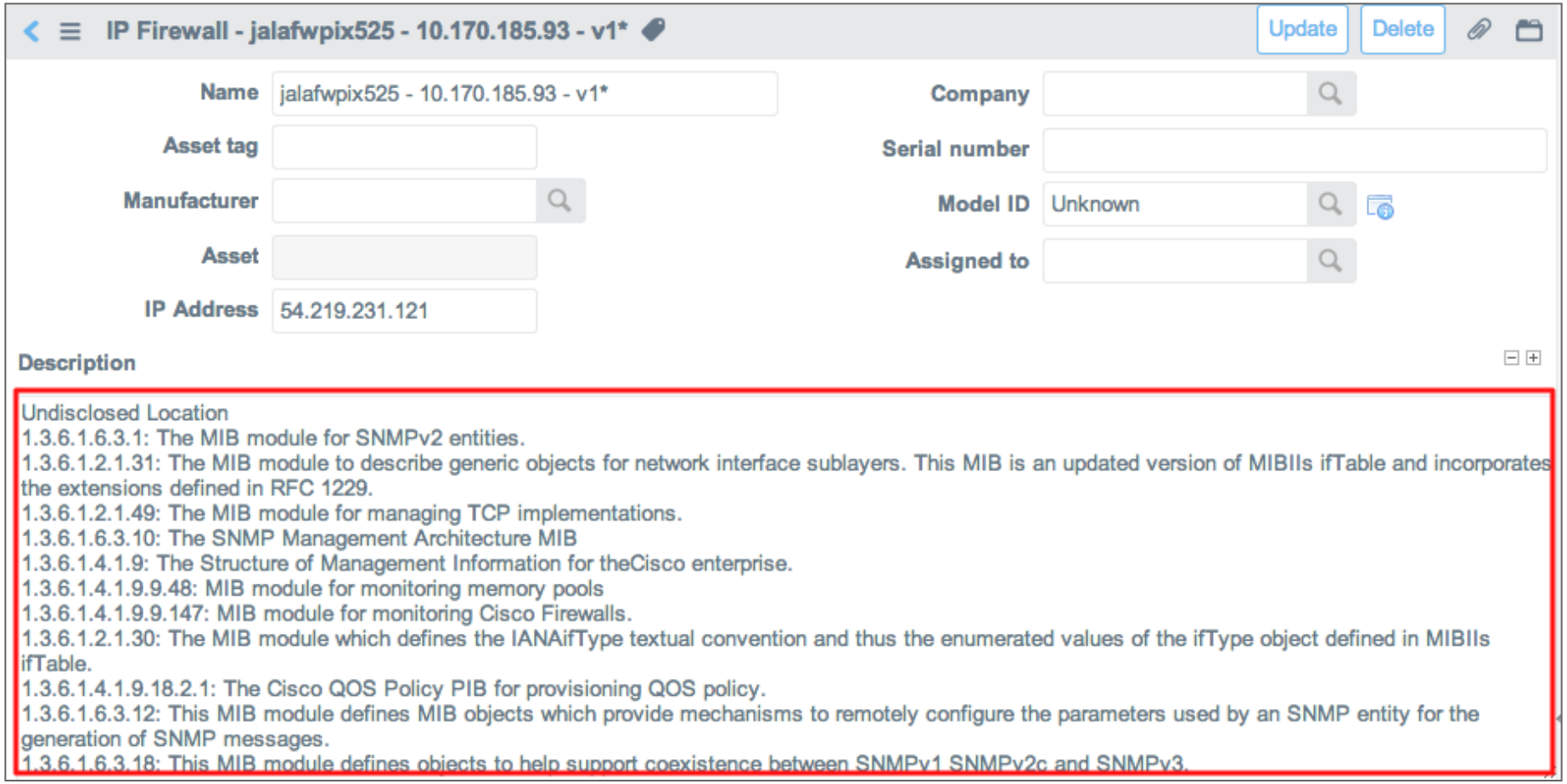
- Observe that the Discovery run indeed has launched the new probe
- Notice there are 26 OIDs the probe brings back now



© 2014 ServiceNow All Rights Reserved

Exercise 2.2 - Gather Tabular OID objects

- Observe that we have added the two values (“sysORID” and “sysORDDescr”) to the short description field



© 2014 ServiceNow All Rights Reserved

Classify a SNMP Device

The trick to classify a particular SNMP device is to use something that is generic to the type of device.

- What are possibilities?
  - sysdescr
  - Object ID
  - An OID that that is mandatory from the RFC MIB



© 2014 ServiceNow All Rights Reserved

Classify an SNMP Device

- In our example, the system OID to use is “1.3.6.1.4.1.9.1.392”

https://disco201-001.lab.service-now.com/ecc_queue.do?sys_id=14dda19a213b11007f44e...

https://disco201-001.lab.service-now.com/ecc_queue.do?sys_id=14dda19a213b11007f44ede...

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<results probe_time="263">
  ▼<result source="54.219.231.121">
    ▼<snmp host="54.219.231.121" ip="54.219.231.121" timed_out="false">
      ▼<iso oid="1">
        ▼<org oid="1.3">
          ▼<dod oid="1.3.6">
            ▼<internet oid="1.3.6.1">
              ▼<mgmt oid="1.3.6.1.2">
                ▼<mib-2 oid="1.3.6.1.2.1">
                  ▼<system oid="1.3.6.1.2.1.1">
                      <sysDescr oid="1.3.6.1.2.1.1.1" type="SnmpOctetString">Cisco PIX
                      Firewall 6.0</sysDescr>
                      <sysUpTime oid="1.3.6.1.2.1.1.3"
                      type="SnmpTimeTicks">129600000</sysUpTime>
                      <sysName oid="1.3.6.1.2.1.1.5" type="SnmpOctetString">JalaFWPix525 -
                      10.170.185.93 - V1*</sysName>
                      <sysObjectID oid="1.3.6.1.2.1.1.2"
                      type="SnmpObjectId">.1.3.6.1.4.1.9.1.392</sysObjectID>
                    </system>
                  ▼<interfaces oid="1.3.6.1.2.1.2">
                    ▼<ifEntry instance=".1">
                        <ifIndex type="SnmpInt32">1</ifIndex>
                        <ifDescr type="SnmpOctetString">lo</ifDescr>
                        <ifType type="SnmpInt32">24</ifType>
                        <ifPhysAddress type="SnmpOctetString"/>
                        <ifAdminStatus type="SnmpInt32">1</ifAdminStatus>
                        <ifOperStatus type="SnmpInt32">1</ifOperStatus>
                      </ifEntry>
```

© 2014 ServiceNow All Rights Reserved

Lab Exercise 3 – Classify using System OID

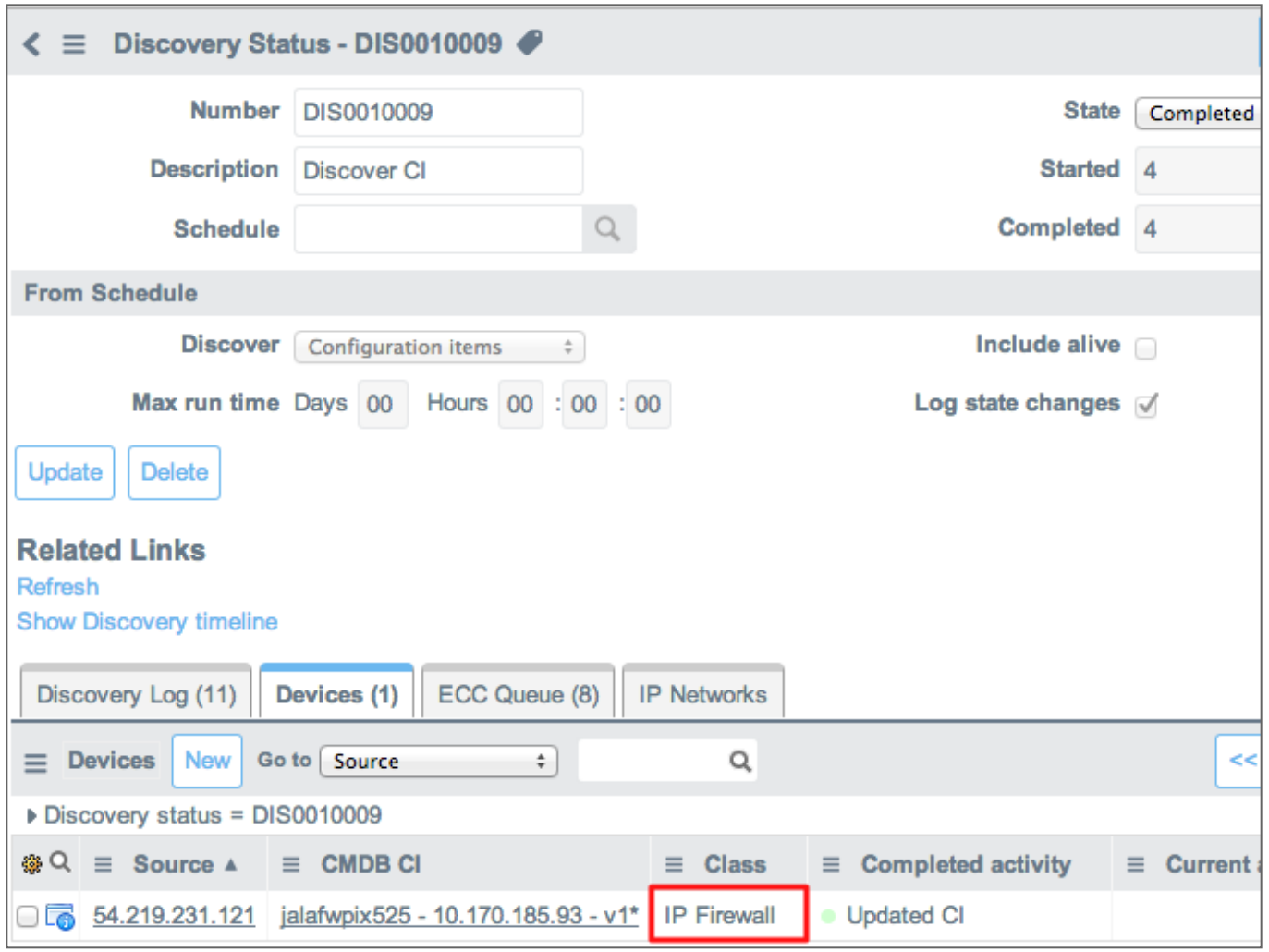
- Identify the system OID
- Create a new System OID classifier
- Run Discovery!



© 2014 ServiceNow All Rights Reserved

Classify an SNMP Device

- Observe that the Discovery has again successfully classified the device using the “system OID”



© 2014 ServiceNow All Rights Reserved

Discovery Identifiers

- CI Identifiers are a set of rules that allows Discovery to determine if a discovered device is already in the CMDB based on discovered information such as “serial numbers”, “hostname” and “mac address”.

- Its goal is to correctly determine the identity of a device and prevent the creation of duplicate CIs.



© 2014 ServiceNow All Rights Reserved

Discovery Identifiers

- How to configure them
- What is the right for you
- How to customize the identifiers
- http://wiki.servicenow.com/index.php?title=Discovery_Identifiers



© 2014 ServiceNow All Rights Reserved

Discovery Identifiers

- How does it work? What are the rules?
  - Discovery runs through each identifier to find a match based on its matching algorithm.
  - The identifier chain will stop as soon as an identifier finds one and only one match.
  - The identifier chain will not stop if there's no match or multiple matches.
  - If all identifiers fail to match and there are no multiple matches, a new CI will be created provided there's matchable info.
  - If all identifiers fail to match and there are multiple matches. Discovery logs duplicate CIs.

© 2014 ServiceNow All Rights Reserved

Discovery Identifiers

- Order matters!
  - Serial number(s)
  - Name
  - Network adapter(s) – IP and MAC address

© 2014 ServiceNow All Rights Reserved

Discovery Identifiers

- Understanding the OOB identifiers
- What works for your environment as the unique identifier for your devices?

CI Identifiers | New | Go to Order | 1 to 14 of 14

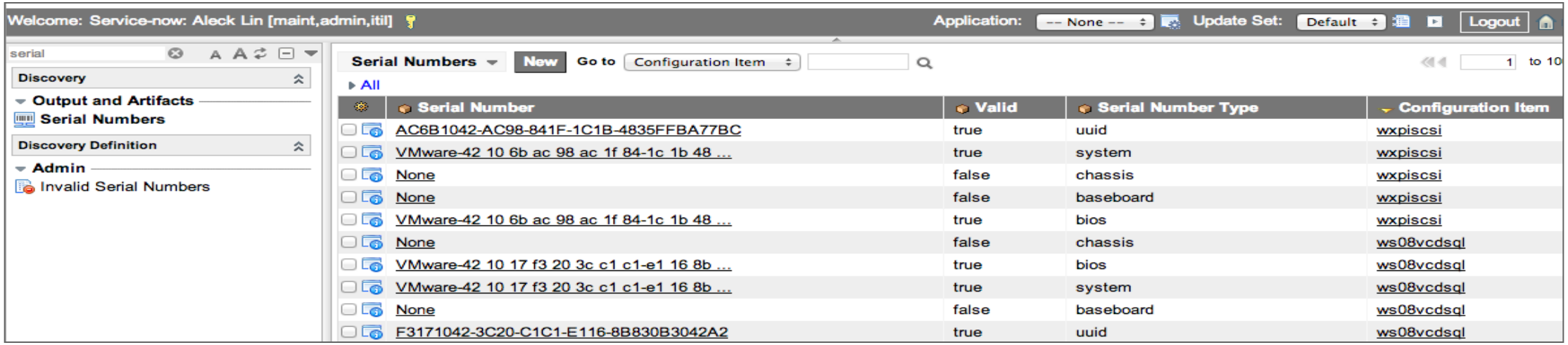
All

Name	Description	Active	Order	Applies to
ESX UUID	Identify ESX servers based on VMWare UUID.	true	900	cmdb_ci_esx_server
Serial Number Table & Class Name	Identify CIs in the CMDB based on serial number(s) in the serial number table and sys_class_name.	true	1,000	cmdb_ci_hardware
Serial Number & Class Name	Identify CIs in the CMDB based on the serial_number field and matching sys_class_name.	true	1,050	cmdb_ci_hardware
Name & Class Name	Identify CIs in the CMDB based on name field and matching sys_class_name.	true	1,100	cmdb_ci_hardware
Network	Identify CIs in the CMDB based on IPs and MAC Address(es) in the network adapter table.	true	1,200	cmdb_ci_hardware
Network & Class Name	Identify CIs in the CMDB based on the mac_address field and the ip_address field and matching sys_class_name.	false	2,200	cmdb_ci_hardware
MAC Address & Class Name	Identify CIs in the CMDB based on the mac_address field and matching sys_class_name.	false	2,300	cmdb_ci_hardware
IP Address & Class Name	Identify CIs in the CMDB based on the ip_address field and the matching sys_class_name.	false	2,400	cmdb_ci_hardware
MACAddress	Identify CIs in the CMDB based on MAC Address(es) in the network adapter table.	false	3,000	cmdb_ci_hardware
Generic Serial Number	Identify CIs in the CMDB based on serial_number field.	false	4,000	cmdb_ci_hardware
Generic Network	Identify CIs in the CMDB based on the mac_address field and the ip_address field.	false	4,100	cmdb_ci_hardware
Generic Name	Identify CIs in the CMDB based on name field.	false	4,200	cmdb_ci_hardware
Generic MAC Address	Identify CIs in the CMDB based on MAC address field.	false	4,300	cmdb_ci_hardware
Generic IP Address	Identify CIs in the CMDB based on the ip_address field.	false	4,400	cmdb_ci_hardware

© 2014 ServiceNow All Rights Reserved

Discovery Identifiers

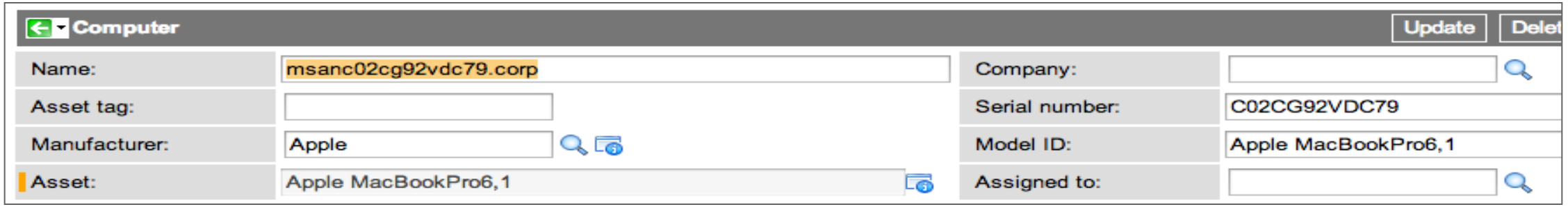
- Serial number table identifier
- Identifies by matching the serial numbers in the serial number table (cmdb_serial_number) that belong to a CI



© 2014 ServiceNow All Rights Reserved

Discovery Identifiers

- Serial number field identifier
- Identifies by matching the serial number field of the CI



© 2014 ServiceNow All Rights Reserved

Discovery Identifiers

- The applies_to class type matters!
- “cmdb_ci_hardware” is the default, which includes child table such as
  - cmdb_ci_computer
  - cmdb_ci_printer
  - cmdb_ci_netgear
  - cmdb_ci_ups
  - For more, see “Tables” modules under “System Definition” menu

© 2014 ServiceNow All Rights Reserved

Lab Exercise 4 - Custom Identifiers

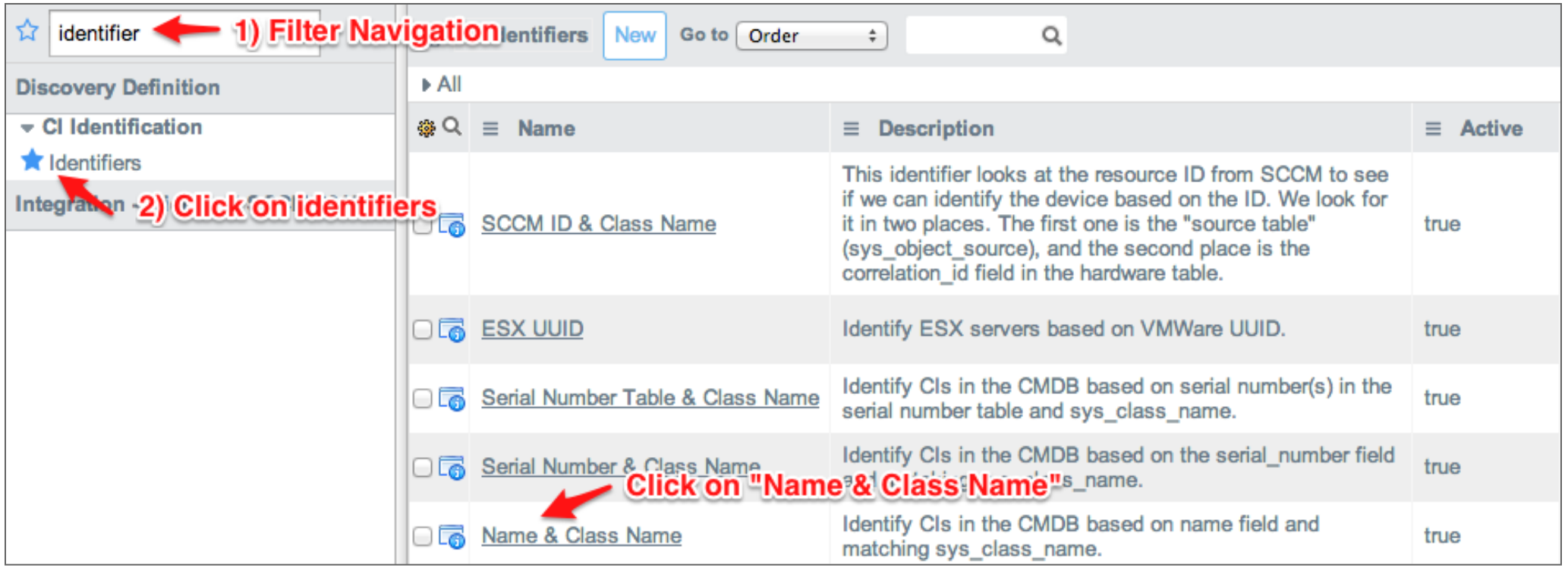
- Scenario: My Solaris global zone and its non-global zones return the same serial number
- Solution: Have a separate identifier just for Solaris using the name value



© 2014 ServiceNow All Rights Reserved

Custom Discovery Identifiers

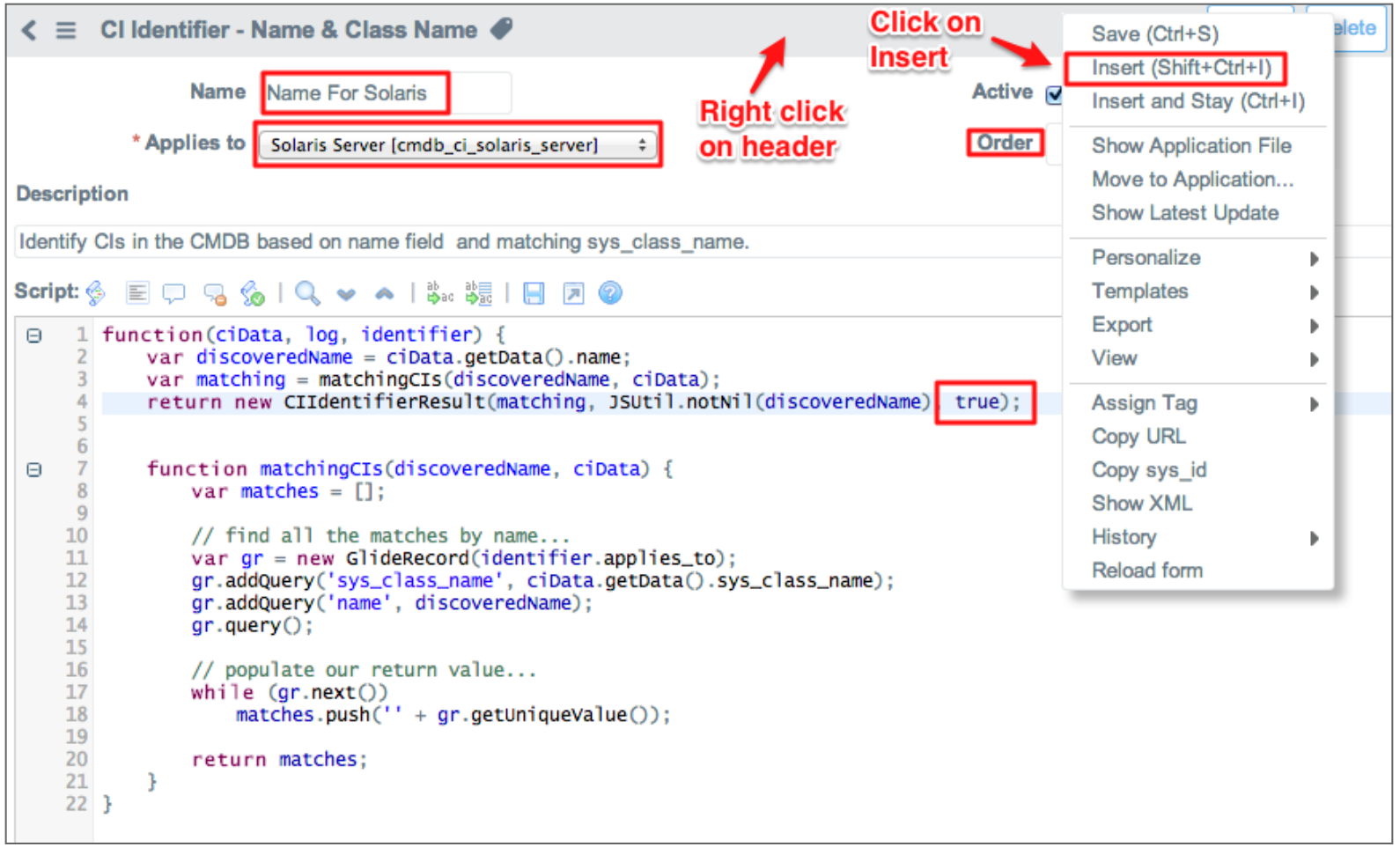
- Go to “Identifiers” module under “Discovery Definition” application and drill into the “Name & Class Name” identifier record.



© 2014 ServiceNow All Rights Reserved

Custom Discovery Identifiers

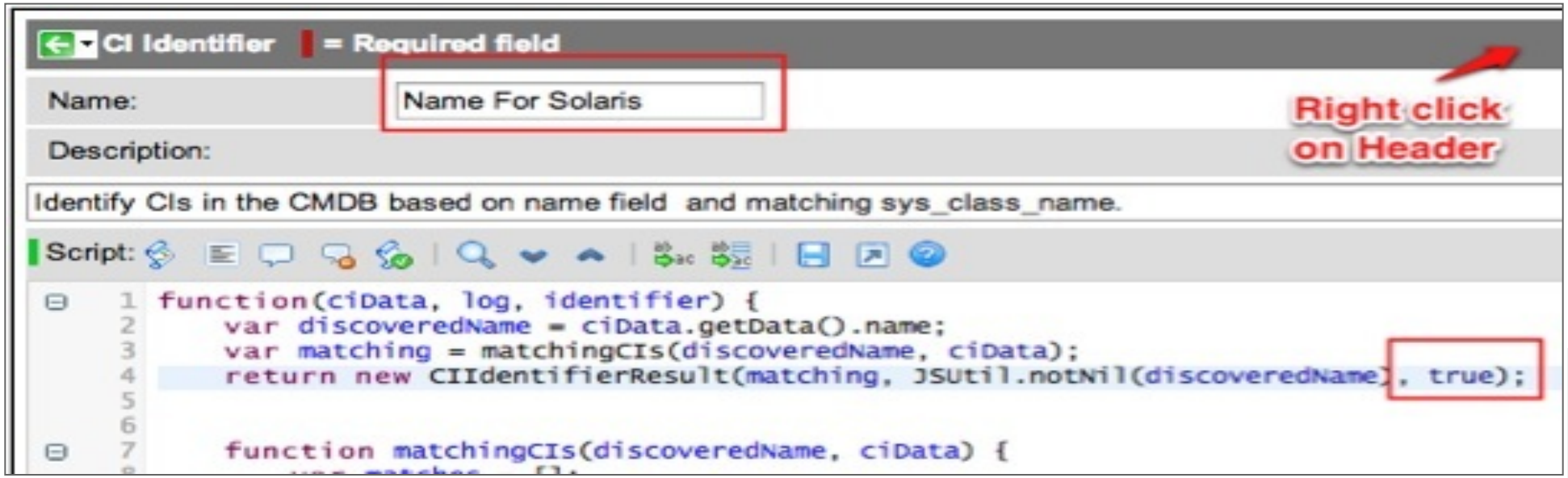
- Modify the following…
  - Name field: for human readability
  - Applies to: Solaris Server (cmdb_ci_solaris_server)
  - Order: 100 (so it runs first)
- Right click on the header and click “Insert”

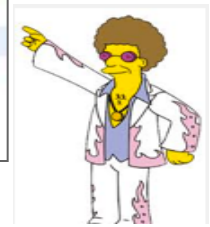


© 2014 ServiceNow All Rights Reserved

Custom Discovery Identifiers

- Make sure the script is modified so the CIIdentifierResult has a third argument of “true”
- `CIIdentifierResult(matching, JSUtil.nil(discoveredName), true);`





© 2014 ServiceNow All Rights Reserved

Custom Discovery Identifiers

- Why do we need the third argument?
- The third argument will dictate whether the other identifiers continue to run after the current identifier (based on order)
- It's false by default

© 2014 ServiceNow All Rights Reserved

Lab Challenge - Create your own identifier for the IP Firewall devices

- Make an identifier using the name field for the IP Firewalls and make sure it will stop going through the other identifiers if the identifier fails to find a match.
- Run a Discovery and verify in the discovery logs that it indeed stops after the identifier.
- Tip: Change the existing Firewall CI name to ensure the identifier will fail.

© 2014 ServiceNow All Rights Reserved

Top Takeaways

1 Creating a probe

2 Creating a sensor

3 Know your identifiers

© 2014 ServiceNow All Rights Reserved

Questions?
(Disco Stu's secret identity)

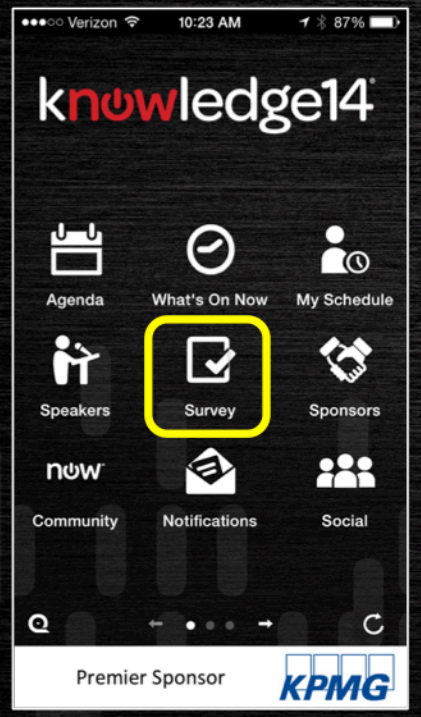


www.yearbookyourself.com



© 2014 ServiceNow All Rights Reserved

Tell us what you think!



Click on the survey link in your Knowledge14 mobile app. Help us make Knowledge 15 even more valuable!

Thank You

Aleck Lin

Discovery Tech Lead

ServiceNow

Aleck.Lin@servicenow.com

Doug Schulze

Senior Discovery Guru

ServiceNow

Doug.Shulze@servicenow.com

Want a copy of this deck?

Sign into the BuzzHub to unlock all the slides and lab guides from Knowledge14 sessions

