

Large Scale Discovery

White Paper

Introduction

Discovery processes are usually not considered to be time critical. However, when addressing large scale organizations, even discovery processes may take too long. This paper explains the problem, and describes a case study where this is relevant.

Case Study

The Need

- A very large tel-co has a huge IP network.
- The address ranges of its IP network exceeds one million (1,000,000) IP addresses, distributed over a list of class B sub-net addresses.
- Of the more than one million addresses, there are approximately 150,000 nodes that actually exist on the network.
- Of the approximately 150,000 existing nodes, there are approximately 20% that do not exist in the users database of the tel-co.

The tel-co's need is to discover all the IPs that are in its network. The requested information for each such node is:

- The IP address of the node
- The SNMP community that the node responds to (if any)
- An indication if the node has a telnet port open
- For each SNMP responding device, the following SNMP fields are needed:
 - SysObjectId
 - SysName
 - SysContact

- SysLocation
- SysDescription

Scale Issues

The huge number of IPs to check and nodes to SNMP query, prevent us from using trivial methods of synchronous polling in order to perform the required discovery, especially if we need to run the discovery more than once.

To illustrate the issue, consider the following:

If we assume that we need 10 seconds to decide whether a node responds to ping (3 retries * 3 seconds for timeout), then it would take 10,000,000 seconds (which is almost 115 days) to check which one of a 1,200,000 IPs network is responding to ping, if we ping the devices synchronously.

To query the responding SNMP community from all the devices will also take a few days, even if most of the devices are answering relatively fast. The same goes for the telnet port scanning and also for the actual retrieving of the SNMP data from the SNMP responding devices.

Such a long discovery period is not reasonable, especially if we want to run the discovery multiple times (at least more than once, to be sure that we did not miss a device that was powered off during the first check).

There are two ways to approach the problem, so that the discovery can be completed in a reasonable time (a few hours to a day):

- Use a method that divides the problem into 100s of sub problems, run the discovery jobs concurrently on multiple machines, and then correlate their outputs.
- Use asynchronous methods when polling the network.

Dividing the Problem into 100s of Sub Problems

This method divides the network we have to discover into 100s of smaller parts, and has a simple discovery mechanism that performs the required discovery synchronously on each of the sub networks.

Pros

- The discovery code is relatively simple.

Cons

- In order to run the discovery jobs concurrently, there is a need for a number of dedicated machines, that have full access to the network, and are used during the discovery process.
 - If a rerun of the discovery process is needed, these machines must be available again.
- There is a need for code, tools, and work to:
 - Divide the problem to sub problems
 - Coordinate the concurrent jobs on the multiple platforms
 - Coordinate the output of the discovery jobs
 - Verify that all jobs have ended well, and resubmit the jobs that failed.

Using Asynchronous Discovery

This method polls the devices asynchronously, by sending a group of poll requests concurrently to multiple devices and then correlating the replies as they arrive. When using this method, we do not wait for each device to respond, but rather for multiple devices to respond simultaneously.

For a network of a given size, we need to perform all the discovery operations in an asynchronous way:

- Checking the ping responding nodes
- Checking SNMP responding nodes and finding their responding community
- Checking the Telnet port responding devices
- Extracting of the SNMP fields from the SNMP responding devices

Pros

- This method can be done from a single machine, or a small number of machines if no full network access exists from one machine.
- The discovery cycle is expected to be relatively short (a few hours or even less).
- Running the test requires little coordination, and is expected to be simple.

Cons

- The discovery code is more complicated, and hardly any discovery product supports this asynchronous discovery method.



Jilroy's Discovery Platform Genie

Jilroy Software's Discovery Platform Genie is the only discovery platform that has support for asynchronous discovery. Based on its already tested, powerful large scale monitoring algorithms, its discovery platform can now discover very large amounts of objects in very short cycles.

Currently, the product supports the following protocols:

- Asynchronous ping
- Asynchronous SNMP community discovery
- Asynchronous SNMP
- Asynchronous Open port scanning

and more.