Large scale discovery

White paper

Introduction

Discovery processes are usually considered slow and not time critical. However when addressing large scale organizations, even discovery processes may take too long. This paper will explain the problem, and describe a case study where it is relevant.

Case study

The Need

- A very large telco has a huge IP network.
- The address ranges of it's IP network, go over a (1,000,000) million IP addresses, distributed over a list of class B subnet addresses.
- Out of the million++ addresses, there are about 150,000 nodes that actually exist on the network.
- Out of the ~150,000 existing nodes, there are About 20% that do not exist in the users database of the telco.

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

- IP address of 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 then once.

A few numbers:

- If we assume that we need ~10 seconds to decide if a node responds to ping (3 retries * 3 seconds for timeout), then it would take ~ 10,000,000 seconds (which are ~115 days) to check which 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
- The same also goes for the actual retrieving of the SNMP data from the SNMP responding devices

We assume that such a long discovery period is not reasonable, specially if we would like 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 time of the first check).

There are 2 methods to approach the problem, so that it will be completed, in a reasonable time (a few hours to a day):

- Have a method to divide the problem to 100s of smaller problems and run the discovery jobs concurrently on multiple machines, and then correlate their outputs
- Use asynchronous methods when polling the network.

Dividing the problem to 100s of sub problems

In this method we divide the network we have to discover into 100s of smaller parts, and have 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, that will be used during the discovery process.
    - If a rerun of the discovery process will be needed then these machines should be available again
- There is a need for code, tools & work to:
    - divide the problem to sub problems
    - coordinate the run of the concurrent jobs on the multiple platforms
    - coordinate the output of the discovery jobs
    - verifying that all jobs have ended well, and resubmitting the jobs that failed.

Using asynchronous discovery

In this method we poll the devices asynchronously, by sending a group of poll requests concurrently to multiple devices and then correlating the replies as they arrive. In this method we do not wait for each device to respond but wait for multiple devices at the same time.

For a network of the 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).
- The run of 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 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 times.

Currently the product support the following protocols:

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

and more are added.