OPERATIONAL METHODS & PROCEDURES

Troubleshooting Methodology

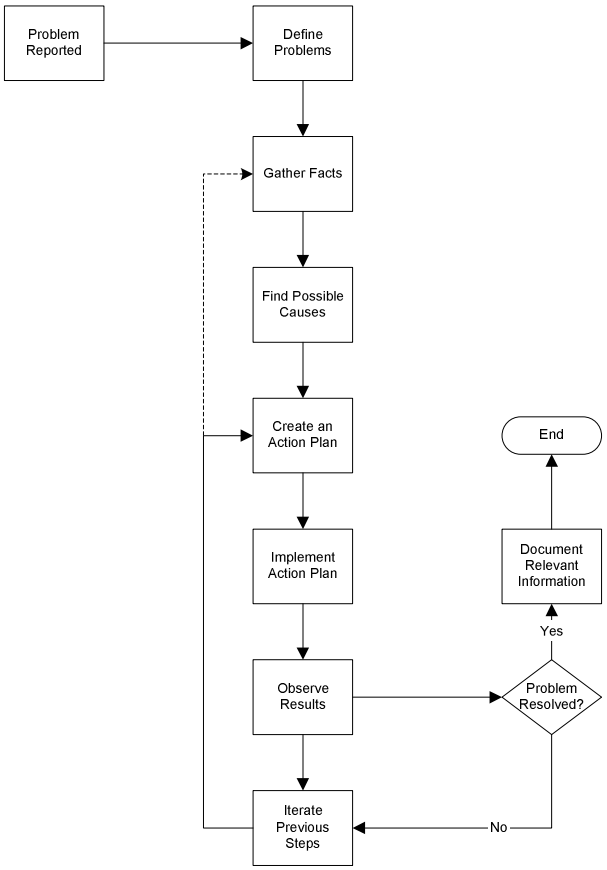
Revision 2.0

A troubleshooting methodology is a list of troubleshooting steps or processes that can be followed to provide an efficient manner of resolving network problems. The Peak XV troubleshooting methodology is a derivative of the eight step model developed by Cisco Systems. It is a proven and effective guideline to analyze problems and achieve faster resolution times.

The flow chart below outlines the eight steps. The process begins when a network failure is reported via management systems or by the customer. The steps followed are:

1. Define the problem.
2. Gather detailed information.
3. Consider possible scenarios.
4. Devise a plan to solve the problem.
5. Implement the plan.
6. Observe the results of the implementation.
7. Repeat the process if the plan doesn't fix the problem.
8. Document changes after the problem is solved.

Peak XV Troubleshooting Methodology



Step 1: Define the Problem

Problem definition is the first step in the troubleshooting model where information is analyzed to define the most likely cause of a problem. The engineer will need to gather facts before formulating a problem statement. By gathering facts to help define the problem, the diagnosis of the problem or problems will be more accurate and help the engineer solve the problem more quickly. Problem definition and fact gathering should be used in tandem for a quick and accurate resolution.

Once enough information is collected, a problem statement is created to define the problem in a specific, concise, and accurate manner. With a good problem statement, it is easier to focus on the problem and not try to troubleshoot problems that do not fall within the problem definition.

Step 2: Gather Facts

At this point, the problem is somewhat vague and requires more definition. This is where the fact-gathering step of the troubleshooting methodology is employed. Fact gathering is the process of using diagnostic tools to collect information specific to the network and network devices that are involved in a problem. Additional information should include data that excludes other possibilities and helps pinpoint the actual problem. An example is to verify that you can ping, Traceroute, or Telnet to the problem device, thus reducing the number of possible causes. It is important to gain as much information as possible to actually define the problem while in the problem-definition phase of the troubleshooting methodology. Without a proper and specific definition of what the problem is, it will be much harder to isolate and resolve. Information that is useful for defining a problem is listed in the Table below.

Information	**Example**
Symptoms	Can't Telnet, FTP, or get to the WWW
Reproducibility	Is this a one-time occurrence, or does it always happen?
Timeline	When did it start? How long did it last? How often does it occur?
Scope	What are you able to Telnet or FTP to? Which WWW sites can you reach, if any? Who else does this affect?
Baseline Info	Were any recent changes made to the network configurations?

Identify Symptoms

The engineer must first define what is working and what is not. This is done by identifying the symptom(s) and defining the scope.

Reproduce the Problem

Before spending time and effort trying to solve this problem, verify that it is still a problem. Troubleshooting is a waste of time and resources if the problem can't be reproduced.

Understand the Timeline

In addition to verifying whether the problem is replicable, it is important to investigate the frequency of the problem. For instance, maybe it happens only once or twice a day. By establishing a timeframe, the engineer can more readily identify any possible causes.

Determine the Scope of a Problem

Determine the scope of the problem by finding out how far the problem reaches. By determining what areas of the network are affected by the problem, the engineer will gain a better understanding of the underlying issue. Further define the scope by determining the boundary of dysfunction. The boundary of dysfunction is the limit or scope of the network problem. A distinction can be made between where nodes are functioning properly and where they are not.

Outside-in Troubleshooting

There are three methods of establishing the boundary of dysfunction. The first method consists of starting from the opposite end of the connection, known as outside-in troubleshooting. Start at the far end of the network and test connectivity back towards the problem device or segment.

Inside-out Troubleshooting

The second method is to start near the problem device or segment and work your way toward the outside of the network, otherwise known as the inside-out troubleshooting method. This method is only viable when the affected device or segment can be reached using in-band or out-of-band techniques.

Divide-by-Half Troubleshooting

The third and final method is the divide-by-half troubleshooting method. Divide-by-half indicates that a point between two ends of a network problem is used as a troubleshooting reference point. Either half may be investigated first. With this information, the engineer can now start to contemplate possible causes of this failure and move on to the Consider Possibilities section.

Using Baseline Information

Baseline information can be a great asset to troubleshooting network problems. Baseline information includes historical data about the network and routine utilization information. This information can be used to determine whether there were recent changes made to the network that may contribute to the problem at hand.

Step 3: Consider Possibilities

This step within the troubleshooting model is used to contemplate the possible causes of the failure. It is quite easy to create a very long list of possible causes. That is why it is so important to gather as much relevant information as you can and to create an accurate problem statement. By defining the problem and assigning the corresponding boundaries, the resulting list of possible causes diminishes because the entries in the list will be focused on the actual problem and not on "possible" problems.

These are only possible causes, the engineer will still have to create an action plan, implement it, and observe to see whether the changes made were effective. When the list of possible problems is long, it may require more iteration to actually solve the problem. The engineer must now check each of these possibilities and fix them if they are the cause of the problem.

Steps 4 and 5: Create and Implement the Action Plan

An action plan is the documentation of steps that will be taken to remedy the cause of a network problem. The investigation should have produced many possibilities for the source of the problem. Now, it is a matter of investigating each possibility. Once the engineer finds the problem, he or she must decide what is needed to fix it.

When an action plan is created and implemented, it is important that the fix for one problem does not create another. Before implementing an action plan, the engineering thinks it through or discusses it with other engineers, and make sure that the solution will fix the problem without doing anything to create adverse side effects.

A good practice when creating and implementing action plans is to change only one thing at a time. If multiple changes must be made, it is best to make the changes in small sets. This way it is easier to keep track of what was done, what worked, and what did not. The observation step becomes much more effective if only a few changes are made at one time; ideally, make only one change at a time. To summarize, the engineer follows these practices and guidelines to create a good action plan:

- Make one change or a set of related changes at a time, and then observe the results.
- Make non-impacting changes. This means trying not to cause other problems while implementing the changes. The more transparent the change, the better.
- Do not create security holes when changing access lists, TACACS+, RADIUS, or other security-oriented configurations.
- Most important, be able to revert to the original configuration if unforeseen problems occur as a result of the change. Always have a backup or copy of the configuration.

Step 6: Observe Results

Observing results consists of using the exact same methods and commands that were used to obtain information to define the problem—to see whether the changes implemented were effective. By making a change and then testing it to see whether the change was effective, the engineer moves toward the correct solution.

It may take one or more changes to fix the problem, but the engineer should observe each change separately to monitor progress and to make sure that the change doesn't create any adverse effects. After the first change is made, one should be able to gather enough information to determine whether or not the change was effective, even though it doesn't entirely solve the problem.

Once all of the changes from the action plan are implemented and the results are observed, the engineer can verify whether the action plan solved the problem. If the problem is solved, the engineer moves on and document the changes made to the network.

If the changes did not work, go back and either gather more information or create a new action plan. While working through the action plan process, the engineer might get more ideas of possible causes. The engineer should write them down; if the current action plan doesn't work, he or she has notes about some other possibilities.

If the engineer feels that all of the possible causes have been exhausted, he or she should probably go back and gather more information that can give insights into more possible causes. These steps are covered in the iteration process.

Step 7: Iterate as Needed

Iterations, or repetitions of certain steps within the troubleshooting model, are ways of narrowing down a larger problem. By implementing action plans and monitoring the results, the engineer can move toward solving the overall problem.

Iterations of the troubleshooting process allow you to focus, with more and more detail, on the possible causes of the failure. The result of focusing on the problem is the ability to identify specific possible causes for the failure. This is also the time to undo any changes that had adverse effects or that did not fix the problem. The engineer makes sure to document what was done, so it will be easier to undo the changes made to any configurations.

When is the Problem Resolved?

The problem is resolved after the engineer implements a change, observes that the symptoms of the problem have disappeared, and can successfully execute the tests that were used to aid in gathering information about the problem.

Step 8: Document the Changes

Documentation is an integral part of troubleshooting. When the engineer keeps track of the changes that were made; which routers, switches, or hosts were changed; and when the changes occurred, one has valuable information for future reference. There is always the possibility that something that was changed affected something else, and the engineer did not notice it. If this happens, one has the documentation to refer to, so the engineer can undo the changes. If a similar problem occurs, one can refer to these documents to resolve the current problem, based on what was done the last time.

Historical information is very useful in the case of a network failure. It provides a reference for the network engineer to use to see what changes were most recently made to the network. Peak XV tracks this historical data for future reference and for the enhancement of our Knowledgebase troubleshooting engine.

The Problem-Solving Checklist

The easiest way to solve network problems is to be able to compare current configurations against previous configurations. A historical baseline is simply a collection of network settings and configurations kept over time. This baseline makes it easy to locate changes or differences between a current configuration and a previous one.

Baselines provide the following types of information:

- Network topology
- Router configurations
- Network and routing protocols
- Traffic levels and network events
- Applications
- Changes made to network configurations
- Historical information that documents previous troubleshooting sessions

In addition to having all of this data available, it is helpful to have a checklist that one can refer to when troubleshooting. A list may be created from baseline information. Each network's checklist will look different.