The Process of Analyzing Failure Reports
to Determine the Number of Variables Interacting in a Software Failure

Author: Menal Modha, SURF 2009, Information Technology Laboratory, Division 893

For combinatorial testing, it is imperative to analyze data from different software to determine the number of interactions that cause a problem or failure. After careful analysis of data from different software applications like medical devices, web browsers and servers, it was concluded that most faults occur due to one way or two way combination. A seven way interaction failure has not been observed in applications that have been studied.

But analyzing failure reports can be a complicated process and any help is always appreciated. To figure out what caused an error you have to sometimes traverse through the labyrinth of description and diagnosis of the error and sometimes you don't have information to traverse through. But careful attention can yield the expected results.

Let us look at some examples to further our understanding of some general principles of analyzing failure reports.   There are different kinds of error reports and the goal is to analyze them and effectively categorize them into *t*-way causes.

Our research group received data from a research laboratory (undisclosed because of proprietary reason). This lab does spacecraft testing and any errors, faults or failures are logged into a database. Each failure was described in detail with a description and diagnosis blog, and the error report also contained a list of tasks to fix the problem. The error report contained terms with which we were completely unfamiliar but that did not hinder us. Sometimes you get lucky and the report says, "This error was caused by _______" and then you can categorize that error into one way combination. It also helps to write the interaction description in the form of an expression. For example, the error occurred when variable *X* = *null*. If you think in terms of an expression like this, it is easier to find the cause of the error.

One thing you need to be careful about is hardware error. In this example of spacecraft testing, sometimes the software works fine but some part of the hardware is malfunctioning which causes a failure in the application as the whole. Some hardware errors are not predictable and cannot be tested by the testing software.

The causes for the errors are not apparent in many cases and you have to read the given information more carefully to categorize the error and sometimes you need to seek the help of an expert to understand the given data.

In some cases, the cause for an error is never determined or there is not enough information in the report to figure out the cause. Undetermined cases vary with application, but for the spacecraft and National Vulnerability Database (NVD) data, about 13-15% of the total error reports fell into this category.

The NVD is the U.S. government repository of standards based vulnerability management data. It contains condensed error reports. Here is one example: “modules/viewcategory.php in Minh Nguyen Duong Obie Website Mini Web Shop 2.1.c allows remote attackers to obtain sensitive information via a request with an arbitrary catname parameter but no itemsdb parameter, which reveals the path in an error message.” *(NVD, 2006)* So this is a two way interaction and the expression can be written as: *catname = arbitrary value* AND *itemsdb = null*.

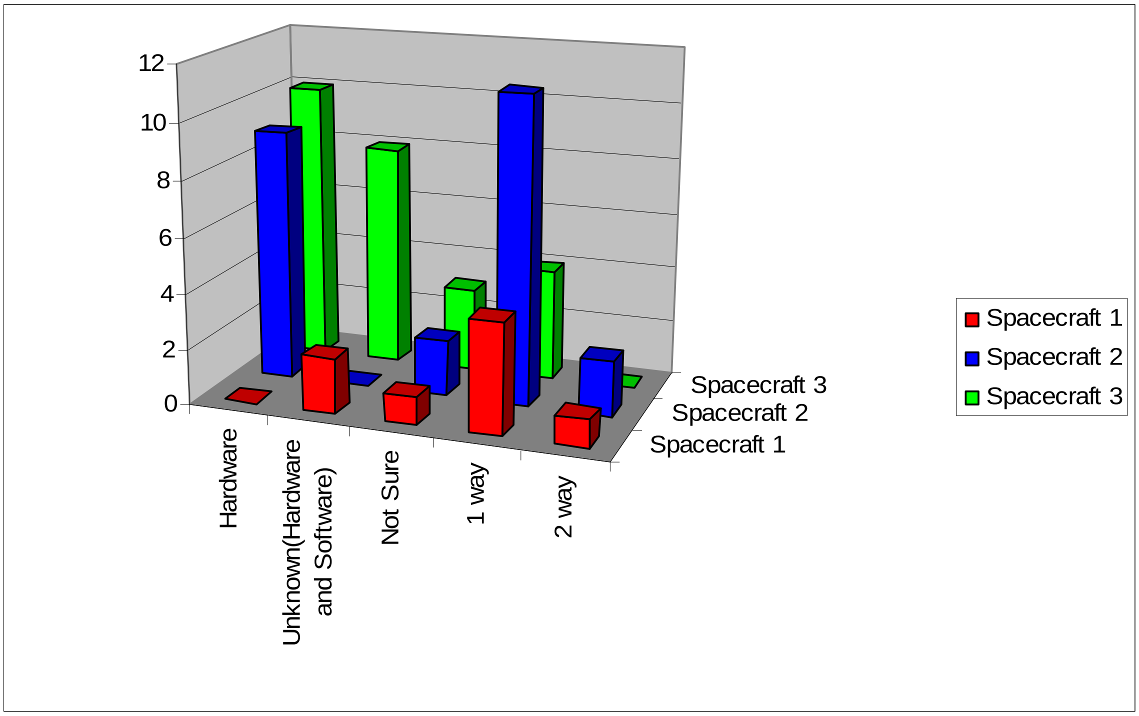
Now let us look at one way and three way interaction errors from the NVD to get a clearer picture. “The Huawei Versatile Routing Platform 1.43 2500E-003 firmware on the Quidway R1600 Router, and possibly other models, allows remote attackers to cause a denial of service (device crash) via a long show arp command.” *(NVD 2007)* This is a one variable interaction and the expression is *show arp length* > *N*.

Let’s look at a three way error. “SQL injection vulnerability in blocks/block-Old_Articles.php in Francisco Burzi PHP-Nuke 7.9 and earlier, when register_globals is enabled and magic_quotes_gpc is disabled, allows remote attackers to execute arbitrary SQL commands via the cat parameter.” (NVD 2007) The interaction expression for this one is *register_globals = true* AND *magic_quotes_gpc = false* AND *cat parameter = arbitrary malicious command*.

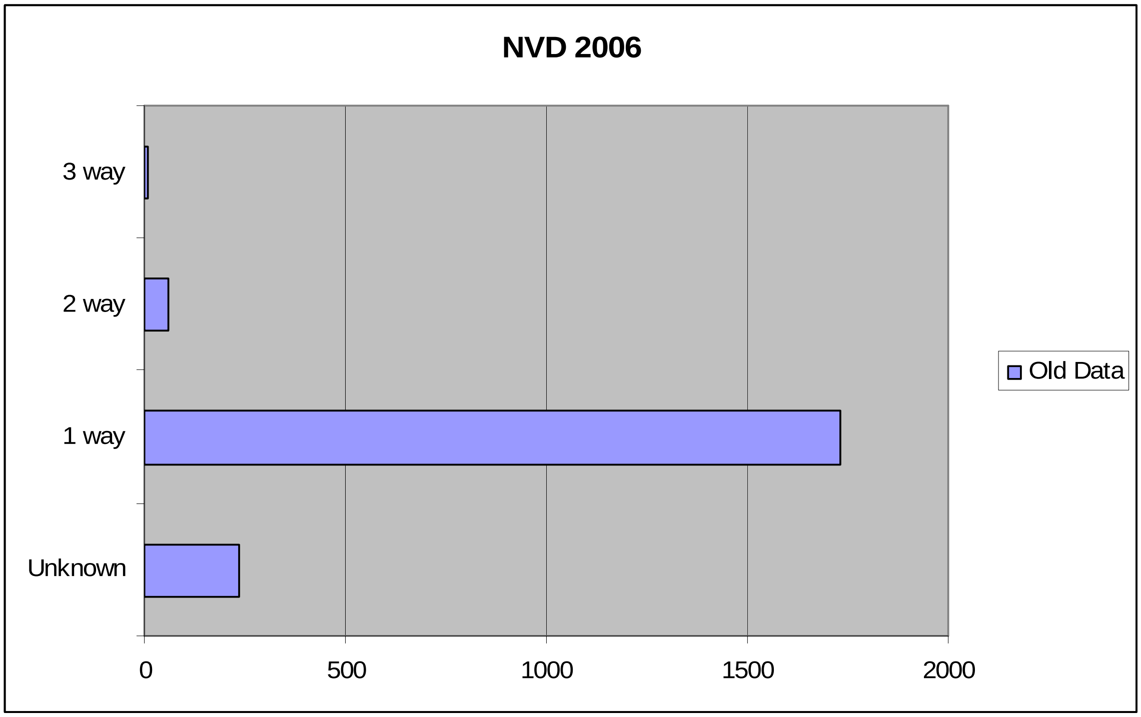
Here is an example of an undetermined error, “Unspecified vulnerability in Oracle Pharmaceutical Applications 4.5.1 has unknown impact and remote authenticated attack vectors, aka Vuln# PHAR01.” *(NVD 2006)*

NVD data from year 2006 consists of total 2037 reports and the data from 2007 has total 1008 reports.

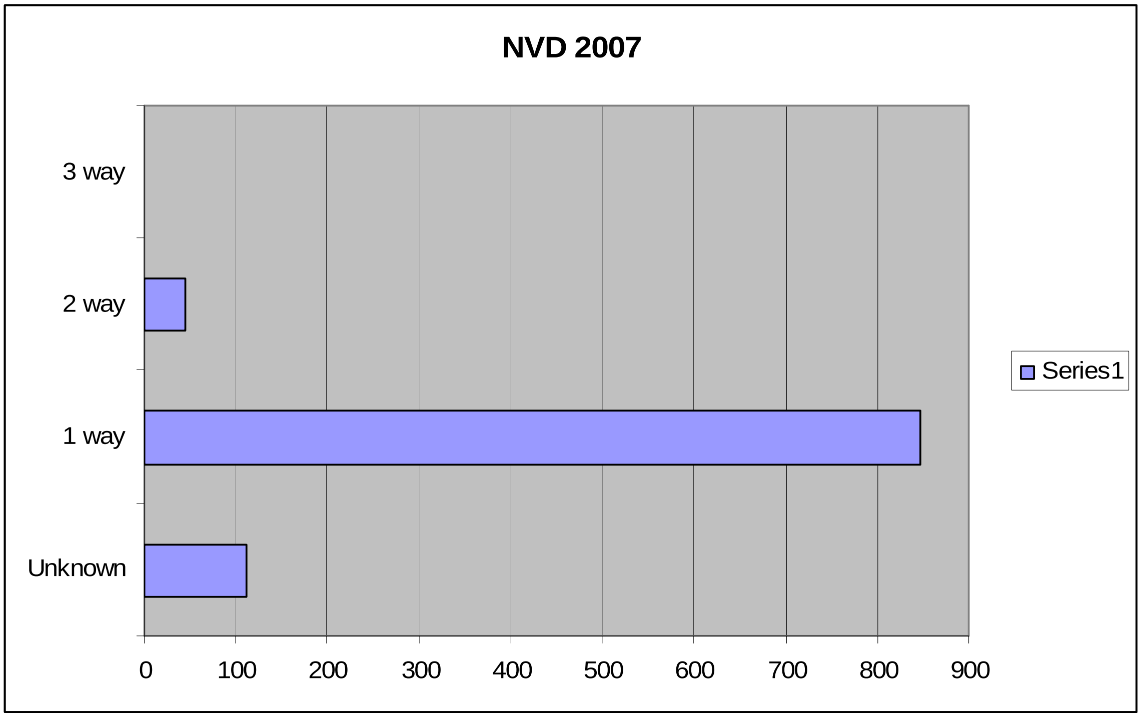
The graphs below show the distribution of the errors:



Spacecraft Data Categorization



National Vulnerability Database, Reports from year 2006.



National Vulnerability Database, Reports from year 2007.