Top 25 hit or miss?

Michael Barwise

Mitre have just published their 2011 "Top 25 Most Dangerous Software Errors" list - a ranking of what they consider to be the most egregious generic online programming goofs perpetrated last year. It's a worthy effort and contains a huge amount of valuable guidance, but I have serious reservations about it nevertheless.

It's a strange mixture of issues in a strange order, and for "generic" errors, they don't seem quite generic enough. For example, number 3 - *"Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')"* has a lot in common both conceptually and technically with *"Incorrect Calculation of Buffer Size,"* but that's at number 20 - obviously much a less critical issue then. The top two slots are occupied by *"Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')"* and *"Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')"* but their primary common root cause (also that of number 4 *"Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')"*) - failure to validate user input properly - is not listed anywhere in the ranked list. It's only referred to in the guidance notes - a standard (albeit admittedly informative) paragraph starting *"assume all input is malicious..."* that occurs in eight of the 25 mitigation sections. This failure to address first principles makes the ranked list longer than it need be and results in massive redundancy in the detailed mitigation advice. But most importantly it obfuscates the real problems to be solved by emphasising superficial externalities at the expense of root causes.

So why is the "Top 25" like this? It's obviously been produced by serious technical experts. But taking a look at the way the list is compiled, I find the method's not terribly scientific. For starters, it's a trawl of subjective opinion from a self-selected public. The responses are voted on by a panel of *"software developers, scanning tool vendors, security consultants, government representatives, and university professors"* and then ranked using the Common Weakness Scoring System (CWSS).

The CWSS is similar to the CVSS from First.org in that it ranks an issue on the aggregate of a bunch of weighted generic characteristics - ease of exploitation, impact and so on. Both these tools do facilitate arriving at a ranking, but they have two attributes in common that make me cautious about relying on their results too heavily. First, the documentation for neither system explains the rationale for either the weighting values or the formulas they use to combine those weightings into the final ranking - they're taken as read, and they're not at all transparent. Second, both scoring systems operate at the highest possible level of detail - "this specific weakness/vulnerability" - ignoring commonalities with others. In the case of the CVSS the latter can be justified by its primary purpose - to rank specific publicly known software vulnerabilities to guide patching decisions. But to be really useful, the CWSS should operate at a level much nearer to first principles as it intrinsically deals with much more generic issues.

One might think the excessively high-level breakdown of "software errors" in the "Top 25" was driven by these characteristics of the CWSS, but I think both are symptomatic of a more general problem - the superficial analysis and poor recognition of the significance of root causes which are endemic in the infosec community. Mitre is not alone - indeed the deficiencies of the "Top 25" are a relatively minor manifestation of this. An excellent paper by Microsoft Research called *"Sex Lies and Cybercrime Surveys"* argues convincingly that the results of "cybercrime surveys" are almost always completely invalid - primarily due to inadequate recognition of the statistical properties of the sampled population. Analysis of reports by small numbers of self-selected respondents is incompatible with sparse populations of stochastically distributed rare events.

We need to encourage software *engineering* in place of mere "development". We should therefore be grateful to the "Top 25" for identifying poor practice so we can eradicate it. We need to raise infosec management from its current status of a medieval black art to that of a modern science, so we should deprecate the way the "Top 25" is presented.

To achieve either of these objectives will require practitioners who can analyse real-world problems to first principles and synthesise from those principles to entire robust systems. While we rely on cursory observation, snap decisions and guesswork to address external appearances, we'll never be able to trust our own judgement - let alone that of others. Sadly, the way Mitre has categorised software errors in the "Top 25" tends to perpetuate the problem by encouraging such superficial thinking.

Originally appeared on the Infosecurity Network, July 2011