FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS OF INFORMATION SECURITY

Manjula Verma and Dr. Pardeep Goel

[1]Research Scholar, CMJ University, Shillong, Meghalaya

[2]Associate Professor M.M. College, Fatehabad

Abstract:

Fundamental principle in security design is to plan for failure. Development projects are mainly focused on base flows of the system since these implement business valuable features. However from a security standpoint, exceptional and alternate flows highlight paths that often become attack vectors once the system is deployed. These flows are worth examination by Information Security to ensure that the system is not likely to enter an insecure state and to identify areas to deploy security mechanisms such as audit logs and IDS tools to catch security exceptions when they occur.

Key words: *business, vectors, examination, security exceptions.*

INTRODUCTION

Software requirements are typically divided into two categories: functional and non-functional. Functional requirements deal with what the system is supposed to do, for example deposit money into a bank account. Non-functional requirements deal with everything else including performance, security, availability, usability, and scalability. Functional requirements are generally written out indicating what business features are required of the software system, decomposing high level business feature sets down into a finite set of requirements. Non-functional requirements tend to be more nebulous. Often there is a lack of precise metrics. In the case of security requirements, the process can seem counter-intuitive. Most requirements stipulate that the system must do something, while security requirements are frequently focused on ensuring something does not happen. This difference is at the root of many security gaps in software.

Review of literature

Use Case pioneer Ivar Jacobson points out that as a whole, all Use Cases describe all possible ways of using a system. Use Cases answer the question: what is the system supposed to do for each user? [1] Use Cases differ from requirements in two main ways. First Use Cases are used to generate a shared understanding of the problem to be solved, the key relationships and actors in a system. Bittner and Spence [2] refer to this as building up a shared understanding as opposed to decomposing features. The result of this is the second difference, which is that a Use Case model places requirements in a certain context. Context is critical in security in that the context can show how the Use Cases are related to the assets which the security mechanisms must protect, and the overall flows, dependencies and assumptions that the system makes.

Let's examine ten ways that Use Case models can be developed in a more security-focused way.

Material and method

Use cases provide a synthetic model that correlates requirements from different domains' concerns into a coherent model and flow. Use Case models provide a format to conduct architectural tradeoff analysis of security mechanisms at different points in the system and establish a document for the tradeoff decisions.
In Information Security, it pays to find allies. Stakeholders who may be concerned about security implications in the system that is being built include not just the core development staff, but also the legal staff, business owners, domain experts, operational staff, customers, shareholders, and users. The Use Case model captures and documents stakeholders that have some stake in the outcome of the development project. The Information Security team should document these stakeholders' unique concerns and viewpoints with regard to security.

Conclusion

Pre and post conditions describe the set conditions that must be satisfied for the Use Case to execute (Pre-conditions) and the set of states that the system can be in after it has completed (Post-conditions). Pre-conditions allow the Information Security team to articulate the security conditions, such as authentication and authorization processes that must be completed before accessing the Use Case functionality.
Information Security policies defines acceptable states for the system to be in to be in accord with the policy. There are typically many different possible states a system can be in at the end of a Use Case. Use Cases describe basic or expected flows, and also exceptional and alternate flows. Each flow may result in one or more states. The Post-conditions document the set of states possible at the end of the Use Case. The Post-conditions illustrate what must be done to tear down a system at the completion of Use Case which could include disabling a user's session, locking accounts, deleting temp files or cache, and closing accesses to resources such as databases. By stipulating, security concerns in the Use Cases' Pre and Post Conditions, early in the development lifecycle the developers are empowered to write more secure code.
A fundamental principle in security design is to plan for failure. Development projects are mainly focused on base flows of the system since these implement business valuable features. However from a security standpoint, exceptional and alternate flows highlight paths that often become attack vectors once the system is deployed. These flows are worth examination by Information Security to ensure that the system is not likely to enter an insecure state and to identify areas to deploy security mechanisms such as audit logs and IDS tools to catch security exceptions when they occur.

References

Abe H., Yokoi H., Ohsaki M. and Yamaguchi, T. (2007). Developing an Integrated Time-Series Data Mining Environment for Medical Data Mining. Seventh IEEE International Conference on Data Mining, 28-31 Oct. 2007, 127-132.

Adriaans P. and Zantinge D. (1999). Introduction to Data Mining and Knowledge Discovery. 3rd Edition Potomac, MD: Two Crows Corporation.

Adriaans P. and Zantinge D. (2003). Data Mining. Pearson Education, Seventh Indian Reprint, 2003.

Agrawal R. and Srikant R. (1994). Fast Algorithms for Mining Association Rule. **Proceedings of the 20th** International Conference on Very Large Databases (VLDB), 487 – 499.

Agrawal R., Faloutsos C. and Swami A. (1993). Efficient similarity search in sequence databases. Proceedings of the Fourth International Conference on Foundations of Data Organisation and Algorithms, Chicago, Vol. 730, 69-84.

Agrawal R., Imielinski T. and Swami A. (1993). Mining association rules between sets of items in large databases. **Proceedings of the 1993 ACM SIGMOD International Conference on Management of** Data, Washington DC, 207-216.

Ahmed S.R. (2007). Applications of Data Mining in Retail Business. International Conference on Information Technology: Coding and Computing, Las Vegas, Nevada, Vol. 2.

Anand S.S., Bell D.A. and Hughes J.G. (1995). The Role of Domain Knowledge in Data Mining. **Proceedings of the Fourth International Conference on Information and knowledge management, 37-43.**

Ankerest M. (2001). Human Involvement and Interactivity of the Next generation's Data Mining Tools. ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. Santa Barbara, CA.

Ankerest M., Ester M. and Kriegel H.P. (2000). Towards an Effective Cooperation of the User and the Computer for Classification. Proceedings of 6th International conference on Knowledge Discovery and Data Mining, Boston, MA.

Aruna P., Puviarasan N. and Palaniappan B. (2005). An Investigation of Neuro-Fuzzy Systems in Psychosomatic Disorders. Expert Systems with Applications. Vol. 28, 673-679.

Bates J.H.T. and Young M.P. (2003). Applying Fuzzy Logic to Medical Decision Making in the Intensive Care Unit. American Journal of Respiratory and Critical Care Medicine, Vol. 167, 948-952.

Bayrak C., Kolukisaoglu H. and Chia-Chu Chiang .(2006). Di-Learn: Distributed Knowledge Discovery with Human Interaction. IEEE International conference on Systems, Man and Cybernetics, Taipei, Taiwan, Vol. 4, 3354 – 3359.

Berks G., Keyserlingk D.G.V., Jantzen J., Dotoli M. and Axer H. (2000). Fuzzy Clustering - A Versatile Mean to Explore Medical Databases. ESIT, Aachen, Germany, 453-457.

Berson A., Smith S. and Thearling K. (1999). Building Data Mining Applications for CRM. First Edition, McGraw-Hill Professional.

Bethel C.L., Hall L.O. and Goldgof D. (2006). Mining for Implications in Medical Data. Proceedings of the 18th International Conference on Pattern Recognition,Vol.1, 1212-1215.

Bicciato S., Luchini A. and Di-Bello C. (2004). Marker Identification and Classification of Cancer Types using Gene Expression Data and SIMCA. Germany: Methods of Information in Medicine, Vol. 43(1), 4-8.

Brause R.W. (2001). Medical Analysis and Diagnosis by Neural Networks. Computer Science Department, Franfurt a.M., Germany.

Chattoraj N.and Roy J. S. (2007). Application of Genetic Algorithm to the Optimisation of Gain of Magnetised Ferrite Microstrip Antenna. Engineering Letters, Vol. 14(2).

Cheung Y.M. (2003). k-Means: A New Generalised k-Means Clustering Algorithm. N-H Elsevier Pattern Recognition Letters 24, Vol 24(15), 2883–2893.

Chiang I.J., Shieh M.J., Hsu J.Y.J. and Wong J.M. (2005). Building a Medical Decision Support System for Colon Polyp Screening by Using Fuzzy Classification Trees. Applied Intelligence, Vol. 22 n.1, 61-75.

Chung H. M. and Paul G. (1999). Special Section: Data Mining. Journal of Management Information Systems, Vol. 16(1), 11 – 16**.**