Integrity Testing in WWW Environment

Martin Stanek and Daniel Olejár
Department of Computer Science
Comenius University
Mlynska dolina,
842 15 Bratislava, Slovakia
Tel. (421 7) 654 26 635
e-mail: stanek@dcs.fmph.uniba.sk, olejar@dcs.fmph.uniba.sk

Key words: integrity testing, WWW, heuristics

Abstract:
The successful growth of Internet, lead to application of World Wide Web (WWW) technologies in current IT systems. One of the traditional high-level objectives of IT security is integrity. We focus our attention on integrity in the WWW environment. We outline content and context of integrity testing as well as specific problems that need to be solved. Most of our attention is devoted, in a separate section, to heuristic approach to integrity testing.

1. INTRODUCTION

The rapid growth of Internet changes the traditional view on information systems (IS). Individual information systems connected to the Net can be viewed as one large virtual IS. The introduction and development of electronic commerce on the Net attract the attention of businessmen, customers and others, mostly laymen in IT. The requirements of the users of Internet include simple common users environment, the possibility to communicate both in the frame of organisation and outside of it, standardisation, interoperability, etc. WWW technologies meet most of these requirements and that is the reason, why many people see the future of IT systems/products in them. Beside the above mentioned functional features of IT, there are also other, less visible to laymen, but nevertheless very important security aspects of IT.

Traditional high-level objectives of IT security are confidentiality, integrity and availability (see [2], [10]). The evolution of web technologies takes into consideration all these needs. Both existing and developing standards contain features increasing the web technologies security awareness (concentrating especially on confidentiality and authenticity) and thus enable the use of web-based

---

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35501-6_14

M. E. van Biene-Hershey et al. (eds.), *Integrity and Internal Control in Information Systems*

systems in variable sensitive/insecure environments. Some of these standards (from the lowest to the highest application layers) are: SET – Secure Electronic Transactions [6], SSL – Secure Socket Layer [7], HTTP Digest Access Authentication [1] and other from the lowest to the highest application layers.

Though there are various ways how to exploit the WWW, in common praxis it serves mostly for the following purposes:

- presentation of an organisation to outside world (via Internet); establishing the "Internet identity" of an organisation,
- providing services for customers (technical support, information desk, etc.),
- internal communication inside of an organisation (Intranet),
- electronic commerce.

While browsing the Internet, almost every user finds references to nonexistent pages, scripts incompatible with his/her browsers, links leading to documents completely different from those they have promised, etc. Similar problems appear in large Intranets, too. The lack of integrity is a general problem concerning not only private, academic or small enterprise's WWW pages, but also the carefully created and maintained web documents of large and important institutions. The WWW integrity will gradually become a major problem. The traditional methods of integrity testing and managing seems to be insufficient in large domains and so to increase the integrity of WWW, new methods tailored to WWW environment are to be developed (and used).

We concentrate on integrity problem in the WWW environment. At first we outline content and context of integrity testing and then we discuss heuristic approach to integrity testing. Since it is impossible to test the integrity of the whole web, we assume that the integrity is tested and maintained in a restricted part of WWW (e.g. single web site).

2. INTEGRITY IN WWW ENVIRONMENT

Integrity testing in WWW environment is neither an easy nor a cheap task. Moreover, despite the invested effort, the satisfiable results cannot be guaranteed. Therefore the opinions of users on WWW integrity testing ranges from the extreme negative to extreme positive attitudes. The opponents consider the integrity testing useless, since

- administrator/web designer checks the integrity when developers particular pages,
- potential problems are fixed, when someone finds them,
- the cost/benefit ratio of integrity testing is too high, since their pages are visited only by a small number of customers, etc.

The supporters of integrity testing consider it vitally important, since

- otherwise the content cannot be trusted, and it is better to make decisions based on uncomplete but correct information, than to risk wrong decision based on complete but slightly incorrect one,
- an integrity problem will accumulate and propagate into other parts of system with unpredictable consequences, etc.

Both extreme opinions are partially true. The necessity of integrity testing depends heavily on the environment where the WWW is used and purposes it serves, too. Therefore it ought to be a subject of a pragmatic analysis, taking into account possible negative consequences (the damaged image of particular organisation; the loss of customer's confidence in organisation offering poor services; the propagation of problems in databases, mail system, etc.) and costs on the other hand. The integrity testing in WWW environment has specific features resulting from:

- the rapid increase of Internet both in size and complexity;
- the frequent and dynamic changes of its content and structure;
- the large number of departments and employees managing/changing parts of the web,
- the necessity of taking into account the developing/emerging standards;
- the limited possibilities of a single person (organisation) to manage the all information needed (e.g. the outside world connections); etc.

These and other factor practically eliminate the possibility of developing a "miraculous" method or a tool able to retain the integrity of web environment at required level. The possible solution is therefore to adopt system approach and continually/dynamically improve it with respect to achievements in this area.

Certainly, contemporary web authoring solutions (e.g. Adobe PageMill, Microsoft FrontPage) offer tools for helping to manage the integrity of site. Automatic uploads to web, updates of links, site structure overview, and other features become a standard in this area.

3. CONTENT AND CONTEXT OF INTEGRITY

This section is dedicated to discussion of content and context of integrity in WWW environment. The content deals with subject of testing, i.e. it answers the question "what to test". The context outlines broader frame surrounding testing, i.e. it answers questions "what to do before it and after it", "what to do with results", etc.

3.1. Content

The word "content" may be slightly misleading in this context, since it does not denote the meaning of web documents in broad sense, but only the meaning of some special kind of information they contained. The integrity of content has two aspects: syntactic and semantic. The web integrity on semantic level means that the

information provided by WWW is consistent; i.e. the information presented by one part of organisation's web site is not in contradiction to information presented in other part of its web site. A special problem of semantic integrity is the correct direction of links/references; e.g. the link labelled as "Model X300" should not lead to "Model X290" page. Another semantic integrity problem is the integrity of temporal data, see [5]. It will be difficult to develop a system which would be able to test the semantic integrity, since the definitions of "semantic rules" (describing what is correct and what is not) are very complex or often missing. Metadata reference link <MREF> ([8]) offers one (partial) solution how to represent semantic correlation. Sneth [8] wrote:

> "The hard questions related to the degree of consistency needed among these related data managed by heterogeneous and independent information resources, and the techniques for enforcing such consistency requirements, remain to be answered."

The semantic integrity cannot be reached without maintaining the syntactic integrity (or, integrity of the structure). The list of "structural integrity" problems is very long and depends on the web technologies used, and therefore we mention only selected items:

- page is conformant with specification HTML (DHTML, XML, JavaScript, etc.),
- links referring to existing objects,
- all required bookmarks on page are present,
- every button has defined action,
- correctness of image maps (regions, coordinates, etc.),
- unused pages (in Intranet), i.e. such pages which are not referenced from anywhere, are removed, etc.

Further we will mostly discuss this type of integrity. On the first sight it seems that mechanisms for managing the integrity of structure are much easier than the ones in the case of content integrity. Difficulty is in large number of questions that have to be answered by testing. It is a difficult task to define all integrity rules with respect to the large number of standards, indeed. Moreover, it is necessary to redefine rules after new standards have emerged or were extended.

3.2. Context

Integrity testing is neither the unique nor the sufficient procedure, we ought take into consideration, when integrity problems are to be solved. The testing has to be set in a broader frame and therefore, it is more appropriate to speak about "integrity maintenance". The integrity maintenance should employ:

- policies/procedures for developing WWW objects and for defining their relationships,
- policies/procedures for integrity testing,
- polices/procedures for integrity recovery,
- utilities which help to fulfil tasks mentioned in previous points.

Procedures for development of WWW objects include the rules defining: who is allowed to develop them, who may upload them to web and where, which tools can be used in this process, who does control them (both their semantics and structure), etc. Briefly speaking, the rules of developing and maintaining WWW objects are to be defined in such a way that the security goals are ensured by using them. We assume that, similarly as other goals, keeping WWW integrity is properly reflected in these rules, too.

Policies for integrity testing covers rules, who and how is allowed to test integrity, which tools must be used, what must be done in the case an integrity problem is detected, etc. We want to stress, that even in the case of using tools (more or less) enforcing integrity for corporate web development, we do not avoid its testing. It is caused by, for example, connections with outside world, with systems that we do not manage.

4. HEURISTIC APPROACH

Approaches to integrity testing can vary. We can classify them by using various criteria: integration with web authoring tools, implementation language, extensibility, user-interface, etc. We can divide them into two categories according to how many integrity problems they address:

- Complete methods,
- Heuristic methods.

The complete (deterministic) methods of integrity testing check all problems that could emerge (when particular check is implemented), they probe every page, every reference. In principle, they scan through whole WWW domain or specified part of it – e.g. with depth first search, breath first search, and so on. The advantage of complete methods is assurance that after successful testing the integrity is approved. Drawbacks of complete methods are relatively low efficiency, especially in large domains, low flexibility and consumption of resources. An example of complete method for checking HTML syntax is weblint program (http://www.weblint.com/). An example of complete method for checking links (outdated, broken, slow, etc.) is weblint program (http://starship.python.net/crew/marduk/linbot/).

Heuristic methods offer only partial solutions; they test only fraction of integrity problems, or they test them only at a certain level. Drawback of these methods is the possibility, that some integrity violation(s) will not be detected. On the other hand, the advantages include efficiency (time requirements, network load, etc.), the possibility to concentrate on the most important integrity problems, the most frequent error sources and so on. As stated in [8], less than absolute integrity methods can better capture the true complexity of IS management.

Regardless of classification, a tool for integrity testing ought to act like automatic and intelligent browser. It should verify the integrity of particular objects and their

relationship in WWW just like regular user do while browsing. This verification should be without human intervention unless a problem is detected.

There are also other aspects (related to WWW technologies) that have an influence on the efficiency of integrity testing. For example, HTTP 1.1 standard [4] allows, as default, multiple objects to be transferred through one opened TCP/IP session (so called persistent connection).

It will be interesting to try to develop a mechanism able to automatically correct (at least selected kinds of) integrity errors, as a tool for web administrator.

4.1. Criteria and requirements for heuristic methods

Before developing or using heuristic methods of integrity testing, the requirements and/or criteria they must meet are to be discussed. The good testing heuristics are expected to satisfy the following requirements:

- Automatic mode – after initial configuration the tool tests (incessantly or in regular intervals) domain integrity without human intervention. In the case that integrity error is detected, responsible person is informed (e.g. via email). Naturally, the decision can be made according to the type of an error, its significance or with respect to its location in domain.
- Configurable accuracy – possibility of configuration of the tool, i.e. how often the heuristic is allowed to make an error – e.g. an integrity flaw is not detected. Certainly, such parameter substantially influences the speed of testing. It is appropriate to set different accuracy parameters for various integrity aspects, such as correctness of references, standards compliance, etc.
- WWW standards are supported – the tool should support all standards and techniques used in WWW domain of an organisation.
- Extensibility – easy extensibility to reflect new technologies and standards and constructs, which we plan to test. It can be done in the form of "plug-in" modules.
- Efficiency – "small and quick". The tool should not, for example, overload network or web server, take hours (or even days) to complete testing, etc.

4.2. Some heuristic rules for integrity testing

We describe several heuristic rules in previous section, which are suitable for WWW integrity testing. These rules are of more general nature and they can be used with various standards/formats (e.g. HTML 4.0 standard [3]). They do not depend on chosen format and have rather "philosophic" nature. The goal of presented rules is to make integrity testing more efficient. Our list should be taken as an initial attempt to accomplish the "efficiency" goal. This list is far from complete and it is not the only right one. Decision what to choose and how to apply heuristic procedures in details depends on a particular situation.

4.2.1. Probability (accuracy) utilisation

Natural heuristic approach is defining probabilities for testing properties. It is suitable to use different probabilities for different issues (e.g. page presence, validity of bookmarks, syntactic correctness, conformation of standards). We can take into account at least the following facts:

- Some tests are much faster than others are. For example, checking syntactic correctness of a page is usually faster than checking the validity of links pointing from the page. Generally, we can perform fast tests in greater detail (the probability can be greater).
- Similar types of integrity problems are more likely to occur more frequently. Hence, we can concentrate our attention on currently detected problems and dynamically change (increase) particular probabilities during the integrity testing. Analogously, we can decrease other probabilities in the case that corresponding problems have not been detected for a long time.

An important factor in web integrity testing is the knowledge of web (domain) structure. The domain structure is an oriented graph, where vertices represent pages or frames and arcs represent links between them. The knowledge of domain structure simplifies the testing – it allows looking at pages uniformly, even at nonadjacent ones. We can visit, for instance, an exact fraction (e.g. 70%) of pages in this case. More detailed structure with additional information allow to accommodate the testing for different parts of domain. We are faced with different problem when the knowledge of domain structure is absent. We traverse the web step by step through adjacent pages. It can take longer to reach a distant, although potentially huge and important part of web. The traversing strategy depends on supposed topology of domain. For example, when we assume the domain structure that is almost tree (with a few exceptions), it is worth to proceed in following way. The links (arcs) in smaller depth are used (we travel along these links) with greater probability and the probability decreases with deeper-lying links. To conclude the successful heuristics should employ procedure that gathers domain structure and is able to utilise this structure in testing.

We already saw that the probabilities should change during testing or between consecutive tests. These changes can be based on:

- Experience – the problems occurring frequently than others in our domain require greater attention, (see section 0);
- Importance – the most important things (pages, pictures, etc.) should be checked more frequently than unimportant ones (see section 0);
- Types of problems – similar types of integrity problems are more likely to occur more times, see above;
- Locations of problems – problems emerge usually in groups. One problem is often accompanied with another. Hence, it can be fruitful to temporary increase the probabilities in environment surrounding the fault item/object.

4.2.2. Hierarchy aspects

We can adopt two different strategies regarding hierarchical aspects of references (i.e. URI, Universal Resource Identifier). The first one is to prefer testing of lower/deeper level objects. The second one is to prefer testing of higher level objects. Example of pair of lower and higher level references are

www.mycompany.xx/products/Model210/discounts.html and
www.mycompany.xx/products/.

Both mentioned strategies can be justified. It is reasonable to assume that the integrity of important parts (high-level objects) is met when "details" (low-level objects) are correct, because these are more likely to be corrupted. On the other hand, high-level objects are more likely to be used/viewed by users and therefore their integrity might be more important. Naturally, it depends on situation, which strategy should be used.

4.2.3. Importance

We should concentrate on important things. That means those, which are widely (extensively) used in our environment. This rule can be implemented via accuracy utilisation. Example of "less important" element is ⟨NOFRAMES⟩ parts of HTML's frame environment and ⟨LINK⟩ elements. Certainly, the importance of elements depends on particular situation.

The important things can be defined also as those other elements depend on (e.g. form that serves for generation of customised pages).

There are sites, where importance changes often. Electronic newspapers, magazines, discussion forums, and others require a dynamical change of our focus. These changes are (mostly) predictable, therefore it is possible to incorporate appropriate mechanisms into heuristics.

4.2.4. Utilisation of users

A uniqueness of the WWW environment is the interaction. Users communicate with web server. A system without users is pointless and there is need to bother with it and with its integrity. The interaction of users with web server provides information, which can be used in integrity testing. Particularly, it can help to answer following questions:

- What is important – It is natural to suppose that the most frequently visited parts of domain are the most important ones, too.
- Where are problems – We can discover some integrity problems (e.g. missing pages) in the way that the user requests cannot be served by web server. Certainly, we need to distinguish between errors on server and user sides of interaction (not all error are caused by integrity problem).

Naturally, the implementation of these ideas requires a server part of testing tool – specific regarding used server.

4.2.5. Heuristic learning

The maintenance of integrity can be used as a feedback and utilized to improve the strategy of testing. The adequate setting and resetting probabilities for a heuristic operating in a large domain is a time-consuming and complicated task. If the administrator would be able to define the way how to compute new probabilities after discovering an integrity fault, he could set the initial probabilities and the futher resetting could be done automaticaly. This may be seen at the first look as a very simple solution, but the inner complexity of the original problem does not disappear. The crucial problems are, how to establish the adequate values of various objects, how to evaluate the integrity faults and how to derive the changes of probabilities from these parameters. One of possible solution is to set a vector of initial probabilities *ad hoc* and create a log file for recording integrity incidents. After some period, the records will be processed (e.g. sorted with respect to objects, kind of integrity problems, etc.) and the probabilities or weights will be modified proportionally to the results of integrity incident analysis.

5. CONCLUSION

We discussed various aspects of WWW integrity testing, its importance and scope. Paper outlines heuristic approach to integrity testing. Further work should be devoted to more detailed specification and implementation of this approach. An automatic recovery from integrity errors seems to be very challenging problem and should be addressed, too.

Acknowledgements.
We would like to thank all anonymous referees for many helpful suggestions.

6. REFERENCES

[1] Franks J., et. al.: An Extension to HTTP – Digest Access Authentication, RFC 2069, 1997.

[2] ISO/IEC 15408 Evaluation Criteria for Information Technology Security (Common Criteria v. 2.0), 1998.

[3] HTML 4.0 Specification, W3C Recommendation, 1998.

[4] Hypertext Transfer protocol – HTTP/1.1, Internet Engineering Task Force (IETF), Internet Draft, 1998.

[5] Knolmayer G.F., Buchberger T.: Maintaining temporal integrity of World Wide Web pages, Integrity and Internal Control in Information Systems, Volume 1, pp. 195-202, Chapman & Hall, 1997.

[6] SET - Secure Electronic Transaction, (http://www.visa.com/set, http://www.mastercard.com/set), 1997.

[7] SSL - Secure Sockets Layers 3.0, (http://home.netscape.com/eng/ssl3), 1996.

[8] Sheth A.: Managing with Less than Absolute Integrity, Integrity and Internal Control in Information Systems, Volume 1, pp.195-202, Chapman & Hall, 1997.

[9] Sheth A., Kashyap V.: Media-independent Correlation of Information: What? How?, Proceedings of the First IEEE Metadata Conference, 1996.

[10] Trusted Computer Systems Evaluation Criteria (TCSEC), US DoD 5200.28 STD, 1985.