Tamper-resistant Peer-to-Peer Storage for File Integrity Checking

Alexander Zangerl
Dipl.-Ing. (TU Wien, Austria)

A thesis submitted for the degree of
Doctor of Philosophy at
Bond University
School of Information Technology

August 2006
“... oba es gibt kan Kompromiß,
zwischen ehrlich sein und link,
a wann's no so afocch ausschaut,
und wann's noch so üblich is ...”
— Wolfgang Ambros, 1975
Abstract

One of the activities of most successful intruders of a computer system is to modify data on the victim, either to hide his/her presence and to destroy the evidence of the break-in, or to subvert the system completely and make it accessible for further abuse without triggering alarms.

File integrity checking is one common method to mitigate the effects of successful intrusions by detecting the changes an intruder makes to files on a computer system. Historically file integrity checking has been implemented using tools that operate locally on a single system, which imposes quite some restrictions regarding maintenance and scalability. Recent improvements for large scale environments have introduced trusted central servers which provide secure fingerprint storage and logging facilities, but such centralism presents some new shortcomings.

This thesis describes an alternative, decentralised approach where peer-to-peer mechanisms are used to provide fingerprint storage for file integrity checking with more flexibility and scalability than offered by currently available systems. A research implementation has been developed to verify the approach as viable and practical, and experimental results obtained with that prototype are discussed.
5 Implementation

5.1 Technological Considerations

5.1.1 Event-driven Operation

5.1.2 Code Reuse

5.1.3 Modular Design

5.2 Functional Hierarchy

5.3 User Interface

5.4 Practical Experiences

6 Analysis

6.1 Design Influences

6.2 Threats and Countermeasures

6.2.1 External Threats

6.2.2 Malicious Administrators

6.2.3 Local Threats

6.2.4 Malicious Insertion of Documents

6.2.5 Conflicting Documents

6.2.6 Communication Misbehaviour

6.3 Replication Analysis

6.3.1 Simplified Model

6.3.2 Applicability of the Simplified Model

6.4 Empirical Cost Analysis

6.4.1 Experimental Environment

6.4.2 Measurements and Expectations

6.4.3 Data Extraction and Plotting

6.4.4 Strategy Evolution

6.4.5 Naïve Flooding

6.4.6 Fan-In

6.4.7 Variable Degree of Parallelism

6.4.8 Directed Flooding

6.4.9 Biased Flooding

6.4.10 Alternative Termination Criteria

6.5 Summary

7 Conclusion
A Specifications and Source Code

A.1 Policy Language Specification
A.2 Protocol and Message Specifications
  A.2.1 Communication Overview
  A.2.2 Communication Startup
  A.2.3 Termination of Communications
  A.2.4 Message Formats
  A.2.5 Protocol Commands
  A.2.6 peerlist Format
  A.2.7 Signature Data Representation
A.3 Source Code Availability

Bibliography