Project Description:

Secure Log Files on Insecure Servers

Application Scenario

In an effort to increase the general level of security at DTU, IT services have decided to collect system log events from all major servers at DTU on a single central log server. Periodic examination of this central log file will hopefully discover signs that a server has been compromised at an earlier stage. It is therefore imperative that the log server facility is robust against compromise of both the individual servers and the central log server. In particular, the system must ensure that even if the log server is compromised, the only way the attacker can delete his tracks is by deleting the entire log file (which suggests that something untoward is happening anyway).

The central log server will collect different types of events and alarms from the different servers and store the events in a local log file (similar to a Unix syslog server). As indicated above, the log mechanism must survive the compromise of any server or any small set of servers. It must be assumed that the compromise of a server will be complete, i.e., key material and time keeping will also be compromised. The Secure Log server must ensure that an attacker cannot manipulate the content or sequence of events recorded on the central log server. Log entries from individual servers must appear the log file in nearly the same order as they are sent from the server (some reordering caused by network retransmissions must be expected). The ordering of events must also extend to the ordering of events issued by different servers.

The goal is therefore to find a cryptographic mechanism to link events in the log file in a way that protects the integrity of content and sequence. The number of events issued by each server is potentially high, which means that the mechanism must employ light weight crypto as much as possible, i.e., use hash functions before symmetric cryptography and symmetric cryptography before asymmetric cryptography.

Project Definition

Design, and evaulate a secure log system, that addresses the issues presented by the scenario outlined above. Issues that *must* be addressed are:

- Appropriate choice of cryptographic primitives to ensure the goals outlined above.
- Threat model; you should explicitly state what threats your system has been designed to handle.
- Security: Make sure that your system protects against the security issues raised in the threat model, and clearly document the threats that you do not protect against.