Bryan Parno
CS 263
10/8/03

Research Proposal: Analyze & Thwart Adversaries in the LOCKSS System

The LOCKSS (Lots Of Copies Keep Stuff Safe) project allows libraries to store and preserve electronic journals and other archival information through a system of inexpensive computers arranged in an ad hoc peer-to-peer network. The peers cooperate in "opinion polls" to detect and repair damage done to the archives. The system is intended to remain robust not only in the face of incidental data degradation and independent node failure, but also in the presence of extremely powerful, patient adversaries actively attempting to subvert the system. This last property will be crucial in faithfully preserving the records archived with LOCKSS, since a distributed system naturally opens itself to malicious attacks, particularly by those hoping to change the data stored within. For example, a tobacco company might want to alter the results of a study linking smoking with lung-cancer, or a certain Redmond-based company might wish to eliminate a pesky article establishing a researcher's patent claim. The LOCKSS system must be able to resist or at least detect such manipulation if it is to serve as a trusted repository of electronic data.

Maniatis et. al. show in simulations that the existing LOCKSS protocol either resists or reports an adversary attempting to infiltrate the system with reasonably high probability [1]. However, they use a relatively simple model for their adversary: the adversary starts with a certain proportion of compromised nodes and these nodes remain corrupted throughout the simulation. A more accurate model would incorporate an adversary that dynamically compromises hosts over time, as well as the process of administrators detecting and repairing some proportion of malign nodes. Such a

dynamic model could reveal further weaknesses (or possibly strengths) of the system and may suggest avenues for strengthening the protocol.

On a more pressing note, the authors describe how an attrition adversary can bring a system of one thousand nodes to a halt with only sixty committed nodes, and further, even a weak adversary can force a particular node to raise an alarm within nine months, thus wasting human time and effort and lowering confidence in the system as a whole [2]. This project will refine the model of the attrition adversary and develop defenses against such attacks. Such measures may involve increasing the amount of work performed by a poll initiator, constraining the time during which polls may be called or responded to (using hotel scheduling), or adding additional state on the peer to detect such attacks. The first two items may be done in an adaptive manner based on the amount of work the node is currently performing, in a manner similar to the method of random early detection used by routers to perform congestion control [3]. Additional measures to generally raise the dichotomy between the level of effort required of an attacker and that required of legitimate nodes will be explored and tested through simulations.

Finally, this project will examine whether either of the two adversaries (the stealth adversary or the attrition adversary) truly model an optimal approach for an attacker. We will either show that they do represent optimal strategies or design an alternate model that does. This will also include an investigation of the system's security guarantees at both large and small scales. How does the system behave if only a few nodes share a subscription? What happens as thousands of nodes join? Do the additional numbers increase the burden on the adversary, the legitimate nodes, or both? Ideally, this project will lead to a better understanding of these issues.

The current research plan includes the following milestones:

- Develop a deeper understanding of the current implementation both through published papers and an examination of the source code.
- Learn how to use the existing simulator.
- Develop a more dynamic stealth adversary model and implement it. Simulate the implementation and observe system behavior. Suggest possible countermeasures.
- Explore the various strategies for thwarting the attrition enemy and implement them to test their efficacy.
- Examine alternatives to or extensions of the memory-bound functions used as proofs of effort throughout the LOCKSS system.
- Analyze the attack models and attempt to demonstrate their optimality or suggest 'improvements'. Test system performance against enhanced adversaries and examine countermeasures.
- Study security guarantees as a factor of scale through analysis and simulation and patch holes discovered along the way.

References:

[1] Petros Maniatis, Mema Roussopoulos, TJ Giuli, David S. H. Rosenthal, Mary Baker, and Yanto Muliadi, "**Preserving Peer Replicas By Rate-Limited Sampled Voting**." *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, Bolton Landing, NY, October, 2003.

[2] Petros Maniatis, Mema Roussopoulos, TJ Giuli, David S. H. Rosenthal, Mary Baker, and Yanto Muliadi, "**Preserving Peer Replicas By Rate-Limited Sampled Voting**." Technical Report arXiv:cs.CR/0303026, Stanford University, March 2003.

[3] Floyd, S., and Jacobson, V., "**Random Early Detection gateways for Congestion Avoidance"** *IEEE/ACM Transactions on Networking*, V.1 N.4, August 1993