Brief Announcement: The Lotus-Eater Attack

Ian A. Kash
Computer Science Dept.
Cornell University
Ithaca, NY
kash@cs.cornell.edu

Eric J. Friedman
School of Operations Research and Information Engineering
Cornell University
Ithaca, NY
ejf27@cornell.edu

Joseph Y. Halpern
Computer Science Dept.
Cornell Univerity
Ithaca, NY
halpern@cs.cornell.edu

Categories and Subject Descriptors: C.4. Performance of Systems: Fault Tolerance

General Terms: Economics, Reliability

1. INTRODUCTION

Many current distributed systems are *satiable*; users will stop providing service to others if they are themselves receiving a sufficient quantity of service. This is often the product of "tit-for-tat-like" designs, which attempt to combat free riding by denying service to those who are not providing it. While this approach provides an incentive for cooperation, it has the unfortunate side effect that if there is no service for a peer to provide, then he will generally receive reduced or no service. Ironically, this opens the systems up to an attack that we call the *lotus-eater attack*: the attacker supplies the service to some peers, thus satiating them. Once those peers are satiated, they stop providing service to others. The peers not being satiated by the attacker then receive reduced or no service.

A wide range of systems are satiable and thus potentially vulnerable to this attack. In BitTorrent [1], peers restrict trade to the best partners they can find. Similarly, in BAR Gossip [3] peers stop trading when there is nothing they want. In scrip systems (e.g. [4]), peers need to perform service for others often enough to maintain a supply of money and will stop once their supply is adequate. Even systems not designed to be tit-for-tat-like may be satiable. For example, a node in a sensor network might shut down to save power if it has received all the updates it needs.

2. ATTACKING BAR GOSSIP

In BAR Gossip, a broadcaster releases updates that nodes need to collect by a deadline. For example, in a streaming video application, the updates are frames of the video. To mount an attack on BAR Gossip, the attacker divides the nodes into two groups: *satiated nodes*, to whom the attacker attempts to provide as many updates as possible, and *isolated nodes*, to whom the attacker provides no service. Since nodes have a fixed number of trade opportunities and only trade updates when they expect to receive something in return, isolated nodes will have a hard time trading.

Figure 1 shows the results of three versions of the attack on a BAR Gossip system using the same parameters as and an updated version of the simulation from [3]. Nodes needed to receive more than 93% of the updates for the stream to be usable. The results in Figure 1 are for the 30% of nodes that are isolated; satiated nodes receive near perfect service. The curve labeled "crash attack" provides a baseline where the attacker simply does nothing. The "ideal lotus-eater attack" curve assumes that attacking nodes can immediately send updates to all satiated nodes as soon as they receive them. The "trade lotus-eater attack" curve makes the typically more reasonable assumption that the attacker can give updates to nodes only during interactions dictated by the protocol.

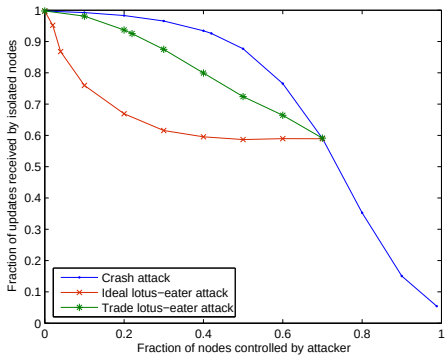

Figure 1: Three attacks on BAR Gossip.

These results suggest that lotus-eater attacks are a potential problem for BAR Gossip. The need for a smaller fraction under the attacker's control may make it possible to launch a lotus-eater attack in some settings where a standard Byzantine crash attack would be impossible. We should note, however, that this does require enough bandwidth at each attacking node to satiate multiple nodes every round while the crash attack requires essentially no bandwidth beyond that needed to maintain the nodes in the system. Additional results and discussion are available in [2]

Acknowledgements

We would like to thank Harry Li and Lorenzo Alvisi for their BAR Gossip simulation. EF, IK and JH are supported in part by NSF grant ITR-0325453. JH is also supported in part by NSF grant IIS-0534064 and by AFOSR grant FA9550-05-1-0055.

3. REFERENCES

[1] B. Cohen. Incentives build robustness in BitTorrent. In *First Workshop on the Economics of Peer-to-Peer Systems (P2PECON)*, 2003.

[2] I. A. Kash, E. J. Friedman, and J. Y. Halpern. The lotus-eater attack. arXiv:0806.1711.

[3] H. C. Li, A. Clement, E. L. Wong, J. Napper, I. Roy, L. Alvisi, and M. Dahlin. BAR gossip. In *Sixth Symposium on Operating Systems Design and Implementation (OSDI)*, pages 191–204, 2006.

[4] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer. KARMA: a secure economic framework for peer-to-peer resource sharing. In *First Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, 2003.

PODC'08, August 18–21, 2008, Toronto, Ontario, Canada.
ACM 978-1-59593-989-0/08/08.