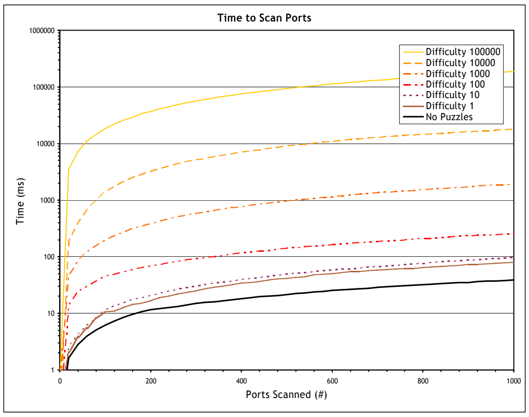
Reducing Malicious Traffic With IP Puzzles [*]

Ed Kaiser Wu-chang Feng Wu-chi Feng Antoine Luu
OGI@OHSU ENSEIRB
{*edkaiser, wuchang, wuchi*}*@cse.ogi.edu* *luu@enseirb.fr*

Amidst the traffic of the Internet is an enormous amount of undesirable communication. Currently there is no significant disincentive for clients who contribute to this flood of undesirable communication. A mechanism for punishing only the malicious is required in order to discourage clients from behaving badly. The standard response has been to disconnect clients exhibiting suspicious behavior from the rest of the network using a binary filter. Ideally though, the mechanism should be analog to allow falsely identified clients to prove that they are legitimate, so that service to them can be reinstated. Client puzzles have been proposed in several protocols as a mechanism well suited for this task; clients do all the work involved in proving their legitimacy.

Until now, client puzzles have been used only as an application layer defense against flooding attacks. Yet, client puzzles in the application layer can be thwarted if any adjacent or underlying protocol does not provide a similar defense. For example, DoS-resistant authentication protocols can be thwarted by basic UDP or IP flooding. Implementing client puzzles in TCP offers no protection from those flooding attacks either. Clearly, the network layer is the lowest layer vulnerable to distributed network flooding attacks. We argue that the network layer is the most defensible layer against flooding and other forms of distributed attacks. This poster describes the design and implementation of our *network layer* client puzzle protocol.

In addition to distributed flooding attacks, network layer puzzles can defend against attacks which have been undefendable until now. As an example, in-network reconaisance attacks such as port scans cannot be defended at higher layers; by the time any higher layer becomes aware of the attack, the attacker has obtained all the information sought. It is important to stop reconaisance attacks since the information gathered is used by worms to create the most potent attack topology possible, which means the difference between a severe network outage or a brief network congestion. We show that network puzzles can effectively throttle port scans.



When deploying a protocol in the network layer, it is important to be flexible about which devices must participate. Our protocol, which can be implemented within the fast path of network hardware, gives every network device along the path from the client to the server the choice of participating as a puzzle issuer or not. Being able to place puzzle issuers arbitrarily close to the client allows quenching undesirable traffic closer to the source, wasting fewer resources deeper within the network.

Another issue with puzzles is that the work load required for each puzzle must be adjusted to meet the real-time needs of a network. Throttling a malicious client cannot be done simply during the establishment of a connection; the need for puzzles may start and stop at any point in a flow's lifetime. Further, the puzzle difficulty required to keep services at maximum utilization changes frequently during the lifetime of any flow.

This poster describes our protocol as well as an `iptables` implementation that addresses these challenges. More information about the project can be found at:
`http://www.cse.ogi.edu/sysl/projects/puzzles`

[*] This material is supported in part by the National Science Foundation under Grant ANI-0230960 and the generous donations of Intel Corporation. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or Intel.