A Critique of

"A Defense Against Address Spoofing Using Active Networks"

for UCSB CS 290I Winter 1999

Mihai Christodorescu

This thesis paper approaches the SYN–flooding Internet attack and proposes a solution based on Active Networks. They describe their implementation of Active Networks, and then a network service that will perform filtering and that will move itself down the path towards the attacking host.

The architecture of ANTS (their Active Networks toolkit) is structurally similar to TCP/IP. On top of TCP/IP, ANTS builds a software component distribution system. The software components can be addressed from packets traveling over the network. When a packet arrives at an active node, the node finds and executes the code specified in the packet. Since IP routing is used, packets from the same message stream might travel different routes to the destination. This might impose a unreasonable slow start for communication, until the majority of the nodes are populated with the necessary code. Also, software components might increase the congestion in a network, by at least a factor of 2 (for each packet, a software component has to be sent across, if the node cannot cache all the software components). This topics are not covered in the paper. There are several implicit assumptions in the paper: much of the code can be shared between data streams, and processing is not always necessary.

A hierarchy of software components is defined, with the understanding that sibling code capsule are related to the same protocol. A special hierarchy is reserved for the ANTS communication. There is not way to authenticate or verify capsule downloaded from another site, which would make it easy for a malevolent user to create denial–of–service attacks. Security is lightly touched upon, with only a couple of suggestions on the requirements needed to deploy such an infrastructure.

ANTS is based on a symmetric routing constraint. Although it simplifies design of user capsules, symmetric routing has several

drawbacks: first of all, each node has to maintain some state information about the routes it is part of. Second, symmetric routing does not seem to adjust seamlessly to network topology changes (routes / hosts going up and down, links cut off, new links introduced). Third, today the Internet does not support symmetric routing, and this might complicate or slow down deployment of ANTS.

An interesting point present in the ANTS architecture is the mixing of host–based services (present as an object in the active node cache), and of packet–based services (each packet has codes identifying the component it needs to run upon arrival on an active node). This mix seems to be a welcome tradeoff between the two extremes (code only on host and code only in packets).

The service that is supposed to stop SYN–flooding is title "Dynamic Filtering." It works by deploying a packet filter when the rate of incoming SYN packets increase over a given threshold. The packet filter tries to extract information about the attacker. Each code capsule (packet) contains information about the sender in two places: the regular IP header (which can be easily forged), and the ANTS header fields (which is presumed hard to modify / forge). The filter works only on SYN TCP packets for hosts that demanded filtering by sending a filter capsule to the current node. The filtering algorithm uses the ANTS source information and the symmetric routing property to determine the next hop on the route back to the source. If the previous hop is not found or is different than the one the capsule came from, the capsule is presumed forged and dropped. In addition to this step, the filter sends a request to the previous hop to start filtering there. This way filtering is pushed towards the attacker as much as possible.

The paper emphasizes that this filter is lightweight compared to other proposals, specifically the Ingress filtering algorithms. This property is achieved at the cost of overall loading of the network: each active node has to perform processing, and the network with have to support extra load needed to transport the code necessary for capsule processing.

There are not reports on the results, and a comparison between today's Internet and the ANTS network, in terms of RTT and jitter, would be useful.