SOS: Secure Overlay Services

Paper by Angelos D. Keromytis, Vishal Misra and Dan Rubenstein

Oleg Ponomarev
Research seminar on trust, security and privacy in the Internet, 25.10.2006
Helsinki, HIIT
Overview

- Related work
- SOS architecture
- Performance Analysis
- Discussion
Motivation

• Denial of service (DoS) attacks threaten the reliability of networking systems. Reactive approaches addressing this problem wait for an attack, identify it and try to block the offenders
• Methods looking for known attack patterns or statistical anomalies can be defeated by changing the attack pattern, distributing it and masking the anomalies
• Since most attacks use random source IP addresses, the only reliable packet field that can be used for filtering is the destination IP address (of the target) => filters impact valid communications of the attack target
• …
Introduction to SOS

• We focus on protecting a site that stores information that is difficult to replicate due to security concerns or its dynamic nature.

• There is a pre-determined subset of clients scattered throughout the WAN (users in the field) who require (and should have) access to the information.

• There is a set of users (attackers) that want to prevent access to this information and will launch DoS attacks upon any network points whose jamming will achieve this goal.

• May be used by emergency response teams, telecommuters, corporate Intranets, subscribers to a news website, etc.
Basic SOS architecture

- Only small set of “approved” source addresses pass through the filter. They are kept secret
- Any transmission must be validated at entry points of the overlay
- Traffic is tunneled securely inside the overlay

* picture courtesy of the paper authors
**SOS assumptions**

- The attacker does not have access to the network core, i.e. “approved” IP’s are secret
- The attacker cannot acquire sufficient resources to severely disrupt large portions of the backbone itself, e.g. to congest all the paths to the target

* picture courtesy of the paper authors
To prevent the attacker from bombarding the target with packets, a filter drops illegitimate packets at some point in the network.

Filtering is done at a set of high-powered routers (typically at the ISP’s POP) that can handle high loads of traffic. They are more difficult to attack.
SOS: Reaching a well-filtered Target

• The target selects a subset of nodes, $N_s$, that participate in the SOS overlay to act as forwarding proxies. The filters allow packets only from $N_s$ (by the source address or more sophisticated techniques).
• The architecture hides the identities of the proxies, the “hidden” proxies are referred to as secret servlets.

* picture courtesy of the paper authors
To allow legitimate users to access the overlay we define a secure overlay access point.

A SOAP receives packets that have not yet been verified as legitimate and performs this verification using e.g. IPsec or TLS.

Having a large number of SOAPs increases the robustness, but complicates the maintenance.

* picture courtesy of the paper authors
• Selecting the next node at random is sufficient to eventually reach a secret servlet, but it is $O(N/N_s)$ long.

• With only one additional node knowing the identity of the secret servlet, the length is $O(\log N)$ by using Chord service for the lookup (more on the next slide).

* picture courtesy of the paper authors
Chord: A Scalable Peer-to-peer lookup Service for Internet Applications

- The Chord protocol supports just one operation: given a key, it maps the key onto a node.
- In the steady state, in an N-node network, each node maintains routing information for only $O(\log N)$ other nodes, and resolves all lookups via $O(\log N)$ messages to other nodes.
- Chord scales well with the number of nodes, recovers from large numbers of simultaneous node failures and joins, and answers most lookups correctly even during recovery.

* pictures courtesy of Ion Stoica et al.
The node in the chord that stores the hash of the target identifier (its IP) is referred as the *beacon*.

The secret servlet hashes the target identifier and informs the beacon (using chord routing) of the secret servlet’s identity.

SOAPs use the hash of the target identifier as the key to reach the beacon, they do not know the target id.
SOS: Redundancy

- Each component is easily replicated within SOS
- Any overlay node can act as a SOAP as long as it is able to check the legitimacy of a user
- The target can choose multiple nodes as secret servlets
- Multiple nodes can act as beacons via applying several hash functions

*picture courtesy of the paper authors
A Static Attack

Varying number of attackers (x-axis) and nodes in the overlay (curve)

- 10 possible access points
- 10 beacons
- 10 secret servlets

Varying number of beacons (x-axis) and secret servlets (curve)

- 10000 nodes in the overlay
- 1000 attackers

* picture courtesy of the paper authors
Dynamic Attacks and Recovery (1/2)

1000 nodes in the overlay
10 secret servlets
10 beacons
10 SOAPs

We assume $D_a$ (attack delay) and $D_r$ (repair delay) are exponentially distributed random variables with rates $\lambda$ and $\mu$. $\rho = \lambda / \mu$ is varied along x-axis. Different curves are different number of attackers.

* picture courtesy of the paper authors
Dynamic Attacks and Recovery (2/2)

We assume $D_a$ (attack delay) and $D_r$ (repair delay) are exponentially distributed random variables with rates $\lambda$ and $\mu$. $\rho = \lambda/\mu$ is varied along x-axis.

Different curves are different number of attackers.

* picture courtesy of the paper authors

1000 nodes in the overlay
10 secret servlets
10 beacons
10 SOAPs
Attacking the Underlying Network

Blocking probability for legitimate traffic as a function of attack traffic load

Performance gain with increasing the capacity of the attacked node

* picture courtesy of the paper authors
Issues that have to be addressed

• **Attacks from inside the overlay.** An evaluation of potential damage that can be done from inside and approaches to limit it.

• **A shared overlay.** The overlay becomes more efficient at protecting users from DoS attacks as it grows, it would be of interest for multiple organizations to utilize a shared overlay, but a breach in one organization’s security system should not lead to breaches in other networks.

• **Timely delivery.** Preliminary simulations have shown the latency to be in the order of 10 times.

• **Better analysis.** More detailed model or through prototype and experimentation.
Summary

• We addresses the problem of securing a communication service on top of the existing IP infrastructure from DoS attacks

• We use proactive mechanism, which is composed of aggressive packet filtering in a site’s network periphery, an overlay network that can self-heal during (and after) a DoS attack, and a scalable access control mechanism that allows legitimate users to use the overlay network

• Through analytical models we show that DoS attacks directed against any part of the SOS infrastructure have negligible probability of disrupting the communication between the parties
The End