

# Protecting Public Servers from DDoS Attacks Using Drifting Overlays

Venkata K. Pingali  
USC/Information Sciences Institute  
Marina del Rey, CA, USA  
pingali@isi.edu

Joseph D. Touch  
USC/Information Sciences Institute  
Marina del Rey, CA, USA  
touch@isi.edu

**Abstract**— Drifting Overlays are dynamic partial network-layer overlays with traffic ‘safe houses’ that enterprises can use to control, at fine granularity, the reachability and predictability of paths taken to important hosts. Drifting Overlays enable enterprises a level of control over their own DDoS defenses and routing choices, rather than leaving them at the mercy of their ISPs.

**Keywords**- VPNs, Security, Deployment, Automation

## I. INTRODUCTION

Network customers often need some control over how their ISPs handle their traffic. When the ISP provides alternative peering points, customers may want to manage how their own traffic leaves, or where it enters, at a fine granularity to avoid Distributed Denial of Service (DDoS) attacks (for incoming traffic) or for policy or performance (for outgoing traffic). This is normally accomplished by route peering relationships with the customer. Unfortunately, this is coarse grained, expensive and not always an available option; there are ISPs who do not support peering with small customers, and peering support doesn’t always propagate beyond the first ISP. Customers connected via a single ISP often lack control over their own network traffic. ISPs neither allow local control of routing, nor can they support per-customer DDoS defenses. ISPs aggregate control and management, and self-managed routing defeats this.

Drifting Overlays enable enterprises a level of control over their own DDoS defenses and routing choices at fine granularity, rather than leaving them at the mercy of their ISPs. The basic architecture starts with “safe houses” - sites on other ISPs, or distributed within an ISP, which are under the limited control of the customer (Fig. 1). These sites are used to redirect traffic to specific host(s). The safe house provides a type of tethered remote network interface, allowing the customer’s traffic to appear as if it originates at any of these remote sites. As a result, traffic terminates its ISP-routed path at the safe house, which results in different traffic paths than are possible from the local customer’s site alone, without needing to relocate customer resource. Traffic between the safe houses and the customer host(s) is carried over network-layer multi-hop IP overlay. A key feature of this overlay is that it drifts over time *i.e.*, the overlay is continuously modified in an open-ended fashion for security, performance and other reasons. The ability to modify is built into the overlay and the strategy used to modify is determined by the enterprise locally. The attacker

must now accurately guess the strategy to be able to reach a server. For some configurations, it may not be useful to even know the strategy.

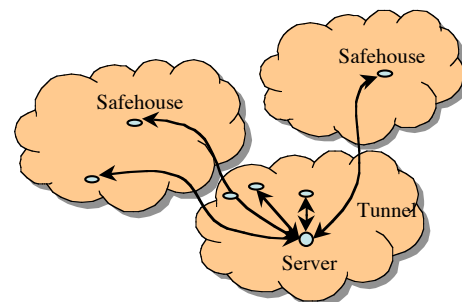


Figure 1. A combination of continuously evolving IP overlay and hosts (called safehouses) are used to direct traffic to and from public servers.

## II. FLEXIBLE ROUTES

Enterprises need flexible route control in addition to existing mechanisms such as filtering, rate control, fairness and firewalling. The need arises from the desire to better utilize the resources and address the asymmetry in capability of the attacker and the enterprise. *First, reachability must be controlled.* Almost all the existing DDoS solutions assume the server is globally reachable by default and then address the possibility of attacks. Some sites might prefer the converse, *i.e.*, a server is globally unreachable by default but reachable through appropriate enabling mechanisms such as tunneling. In case of the reachable by default model, no amount of filtering is sufficient because the attacker has far more capacity to generate traffic than the enterprise can handle and can increasingly mimic non-attack traffic accurately. Security through server address obscurity is not really an option because the attackers’ sophistication is growing with time and it takes only one security breach for the server address to be known. In case of the unreachable by default mode, even if the enabling mechanisms, such as the safe houses mentioned above, are compromised, they are relatively few in number and under local control. *Second the cost must be payable where necessary.* All clients are not equal, and disruption of traffic to and from certain destinations is acceptable and not from some others. All servers are not equal either. Some need more protection than others. Further this changes with time. It should be deployable anywhere in the network with minimal additional configuration/disruption of hosts around. *Third, the cost must*

be payable *when necessary*. The solutions must be simple to enough to allow fast deployment when necessary and have non-linear impact to make best use of each incremental resource. Solutions that require cooperation of many hosts/routers in the network, extensive reconfiguration and/or large amounts of state are unlikely to be deployable quickly. Further, a major problem with some existing solutions is that they require the DDoS defense designed for worst-case scenario to be the common case resulting in poor utilization of resources. *Last, the requirement of each enterprise is different*. The solution must be simple enough to be rapidly customizable to each site. The amount of resources, granularity and strength of the protection must all be configurable.

There is no one solution that meets all the needs above that is simple, scalable and protects all communication. Drifting Overlays allows tradeoffs to be made at the enterprise level through a simple framework that provides a degree of control over visibility, reachability and predictability of the paths taken to and from important public servers. The continuous change over time could potentially invalidate the information that the attacker has regarding the location of the safe houses and/or the paths. This has the effect of changing the nature of the attack from one of physical resources to one of information. It is much easier to design defenses for information attacks. This capability, however, comes at the cost of increased protocol overhead, additional hosts and management complexity.

Drifting Overlays, however, do not completely eliminate the problem of DDoS – especially those that are capable of engaging network ingress and egress routers. A successful defense of the server will force the attacker to focus on other services upon which the server is functionally dependent such as safe houses, routing, name lookup, and databases. The dependent services could potentially have lesser complexity and/or value. The safe houses, for example, are not expected to have any critical data and are easily replaceable. Similarly, attacks on routing and name lookup services have been studied extensively and effective mechanisms exist including replication [18] and line-rate filtering.

### III. ARCHITECTURE

The Drifting overlay architecture consists of: (1) a network of nodes that are either safe houses or servers, and links that IP-IP tunnels (2) a low-level tunnel management service that includes a coordination entity, called tunnel manager, and a protocol to establish and destroy tunnels between nodes (3) a topology management service that provides a higher level management abstraction.

Safe houses are either public or private depending on whether their existence is global information or not, and active or passive depending on whether or not they participate in the overlay. Servers/hosts are advertised as being only reachable through active public safe houses, typically through the DNS. Active private safe houses are used either for routing traffic between safe houses or as edge nodes co-located with important clients. There is no requirement that private safe houses or the server be globally reachable. However, all safe houses and hosts must be reachable from the tunnel management service. The server must be reachable from all the

safe houses within the overlay. Safe houses may be available for selection as a commercial service similar to one provided by Akamai [12]. The safe houses that are available for use must be discovered at runtime. There are many options for the same including a distributed host registry [15].

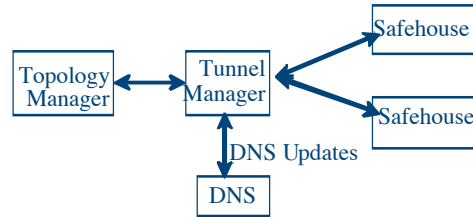


Figure 2. Architecture of the Drifting Overlay System

The tunnel manager coordinates the process of link establishment and route computation. It separates the network construction mechanism (how) from the policy (what). The tunnel manager provides a simple set of primitives to the policy module (topology manager) such as “discover nodes,” “add node,” “add link,” “delete node,” and “delete link”. Link properties include (1) tunnel endpoint addresses, (2) security (IPsec) and (3) QoS (delay, bandwidth). The overlay may use static or dynamic routing. In case of the former, the tunnel manager must also compute end-to-end routing information.

The topology manager translates high-level commands from the administrator into sequences of calls to the tunnel manager. The commands may specify overlay attributes such as the topology, node selection criteria and reconfiguration strategy. The strategies may vary in complexity, reactivity and resultant “strength” of the deployment. Because the number of safe houses is expected to be relatively small, the algorithms are not required to scale. The strategies are expected to be reusable across networks

### IV. RELATED WORK

The various DDoS prevention solutions can be seen as a combination of resource duplication a strategy to exploit that redundancy and traffic management. The resource is typically hosts, paths or information (e.g., addresses). The strategy in most cases is either proactive hard-to-predict resource instance selection or reactive fault tolerance mechanisms. Both can be combined with marking of legitimate traffic and associated traffic management at nodes providing fairness and QoS guarantees.

Host-based solutions use replication and state transfers to reduce the impact of an attack. Examples of simple host-based solutions include server roaming [6], roaming honeypots [7] and MOVE [9]. State transfers across hosts in real time tend to be complex and require client cooperation. Drifting Overlays lets the client decide the level of cooperation and could potentially incorporate server migration mechanisms.

Path-based solutions on the other hand, use alternative paths to reach the server when one or more paths are attacked. Examples of path-based solutions include Secure Overlay Services (SOS) and its derivatives [9][10][11] and Mayday [8]. Again these approaches are resource intensive and all extra nodes are participating in traffic distribution all the time. The

local control over topology is limited to selecting to DHT routing parameters. Drifting Overlays uses traditional intra-domain routing algorithms and allows control over topology. OverDoSe [19] extends this work by providing resilience against overlay node compromise and fairness after the connection is established. Drifting Overlays can incorporate OverDoSe mechanisms. FONet [20] suggests construction of a single global shared overlay, instead of a per-server overlay, to share the costs. FONet does not quite eliminate the problem for customers whose ISP does not host FONet nodes and/or if FONet nodes are globally reachable at IP layer as would be the case during initial phases of deployment. This creates a significant barrier to deployment.

Approaches that exploit information such as IP address [16], and frequency [15], have existed for a long time and in general are very effective. Information-based solutions fundamentally depends on resource redundancy such as addresses, frequency etc. Drifting Overlays introduces redundancy and randomization in paths that turn DDoS attack into information attacks.

The basic solution structure of Firebreak [14] is similar to that of Drifting Overlays. Each server is associated with a set of firebreak hosts distributed on the Internet that are similar to safe houses. The emphasis of Firebreak is on the mechanism for routing of traffic to and from the firebreak hosts (anycast) whereas the emphasis of Drifting Overlays is on the tunneling component of the solution. The solutions are complementary to a large extent. Anycast mechanism could be used to route traffic to the safe houses in the Internet. The dynamic overlay concept can be adapted to the Firebreak environment.

The Akamai Edge Service [12] provides an application-level DDoS attack prevention service and a successful business model. Drifting Overlays' reasoning is similar to the Edge Service but operating at network-level. Drifting Overlays' topology management strategies can be adapted to Akamai's environment.

Drifting Overlays are based on previous work on Dynabone [3] and TetherNet [1]. Dynabone achieves fault tolerance through sophisticated deployment and runtime selection of IP overlays for data transmission. Drifting Overlay uses a single one-layer simple overlay that is modified over time. TetherNet is a special case of a virtual private network (VPN) in which a sub-network is reachable from the rest of the Internet at a chosen point in the global address space. While TetherNet was trying to address the unreachability, Drifting Overlays exploits that very unreachability to force traffic along controlled paths and can be considered to be a network of TetherNet links.

## V. STATUS

A prototype system is under development on FreeBSD platform. The prototype uses a centralized tunnel and topology manager that coordinate the construction of tunnels, routing entries and DNS updates (Fig 2). A control daemon runs on each of the safe houses and receives and executes instructions from the tunnel manager. The system uses IP tunneling, static routing tables and address randomization. Tunnel addresses are chosen from a large private address space (10.0.0.0/8). The manager ensures significant delay in time

before reuse of tunnel addresses and consistent timing relationships between the DNS caching time, client end of the configuration and the server end configuration. Early experimentation showed that the header matching in the kernel is a bottleneck when a large number of tunnels are created. Much work remains to be done in terms of topology choices, reconfiguration strategies, characterization of the impact of an attack and deployment issues.

## VI. REFERENCES

- [1] TetherNet web pages, <http://www.isi.edu/tethernet>
- [2] X-Bone web pages, <http://www.isi.edu/xbone>
- [3] DynaBone web pages, <http://www.isi.edu/dynabone>
- [4] J. Touch, "Dynamic internet overlay deployment and management using the X-Bone", *Computer Networks*, Jul. 2001, pp. 117-135.
- [5] J. Touch, Y. Wang, and L. Eggert, "Virtual Internets," *ISI Technical Report ISI-TR-2002-558*, July 2002.
- [6] S. M. Khattab, C. Sangpachatanaruk, R. Melhem, D. Mosse, and T. Znati, "Proactive server roaming for mitigating denial-of-service attacks," *Proceedings of International Conference on Information Technology: Research and Education, 2003 (ITRE2003)*. 11-13 Aug. 2003 Page(s):286 - 290
- [7] S. M. Khattab, C. Sangpachatanaruk, D. Moss, R. Melhem, T. Znati. "Roaming honeypots for mitigating service-level denial-of-service attacks," *ICDCS*, pp. 328-337, 2004.
- [8] D. G. Andersen, "Mayday: distributed filtering for internet services," In *Proc. USENIX Symposium on Internet Technologies and Systems (USITS)*, March 2003.
- [9] A. Stavrou, A. D. Keromytis, J. Nieh, V. Misra, D. Rubenstein "MOVE: an end-to-end solution to network denial of service," *Proceedings of the Internet Society (ISOC) Symposium on Network and Distributed Systems Security (SNDSS)*. San Diego, CA, February 2005.
- [10] A. Keromytis, V. Misra, and D. Rubenstein, "SOS: Secure Overlay Services," In *Proceedings of ACM SIGCOMM '02*, Pittsburgh, PA, August, 2002.
- [11] D. L. Cook, W. G. Morein, A. D. Keromytis, V. Misra, and D. Rubenstein, "WebSOS: protecting web servers from DDoS attacks," In *Proceedings of the 11th IEEE International Conference on Networks (ICON)*, pp. 455 - 460. September/October 2003, Sydney, Australia.
- [12] Akamai Edge Computing <http://www.akamai.com>
- [13] J. Touch, Y. Wang, V. Pingali, L. Eggert, R. Zhou, G. Finn. "A Global X-Bone for network experiments," *Proc. IEEE Tridentcom 2005*, Trento Italy, Mar. 2005, pp. 194-203.
- [14] P. Francis, "Firebreak: an IP perimeter defense architecture," *Webpage* <http://www.cs.cornell.edu/People/francis/firebreak/>
- [15] A. Ephremides, J. E. Wieselthier, D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol.75, no.1pp. 56- 73, Jan. 1987
- [16] J. Jones, "Distributed denial of service attacks: defenses," *A Special Publication, Technical Report, Global Integrity*, 2000.
- [17] N. Fujita, J. Touch, V. Pingali and Y. Wang, "P2P-XBone: a virtual network support for peer-to-peer systems," *Technical Report ISI-TR-2005-607, USC/ISI*, September 2005.
- [18] R. Naraine "Massive DDoS attack hit DNS Root Servers," [www.internetnews.com/dev-news/article.php/1486981](http://www.internetnews.com/dev-news/article.php/1486981), Oct 2002.
- [19] E. Shi, I. Stoica, D. Andersen, and A.Perrig, "OverDoSe: a generic DDoS protection service using an overlay network," *CMU Technical Report CMU-CS-06-114*, February 2006.
- [20] J. Kurian and K. Sarac, "FONet: a federated overlay network for DoS defense in the Internet (a position paper)," *Global Internet Symposium, Barcelona, Spain, April 28-29, 2006*.