Proc. 3rd DARPA Information Survivability Conference and Exposition (DISCEX-III),
Washington, DC, USA, April 22-24, 2003, Vol. 2, pp. 271-276.

DynaBone: Dynamic Defense Using Multi-layer Internet Overlays

Joseph D. Touch, Gregory G. Finn, Yu-Shun Wang, Lars Eggert
USC/Information Sciences Institute
{touch, finn, yushunwa, larse}@isi.edu

Abstract[1]

DynaBone provides a dynamic defense against distributed denial-of-service attacks among private groups of networked systems. It uses multi-layer Internet overlays to apply encryption, routing, and configuration diversity, providing multiple alternate networks over which traffic is automatically routed. The DynaBone software has been implemented and demonstrated to utilize as many as 50 concurrent interior networks, while providing a single network view to the end systems and applications.

1. Introduction

Distributed Denial of Service (DDOS) attacks are becoming a more prevalent threat to network service [3]. There are a number of defenses deployed to resist DDOS attacks, including encryption of packets, firewalls to disable traffic from untrusted sources, and specific filtering of packets with known attack signatures [8]. Ultimately, the use of these defenses presents a trade-off between network service and security (Figure 1).

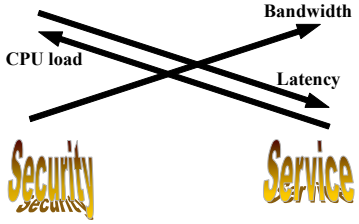

Figure 1 Spectrum of security and service, and its effect on system performance

[1] Effort sponsored by the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number F30602-01-2-0529 entitled "DynaBone". The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory, or the U.S. Government.

Typically, individual DDOS solutions trade service level for security, resulting in overall decreased service performance. Further, each single DDOS solution presents a target for subsequent attacks; the more successful or pervasive the solution or system, the more potential targets it presents, and the higher the payoff to the attacker. Clearly:

Distributed attacks necessitate distributed defenses

A common example of distributed defense is layered security, where a combination of 'locks' together provides a stronger defense than any one individual lock. However such layering exacerbates the trade-offs in Figure 1; each layer of lock results in additional delays, CPU load, and decreased throughput.

DynaBone provides an alternate approach, deploying a variety of different DDOS defenses in parallel overlay networks, and scattering packets on these overlays based on their defense status and throughput (Figure 2). The result is a dynamic backbone (a DynaBone) that provides DDOS resistance with increased performance and the ability to react to attacks.

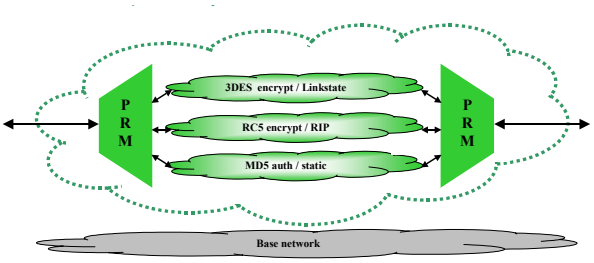

Figure 2 DynaBone dynamic parallel overlays

DynaBone's parallel interior overlays, known as *innerlays*, provide multiple targets for DDOS attacks while providing a single, coherent network service as an outer overlay (an *outerlay*). The proactive/reactive multiplexer (*PRM*) uses attack statistics and performance monitoring statistics to decide how to distribute packets among the innerlays. Innerlays under attack are disconnected (unused); the remaining innerlays are used in proportion to their performance.

DynaBone thus allows innerlays under attack to be disabled. Concurrent use of the remaining multiple innerlays for remaining traffic makes DDOS attacks more difficult, and allows overall service to degrade gracefully. Weaknesses of individual protocols or attacks on specific addresses undermine only one innerlay at a time; service is completely disabled only when all innerlays are successfully attacked simultaneously. It is more difficult for service to be completely disabled, and service is automatically restored and reconstituted by shifting traffic to the remaining innerlays.

There are other, additional capabilities enabled by the DynaBone architecture. The use of different innerlays and nonuniform distribution of traffic could present information to an attacker, enabling them to interrupt service on the most heavily used innerlay. Although this does not completely compromise the overall architecture, it could result in slower restoration of original service. Because the innerlays are deployed in a coordinated fashion, traffic confidentiality techniques, e.g., hiding real packets in random packet streams, can be more easily deployed to defeat such information gathering. Other techniques, such as honeypot deployment, dynamic addition of new innerlays, and relocation of critical services are also enabled by this architecture.

2. Approach

Distributed denial-of-service (DDOS) attacks are an increasing threat to wide-scale network infrastructure. The simplest solution to these attacks is to disconnect the attacked sites from the network. This has the benefit of stopping the attack, but the detriment of removing network service for those sites. Disconnected hosts can be reconnected at other locations in the network; disconnected routers are somewhat more difficult to 'replace' or relocate.

DDOS defenses using disconnection vary in scope. They include 'air-gap' security, firewalls, and the use of multihoming to provide redundant connectivity [6]. DynaBone extends these paradigms, providing automated management of multi-connected hosts and routers, allowing DDOS responses to include disconnection from attacked networks without completely severing network connectivity.

DynaBone extends the X-Bone overlay deployment system to configure and coordinate a set of concurrent, parallel innerlays, and distribute traffic among them dynamically [12][14]. Internally, it applies feedback-controlled trunk grouping to distribute traffic across innerlay networks based on a variety of security, performance, and attack status parameters [2][9][16]. It applies a variety of innerlay configurations that have different performance and security metrics, providing configuration and algorithmic diversity.

The following sections of the approach provide an overview of overlay networking in general, the X-Bone overlay deployment system and its specific capabilities, and the architecture and features of the DynaBone.

2.1. Overlays

An overlay network is an isolated virtual network deployed over an existing network. It is composed of hosts, routers, and tunnels. Tunnels are paths in the base network, and links in the overlay network. Hosts are packet sources or sinks, and routers are packet transits, as in conventional networks. Individual components (routers or hosts) can participate in more than one overlay at a time or in multiple ways (router, host) in a single overlay. Figure 3 shows an IP network (left); on that network, a deployed ring (center) and star (right), using various subsets of the nodes of the base network, connected by a set of tunnels. These tunnels determine the overlay topology, and may traverse multiple links in the base network, or a single link multiple times.

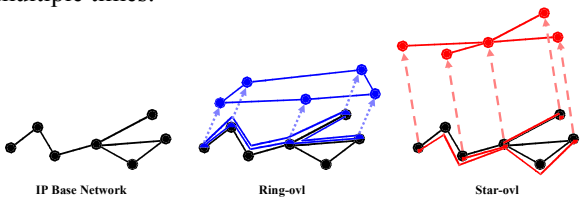

Figure 3 A ring (center) and a star (right) overlay deployed on a base network (left)

Overlays have three primary uses: containment, provisioning, and abstraction. Containment is the ability of an overlay to restrict the visibility of its contents. Tunneling encapsulates the packets of new protocol so it can be tested in a controlled environment. Containment was one of the first uses of overlays in the early 1980's, and motivated their re-emergence in the early 1990's for the M-Bone and later 6-Bone [7]. Tunnels allow incremental deployment, where (primarily) routers lacking new protocol capabilities can be skipped over (or through), avoiding the need for contiguous availability.

Provisioning uses reservation of components and capacity along tunnels to provide service guarantees to

the overlay. Provisioned overlays can be used during emergencies to create virtual infrastructure when it is not feasible to deploy new physical resources. They can also limit the scope and impact of network experiments, e.g., limiting them to nominal use of surplus capacity.

Abstraction is a new use of overlay networks. Both provisioning and containment imply the interim the use of overlays that will be supplanted by advanced hierarchical reservation in the former case, or more sophisticated dynamic service deployment in the latter. In these cases, overlays provide such capabilities without requiring contiguous deployment; once a new protocol or service is ubiquitous, tunnels (and thus overlays) could be avoided. However, abstraction remains a useful tool for education (networking classes), deploying testbeds, and, most importantly, simplifying applications. For example, a single lab can support a large number of concurrent experiments, each using a different topology. A testbed can be configured using a graphical user interface, in *do what I mean* style. Applications can request a deployed topology (e.g., ring) without needing to incorporate network management. In each case, manual intervention by a network manager is avoided, and applications and tools can be simplified.

2.2. X-Bone

The X-Bone is a system for the dynamic deployment and management of Internet overlay networks [12][14][15]. Overlay networks are used to deploy infrastructure on top of existing networks, to isolate tests of new protocols, partition capacity, or present an environment with a simplified topology. Current overlay systems include commercial virtual private networks (VPNs), IP tunneled networks (M-Bone, 6-Bone), and emerging research systems providing quality-of-service guarantees [7][10]. The X-Bone system provides a high-level interface where users or applications request DWIM (do what I mean) deployment, e.g.: *create an overlay of 6 routers in a ring, each with 2 hosts*. The X-Bone automatically discovers available components, configures, and monitors them.

The X-Bone uses a two-layer tunnel mechanism, rather than the single layer used in conventional overlays. It is this two-layer scheme which supports stacked overlays, as well as permitting use of unmodified applications and network services inside a deployed overlay [12]. It also permits network resources (hosts, routers) to participate multiple times in a single overlay, and is the only known overlay system that integrates both IPsec support and dynamic routing [13].

3. DynaBone

The DynaBone extends the X-Bone architecture to deploy a layered set of inner overlays (innerlays) together with a feedback and distribution proactive/reactive multiplexer (PRM), encompassed within an outer overlay (outerlay). The result is a composition of overlays that endures DDOS attacks, because any attacked individual network can be disconnected without substantially affecting the overall connectivity of the group (Figure 4). When the innerlays of a DynaBone are attacked, its PRMs shift traffic to the the remaining unaffected overlays (Figure 5). All of these are handled without the knowledge of the users or the applications running on top of the overlays.

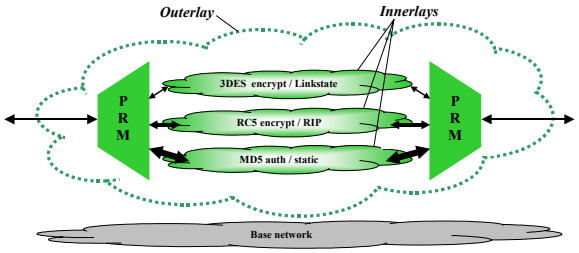

Figure 4 DynaBone architecture, PRM weights towards simpler innerlays

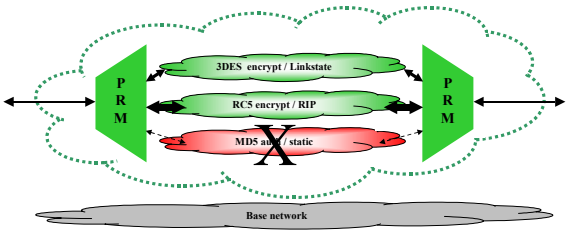

Figure 5 Reacting to DDOS attack, PRM shifts traffic away from affected innerlay

There are two components of the DynaBone architecture: its use of layered overlays, and its feedback-based multiplexer (PRM). The layered overlays are based on a unique capability of the X-Bone system that provides a recursive overlay structure. The PRM dynamically redirects outerlay

traffic to the innerlays best able to provide network service.

3.1. The Proactive-Reactive Multiplexer

The PRM consists of a multiplexer, demultiplexer, monitor, and proactive/reactive control components (Figure 6). The multiplexer distributes outgoing traffic among the innerlays on a per-packet, per-session, or per-connection basis depending on the multiplexing algorithms, or it may include FEC-style replication of packets across multiple innerlays at once [1]. It adds labeling information if needed, e.g., to reestablish order at the receiver or to coordinate FEC extraction. The demultiplexer gathers incoming packets, and may reestablish order, drop duplicates, or extract data from the FEC encoding.

The multiplexing algorithms are based on a combination of policies (built-in or user-specified), tunnel management and they interface to existing bandwidth reservation and allocation mechanisms. The policies and tunnel management determine which tunnel is used.

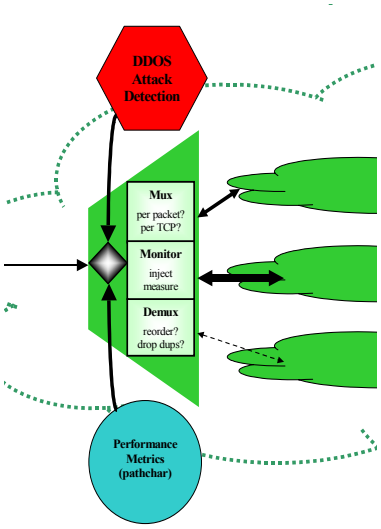

Figure 6 Detail of the PRM

Bandwidth allocation is provided by a combination of external interfaces to traffic shapers (e.g., ALTQ) to limit bandwidth on particular interfaces, RSVP to reserve bandwidth on paths, and RSVP extensions to reserve bandwidths over tunnels (the latter, provided they are extended by ongoing efforts to handle layered tunnels) [4][11].

The monitor coordinates the analysis of the status of individual innerlays. It may emit heartbeat messages itself, or invoke external mechanisms, such as *pathchar*, (circle) to track the performance of innerlays.

All three components are coordinated by a proactive/reactive controller (dark diamond in the figure). This controller determines how to configure the multiplexer to distribute traffic, incorporating information on attacks from external detectors (hexagon) [5], as well as incorporating performance information from external systems (circle) and the monitor.

3.2. Variety of Security and Performance

The DynaBone's deployment of alternate parallel concurrent innerlays utilizes a variety of existing network protocols and security algorithms. Recent measurements on 700 MHz PCs running FreeBSD 4.2 indicate a substantial variation in the bandwidth and latency between conventional IP (far left), encryption (middle-left group), and authentication (middle-right group) (Figure 7 and Figure 8). For completeness, IP compression is also shown (far right). In addition to these objective performance metrics, there are subjective security strength metrics as well; stronger algorithms are shown to the right in each group.

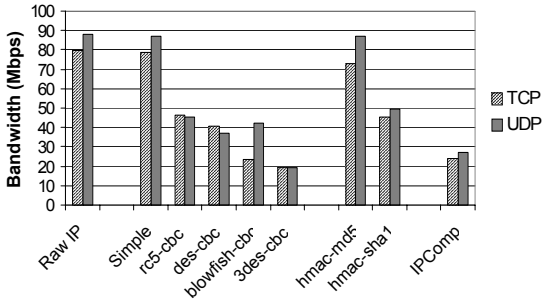

Figure 7 Bandwidth of IPsec algorithms

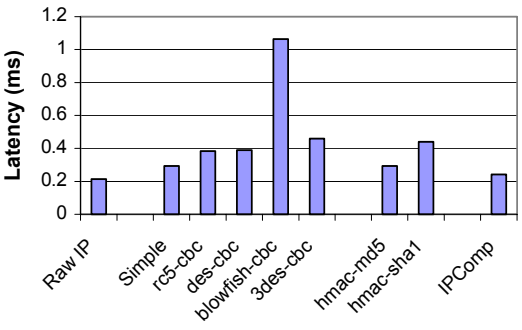

Figure 8 Latency of IPsec algorithms

This shows that there is substantial diversity in the deployed IPsec algorithms. There is also diversity in available software for network configuration and management. There are a number of IP multicast (e.g., dense-mode PIM, sparse-mode PIM, DVMRP) and interior routing protocols (e.g., RIP, RIPv2, linkstate). There are also a number of versions and implementations of various network services, such as DNS (ISC Bind 4.x, 8.x, 9.x, non-ISC versions, etc.). Recent CERT advisories have shown that some DDOS attacks are highly specific to particular tools or even versions of these tools. Maintaining parallel overlays with alternates provides the opportunity to disable compromised versions dynamically.

DDOS attacks come in many forms, focusing on protocols and OSs, particular implementations of protocols and OSs, specific hosts, subnets, or entire networks. Firewalls are effective at controlling attacks on services on well-known ports, based on the source IP address. However, as network services become more sophisticated (i.e., numerous, and using more dynamically-allocated ports), it becomes more challenging for a firewall to filter traffic on port information alone. DynaBone's parallel innerlays provide opportunities for more effective traffic control – both for entirely virtual deployments, and where multiple physical networks exist. In a virtual system, the parallel innerlays allow firewalls to filter on destination IP address alone, rather than requiring potentially complicated parsing to filter on TCP or UDP port numbers (e.g., when the IP packet has options). In networks with multiply-connected components, DynaBone helps automate the deployment of parallel overlays running on the separate infrastructure.

3.3. Current Status

The DynaBone has been implemented as an extension to the X-Bone overlay system. A number of preliminary technical issues have been addressed, and the current work is focusing on extending the implementation of the X-Bone system for native DynaBone deployment.

Current DynaBone deployments use the scripted application deployment capabilities of the X-Bone. A single X-Bone is deployed, and a script in the OM cyclically deploys innerlays. Finally, the application deployment script is used to tie the innerlays together and link their endpoint information into the PRMs. This is an intermediate method; a more appropriate solution would utilize the recursive overlay deployment. Although the X-Bone architecture was designed with recursion in mind, the implementation of native recursion, both as an extension to the overlay description language and as an implementation of the OM, is currently underway.

The system has successfully utilized over 50 concurrent innerlays for DDOS resistance. The current implementation does not differentiate these innerlays by security algorithm or performance; current work is focusing on incorporating that information into the PRM to direct the proportional use of the innerlays.

4. Summary

DynaBone provides a way to combine individual network DDOS defenses into a coherent whole that is stronger than its parts. In most DDOS systems, the use of COTS components increases the likelihood of successful attack; in DynaBone, COTS results in a larger variety of innerlay configurations, strengthening the resulting deployed outerlays. In fact, by their nature, even a single winning DDOS solution can be the target of likely attack; DynaBone reduces this liability, and encourages the combined deployment of diverse solutions. DynaBone also allows the use of weaker DDOS defenses; in combination, a sufficiently large set of such defenses itself provides a strong defense, because it would fail only when all component innerlays were compromised at once.

DynaBone extends the notion of air-gap defense and firewalls to isolate individual innerlays when attacked. Firewalls are limited air-gap defenses, where particular ports or address combinations are blocked from services at a site (host or router). DynaBone extends that capability to disconnect whole networks, including their networking protocols and services, when they are attacked. When its multiple innerlays are deployed virtually (over tunnels), they enable more efficient firewalling based solely on IP addresses (of the innerlay that is disconnected) – rather than requiring complicated transport packet parsing. When multiple innerlays can be configured over physically distinct interfaces and paths, they afford the same protection as multihoming, but with automated configuration and control.

DynaBone is a distributed service that protects against distributed attack. It enables continuous operation when attacks are successful. By disconnecting the affected innerlays, it has the added benefit of negatively affecting the attacker. E.g., when

DES is used for an encryption attack, all DES traffic is disabled (DES-based innerlays are shutdown). The result is that the attackers traffic has no network over which to travel, effectively pushing the attack back to the source. If the attacker uses a DES-based network, that network will suffer decreased service. However, at the same time, DynaBone overlays shunt traffic to non-DES innerlays, restoring service dynamically.

DynaBone supports graceful degradation, because an attack on each overlay compromises only a portion of the outerlay's traffic. Recovery is provided passively, in the use of a set of innerlays. Restoration is provided by detecting the DDOS attack, identifying it with particular innerlays, and shunting traffic away from those innerlays. DynaBone supports multiple kinds of reconstitution – where traffic migrates to unaffected innerlays, services can migrate, and defensive tactics (e.g., honeypots, traffic hiding) can be employed.

DynaBone provides these benefits to existing applications, without modifying or augmenting existing operating systems or network protocols. Finally, DynaBone extends X-Bone's coalition support, where these capabilities can be deployed across 'administrative domain' boundaries.

5. References

[1] Bestavros, A., Kim, G., "TCP Boston: A Fragmentation-tolerant TCP Protocol for ATM Networks," Proceedings of Infocom'97, Kobe, Japan, April 1997.

[2] Braden, R., ed. "Requirements for Internet Hosts -- Application and Support," RFC-1123, Oct. 1989.

[3] Cert, "Denial-of-Service Developments," CERT Advisory CA-2000-01, Jan. 3, 2000. http://www.cert.org/

[4] Cho, K., "Managing Traffic with ALTQ," In Proceedings of USENIX 1999 Annual Technical Conference: FREENIX Track, Monterey CA, June 1999.

[5] Deception Tool Kit, http://www.all.net/dtk/

[6] Early, S., Anderson, R., "The XenoService - A Distributed Defeat for Distributed Denial of Service," Proc. IEEE Info. Survivability Workshop 2000, October 2000.

[7] Eriksson, H., "MBone: The Multicast Backbone," Communications of the ACM, Aug. 1994, pp.54-60.

[8] Kent, S., Atkinson, R., "Security Architecture for the Internet Protocol," RFC-2401, Nov. 1998.

[9] Kung, H.T., Wang, S.Y., "TCP trunking: design, implementation and performance," Proc. ICNP 1999, Page(s): 222 –231.

[10] Scott, C., Wolfe, P., Erwin, M., *Virtual Private Networks*, O'Reilly & Assoc., Sebastapol, CA, 1998.

[11] Terzis, A., Krawczyk, J., Wroclawski, J., Zhang, L., "RSVP Operation Over IP Tunnels," RFC-2746, Jan. 2000.

[12] Touch, J., "Dynamic Internet Overlay Deployment and Management Using the X-Bone," Computer Networks Jul. 2001, pp. 117-135. Previously in Proc. ICNP 2000, pp. 59-68.

[13] Touch, J., Eggert, L., "Use of IPsec Transport Mode for Virtual Networks," (work in progress), Mar. 2000.

[14] Touch, J., Hotz, S., "The X-Bone," Proc. Global Internet Mini-Conference at Globecom, Nov. 1998.

[15] Touch, J., Wang, Y., Eggert, L., "Virtual Internets," ISI Technical Report ISI-TR-2002-558, July 2002.

[16] Traw, C., Smith, J., "Striping Within the Network Subsystem," IEEE Network, Vol. 9, No. 4, July/August 1995, pp. 22-29.