Countering DDoS Attacks with Multi-Path Overlay Networks

by Angelos Stavrou and Angelos Keromytis

Distributed Denial of Service (DDoS) has emerged as a major threat to the operation of online network services [1, 2, 3]. Current forms of DDoS attacks implicate multiple groups of Internet machines that have been taken over and controlled by an attacker. These machines, called *bots*, are manipulated by the attacker to produce an excessive surge of traffic toward a target server, the victim. The target server is forced to processing and/or to link-capacity starvation, since malicious traffic is blended with normal traffic, making it difficult to weed out. Figure 1 depicts a DDoS attack and its impact on the target server.

Unfortunately, DDoS attacks can only become worse: Despite network and processing speeds that increase with every passing day, real-world botnet sizes and attack capabilities increase at the same rate. Furthermore, attackers devise sophisticated software to infect and subsequently control thousands of infected machines while remaining stealthy. [4]

Addressing the network (DDoS) problem is extremely hard, given the fundamentally open nature of the Internet and the apparent reluctance of router vendors and network operators to deploy and operate new, potentially complex mechanisms. [5] Overlay-based approaches such as Secure Overlay Services (SOS) [6], funded by the Defense Advanced Research Projects Agency (DARPA) and the National Science Foundation (NSF Grant ITR CNS-04-26623); I3 [7]; and MayDay (Distributed Filtering for Internet Services) [8] offer an attractive alternative, as they do not require changes to the existing routing infrastructure. Furthermore, such systems require minimal or no collaboration from Internet Service Providers (ISPs), making their deployment completely transparent and thus practical. Overlay-based protection systems use an Internet-wide network of nodes that act as first-level firewalls, discriminating between legitimate traffic and potentially malicious traffic, enforcing some form of user or end-host authentication. Their distributed nature requires an extremely well-provisioned adversary to suppress their functionality, because, to disrupt protected communications, attack traffic must be split among all nodes. But how do these systems operate in practice?

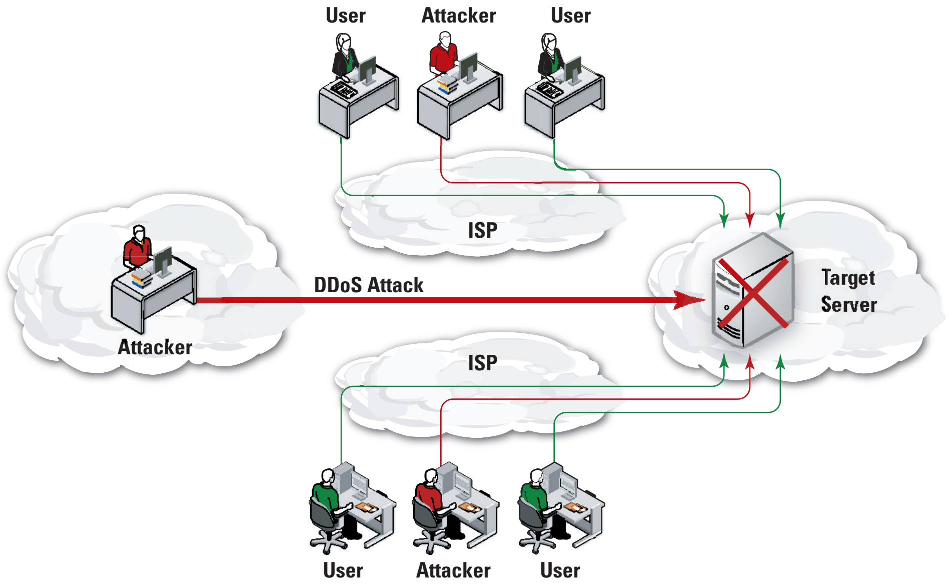

Figure 1 The target server is the victim of a DDoS attack. Legitimate users are denied access to the actual service since attackers generate overwhelming requests toward the target server's network.

Protection *via* Indirection Overlay Networks

In Figure 2, we present the main characteristics of the original SOS architecture, which is representative of indirection *via* overlay-based protection systems. We distinguish the three parts of the system:

the users, the overlay, and the protected server. Users want to establish a connection with the protected server but cannot do so directly—only a few select overlay nodes are allowed to communicate through the router-filtered area. These nodes can change over time in a random manner (but in coordination with the filtering routers). Although all overlay nodes are assumed to be publicly known, the precise identity of those nodes that can forward traffic through the filtered region at any given point in time is kept secret.

Users have to first authenticate themselves to the overlay network by connecting to a publicly advertised overlay node. This authentication can be either *via* cryptographic protocol and/or reverse Graphic Turing Tests (GTTs) [9] to determine valid users. (In some scenarios, this may simply mean "humans," while in other cases, some form of "proper" authentication may be required). Traffic from legitimate users is routed *via* the overlay and through the allowed overlay nodes to the protected service. However, malicious (or simply unknown) traffic is simply dropped by the overlay nodes, keeping the DDoS attack far from the protected service and potentially close to the attacker, using the overlay network as an indirection mechanism. One assumption made by systems such as SOS is that there is enough capacity leading to the filtering router to withstand a direct DDoS attack (*i.e.*, the unprotected links cannot be saturated). In most instances of DDoS attacks to date, the upstream ISP can handle the additional traffic; it is the target's uplink that is typically less well provisioned. By allowing only a few, select overlay nodes to forward traffic through this router, we avoid the need for new (potentially expensive, computationally or otherwise) router features.

Unfortunately, the original approaches of the Indirection-based Overlay Network (ION) depend on the inability of an adversary to discover connectivity information for a given client and the infrastructure (*e.g.*, which overlay node a client is using to route traffic). This makes them susceptible to a variety of easy-to-launch attacks that are not considered in the standard threat model of such systems. For example, adversaries may possess real-time knowledge of the specific overlay node(s) through which a client is routing traffic or may be attacking nodes using a time-based scheme that will try to maximize the impact of the attack on a client's connectivity. Such attacks can be network-oriented such as Transmission

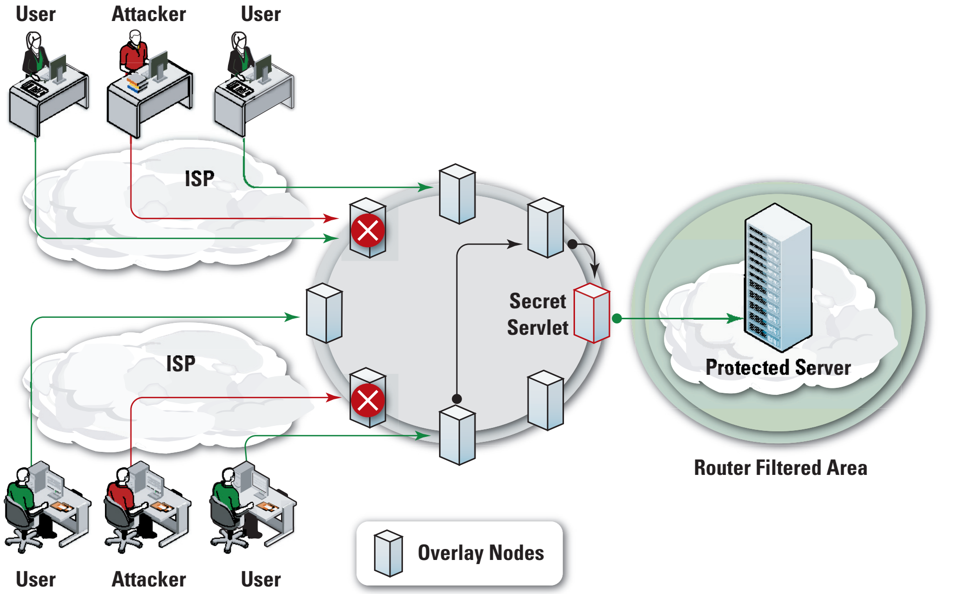

Figure 2 An overlay-based protection system. The users connect through the overlay nodes to the protected server. The overlay nodes act as distributed filters deep inside the network, mitigating the effects of a DDoS attack by dropping all unauthorized and unknown requests.

Control Protocol Synchronize (TCP SYN) attacks, application-related "sweeping" attacks, or "targeted" attacks.

In targeted attacks, an attacker who has knowledge of a client's communication parameters can "follow" the client's connections and bring down the nodes that he tries to connect to. As soon as the client realizes (typically, after some time-out period) that the overlay node is unresponsive and switches to a new node, the attacker also switches the attack to this new node. Thus an attacker that can bring down a single node can succeed in a targeted DDoS attack for specific clients. Similar attacks, exploiting information that must only be available to trusted components of the system but which an attacker can feasibly gain access to, are possible against almost all previously proposed anti-DDoS mechanisms.

Furthermore, IO networks are susceptible to an even worse type of attack: the sweeping attack. For this, an attacker uses its power (which is insufficient to bring down an entire ION) to target a small percentage of the overlay nodes at a time. The weak point of the overlay network is the application-level state maintained by the overlay node that is responsible for a client. Destroying this state forces the client to re-establish both network and application-level connectivity, degrading the clients' connection and leading to DDoS for time-critical or latency-dependent applications. Repeating this attack can force clients to re-establish their credentials multiple times within short periods of time, making IONs completely impractical. Thus, although IONs can counter blind DoS attacks, they remain vulnerable to a range of simple but debilitating attacks.

A Novel, Stateless Architecture

We believe that these inherent limitations of first-generation, overlay-based, traffic-redirection mechanisms can be addressed by adopting a spread-spectrum-like communication paradigm. Note that although we use the term "spread-spectrum" to describe our approach, our work is *not* geared toward wireless networks nor does it touch on physical-layer issues. Our approach, as shown in Figure 3, is straightforward: Spread the packets from the client across all overlay nodes in a random manner, storing no network- or application-level state in the overlay nodes. The path diversity naturally exhibited by a distributed overlay network serves as the "spectrum" over which communications are "spread." In our system, a token issued by the overlay network to the client is used to verify the authenticity of each packet communicated by the client. The use of a token (akin to a Kerberos ticket) alleviates the necessity to maintain application- or network-level state at any overlay node (unlike previous IONs) at the expense of bandwidth (since the ticket must be included in every packet routed through the ION). In return, our system is impervious to attacks that use this state dependence to attack the overlay.

An attacker will not know which nodes to direct an attack to; randomly attacking a subset of them will only cause a fraction of the client's traffic to be dropped. By using Forward Error Correction (FEC) or simply duplicating packets (*i.e.*, simultaneously sending the same packet through two or more different overlay nodes), we can guarantee packet delivery with high probability, if we place an upper bound on the number of nodes an attacker can simultaneously attack.

Attack Resilience and Performance

To evaluate our system, we used a testbed consisting of PlanetLab Consortium machines located at various sites in the continental US. These machines were running User Mode Linux (UML) on commodity *x*86 hardware (Intel and compatible computer processors) and were connected using the Abilene Network Internet-2 high-performance backbone. Using these fairly distributed machines, we constructed our overlay network of overlay nodes by running a small forwarding daemon on each of the participating machines. We also used two more machines, acting as client and server, respectively. In our experiments, we measured link characteristics such as end-to-end latency and throughput when we interposed the overlay network of overlay nodes between the client and the protected server. To measure throughput, we used a protected server that was located at Columbia University in the City of New York. For our latency measurements, we used *http://www.cnn.com* as the "target." In both cases, the goal of the

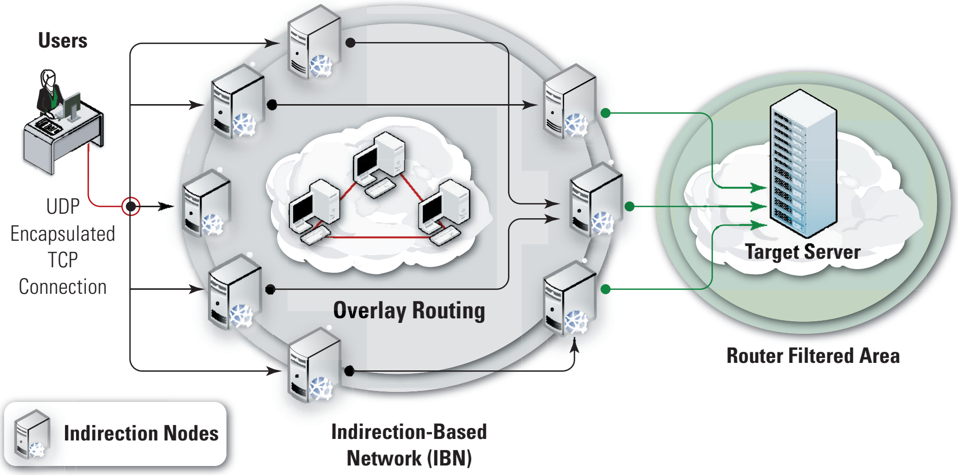

Figure 3 Users spread their packets to the network using a pseudo-random generator to avoid creating state to a single indirection node. An attacker cannot succeed by focusing his attack to some of the indirection nodes. Our system can sustain attacks that bring down up to 40% of indirection nodes, making it suitable for applications that require high levels of resiliency.

client was to establish communication with the protected server. To do so, the client used User Datagram Protocol (UDP) encapsulation on the TCP packets generated by a Secure Copy (SCP) session and then spread the UDP packets to the nodes participating on the overlay network. Those packets were in turn forwarded to a pre-specified overlay node that was permitted to connect to the protected server. Since our throughput connection measurements involve a client and a server that were co-located, we effectively measured the worst-case scenario (since our otherwise-local traffic had to take a tour of the Internet). A non-co-located server would result in a higher latency and lower throughput for a direct client-server connection, leading to comparatively better results when we use the overlay. Surprisingly, in some cases, we can achieve better latency using the overlay rather than by connecting directly to the server.

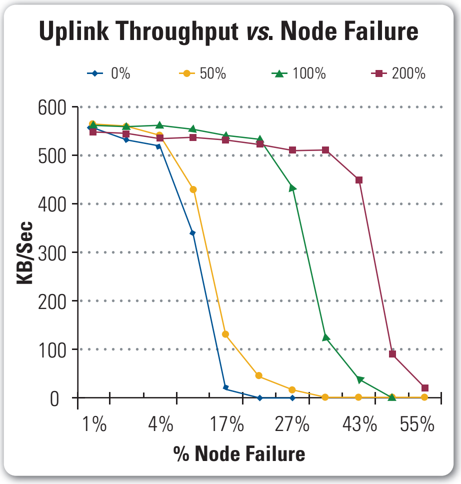

Figure 4 Throughput results in KB/s when we use the uplink of our client under attack. The attack happens on a random fraction of the overlay nodes. Each line represents different packet replication levels: For 100% packet replication, the client sends twice the amount of traffic by replicating each packet. Allowing packet replication helps us achieve higher network resilience.

Through our experiments and theoretical analysis, we show that, for an attacker to successfully attack our system, he will have to subvert or suppress more than 40% of the overlay nodes before the system becomes unusable for all users. Of course, our ability to thwart attacks depends on the packet replication (redundancy) we use. For example, a packet replication of 100% means that the client will replicate all packets once, effectively sending twice the amount of traffic. Figure 4 presents the system uplink performance when we vary both the number of overlay nodes that are under attack and the packet replication factor. For 200% packet replication we can sustain attacks up to 40% of the overlay nodes. Thus, our system has an operational threshold on the order of 40% of the nodes being subverted. Before this 40% threshold is reached, the users will not notice a significant impact to their connectivity. As a comparison, in the original SOS architecture, the user had to find an overlay node that was not under attack, which becomes increasingly difficult as we increase the portion of nodes under attack. We quantify the increase in the system's resistance to attacks using a simple analytical model and provide experimental validation by deploying a prototype over PlanetLab, a wide-area overlay network testbed. PlanetLab nodes are distributed across the Internet, serving as an ideal platform for experimentation.

Our analysis shows that an Akamai-sized ION with 2,500 nodes can withstand attacks that bring down up to 40% of the overlay. This corresponds to attacks that involve several million bots (attacking hosts), which is an order of magnitude larger than the biggest bot network seen to date. One expects that using an ION will impose a performance penalty. In our case, end-to-end latency increases by a factor of 2.0 in the worst case, but, by using packet replication, we maintain latency at the same level as that of the direct-connection case. These results confirm the findings from other research on multi-path routing.

Finally, we evaluated the overhead of our system to the end-to-end latency experienced by the clients. Although latency increase is a big concern whenever we add a network indirection system, our experiments show that, in the worst-case scenario, we have a 2.5 times increase in latency when compared to the direct connection to the protected server. However, this increase drops to just 1.5 times when we introduce a small packet replication of 50%. (For each two packets, we transmit another one.) In Figure 5, we present our latency results: As we increase the replication factor and for larger networks, we get better average latency results. In some cases, the latency observed when the client connects directly to the server can be higher than the one measured through the overlay. [The To (Overlay)/Td (Direct Connection) ratio in Figure 5 is below 1.0] This is true when some overlay nodes happen to have a lower latency route to the protected server when compared to the direct client-to-server route.

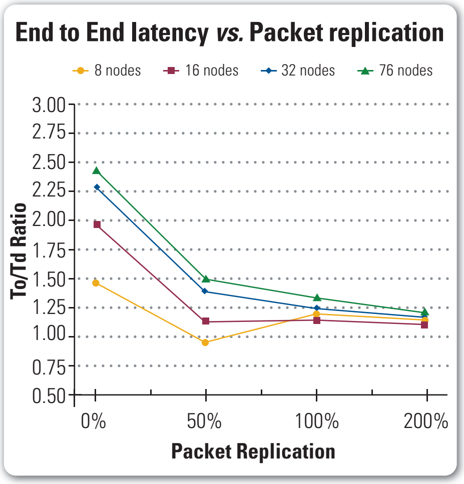

Figure 5 End-to-end average latency results for the index page and a collection of pages for *http://www.cnn.com*. The different points denote the change in the end-to-end latency through the Overlay, To, when compared to the Direct Connection, Td. Different lines represent different-sized overlays. Increasing the replication factor and for larger networks, we get lower average latency results because of the multi-path effect on the transmitted packets.

Conclusion

Our approach offers an attractive solution against congestion-based DDoS attacks in most environments, as it does not require modifications to clients, servers, protocols, or routers, both in terms of hardware and in existing software. Our plans for future work include developing a better characterization of the trade-offs that we have explored so far by introducing a coding scheme for the data transmission that will adapt to the network characteristics of each path used. Furthermore, we are looking into mechanisms to protect our system against attackers that can take over overlay nodes, thereby subverting part of the infrastructure. Finally, we are interested in deploying and using such a protection system on a larger scale than our experimental testbed to acquire operational experience in a real environment. Our article, *Countering DoS Attacks With Stateless Multipath Overlays,* [10] contains additional details about our system and the analysis and experimental evaluation.■

References

1. Hulme, G. (2004, September 13). Extortion online. *Information Week.*
2. Worldwide ISP Security Report. (2005 September). Retrieved from *http://www.arbor.net/downloads/Arbor_Worldwide_ISP_Security_Report.pdf*
3. Lipson, Howard F. (2002 November) Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues. Retrieved from *http://www.cert.org/archive/pdf/02sr009.pdf*
4. Ianelli, Nicholas. & Hackworth, Aaron. (2005 December). Botnets as a Vehicle for Online Crime. Retrieved from *http://www.cert.org/archive/pdf/Botnets.pdf*
5. The Cambridge-MIT Institute. (2005 January) DoS-Resistant Internet Working Group Meetings. Retrieved from *http://www.communicationsresearch.net/dos-resistant*
6. A. D. Keromytis, A. D. Misra, & Rubenstein, D. (2004 January). SOS: An Architecture For Mitigating DDoS Attacks. *IEEE Journal on Selected Areas of Communications (JSAC).*
7. I. Stoica, I., Adkins, D., Zhuang, S. & Surana, S. (2002 August) Internet Indirection Infrastructure. *Proceedings of ACM SIGCOMM.*
8. Anderson, David G. (2003 March). Mayday: Distributed Filtering for Internet Services. *Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems (USITS).*
9. Luis von Ahn, Luis, Blum Manuel, Hopper, Nicholas J. & Langford, John. (2003 May) CAPTCHA: Using Hard AI Problems For Security. *Proceedings of EUROCRYPT.*
10. Stavrou, A. & Keromytis, A. D. (2005). Countering DoS Attacks With Stateless Multipath Overlays. *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS)*, pp. 249–259, Alexandria, VA.

About the Authors

Angelos Stavrou | is currently a Research Assistant at the Network Security Laboratory at Columbia University. His research interests are security using Peer-to-Peer (P2P) and overlay networks. He received a BS in Physics with honors from the University of Patras, Greece, and an MSc in the theory of Algorithms, Logic, and Computation from the University of Athens, Greece. He also holds an MSc in Electrical Engineering from Columbia University and is currently a PhD candidate at Columbia.

Angelos Keromytis | is an Associate Professor of Computer Science at Columbia University. He received Masters and PhD degrees from the University of Pennsylvania and a Bachelors degree (all in Computer Science) from the University of Crete, Greece. His research interests include network and system survivability, authorization and access control, and large-scale systems security. His complete *curriculum vitae* may be found at *http://www.cs.columbia.edu/angelos/cv.html*

Letter to the Editor

Q ***I recently attended the Information Assurance Technical Framework Forum (IATFF) at Johns Hopkins Applied Physics Laboratory in Laurel, MD. While there, I heard a briefing on the protection of data at rest and noted something: the Secure Mobile Environment-Portable Electronic Device. This is the first I've heard of this device. Might you know something more about it?***

A The Secure Mobile Environment-Portable Electronic Device (SME-PED) is the National Security Agency's (NSA) concept for a secure, wireless, handheld product. Currently in development, the SME-PED will be a secure Personal Digital Assistant (PDA) and wireless phone. It will provide users with protected voice and data communications and support security levels up to the Top Secret level and email exchanges up to the Secret level.

The SME-PED will not only permit secure phone usage but will also be the first product to provide remote, wireless access to the Secret IP Router Network (SIPRNet). With NSA's Type 1 and Non-Type 1 encryption implemented, individuals will be able to access the Internet, NIPRNet, and SIPRNet *via* the SME-PED.

Only two companies were awarded the $36M contract to develop this product, with a scheduled delivery date of 2Q 2007. Although the SME-PED's release is scheduled almost a year from now, several government organizations have seen the value of this product and are already integrating the SME-PED in future plans and programs. For more information, please do not hesitate to contact us at iatac@dtic.mil. ■