

A DDoS Defense Mechanism with Topology Reconfiguration

Yu Cai, Assistant Professor, Michigan Technological University, cai@mtu.edu

Abstract

In this paper, we present design and implementation of a Secure Collective Defense (SCOLD) system against DDoS attacks. The key idea of SCOLD is to follow intrusion tolerance and topology reconfiguration paradigm, and provide alternate routes via a set of proxy servers and alternate gateways when normal route is unavailable or unstable due to network failure, congestion, or DDoS attack. The BIND9 DNS server and its DNS update utilities were enhanced to support new DNS entries with indirect routing information. The indirect route was implemented by utilizing IP Tunnel. Protocol software was developed on Linux systems. Experimental results showed that SCOLD can improve network security, availability and performance.

1. Introduction

Distributed Denial of Service (DDoS) attacks exploit a number of comprised machines and launch large coordinated packet floods towards a target, thereby causing denial of service for legitimate users. DDoS attacks have been an immense threat to the Internet for years, e.g., MyDoom [4] that knocked SCO Group website offline in 2004.

The increasing frequency and severity of network attacks reveal some fundamental security problems of today's Internet. The Internet was designed to provide fast, simple and reliable communication mechanisms, and its tremendous success is a credit to the original design. However, many network services (e.g., Domain Name Server (DNS)) and protocols (e.g., TCP/IP) were not designed with security as one of the basic considerations. Also, the highly distributed and interdependent nature of Internet provides opportunities and resources for coordinated and simultaneous attacks by malicious participants. Due to the same nature of the Internet, it is difficult to enforce common security policies, measurements and coordination among participants of the Internet. Therefore, the existing Internet architecture needs to be strengthened and services / protocols need to be enhanced or re-designed with security in focus.

In this paper, we present a DDoS defense mechanism named Secure COLlective Defense (SCOLD) system [18]. The key idea of SCOLD is to follow intrusion tolerance and network reconfiguration paradigm by providing clients with alternate routes via a set of proxy servers and alternate gateways when the normal route is unavailable or unstable due to network failure, congestion, or DDoS attack. The main techniques utilized in SCOLD are the enhanced Secure DNS Update and Indirect Route.

In SCOLD, the enhanced DNS system is utilized to store and convey indirect routing information, like proxy server IP addresses. There are two steps to enable indirect route in SCOLD. First, client DNS server needs to get indirect routing information from target DNS server. This is accomplished by enhanced secure DNS update. Second, after clients get indirect routing information from client DNS servers, clients can set up indirect route to target server. The communication channels between clients and target are kept open by using indirect routes.

SCOLD protocol software was developed for Linux systems. Experimental and simulation results showed that SCOLD can significantly improve network security, availability and performance, with acceptable overhead and satisfactory scalability.

The balance of this paper is organized as follows. In Section 2, we give an overview of SCOLD system. In Section 3, we present enhanced secure DNS update. In Section 4 indirect routing using IP Tunnel is presented. In Section 5 we present experimental results and simulation results. Related work is surveyed in Section 6 while the conclusion is drawn in Section 7.

2. System Overview

2.1 Motivation

Most organizations today deploy multiple gateways or multi-homing scheme as a backup measure in case of network congestion or failure. When main gateway is congested or unavailable due to DDoS attacks, legitimate traffic should be redirected to alternate gateways. However, once alternate gateways are exposed to public domain, they are subject to DDoS attacks. If the amount of attack traffic is large enough, alternate gateways will soon get congested too. Therefore, simply adding more alternate gateways is not sufficient to defend DDoS attacks.

Most existing DDoS defense mechanisms presume a scenario where packets are transmitted along a normal Internet route and via main gateway. Under very large-scale DDoS attack, malicious traffic at main gateway will consume most available network bandwidth. Some techniques that are performed behind main gateway (e.g., rate-limiting [11] and filtering [12]), will become much less useful. Other techniques (e.g., traceback [10], [9]) may suffer from significant performance degradation.

The SCOLD system defends against DDoS attacks by setting up indirect routes between clients and target server. The traffic between clients and target server is transported over the Internet through indirect routes [22, 23]. In SCOLD, three main issues need to be considered and solved.

- a) How to redirect heterogeneous clients' traffic through indirect route;
- b) How to utilize alternate gateways while protecting them from attacks;
- c) How to prevent attack traffic from using indirect route.

We solved the first problem by setting up indirect route via a collection of geographically separated proxy servers and alternate gateways. We solve the second and third problem by using proxy servers that are equipped with IDS, firewall and rate-limiting mechanism.

2.2 System architecture

Figures 1-3 illustrates how SCOLD system works. Figure 1 shows a target site under DDoS attacks where R is the main gateway, and R1-R3 are alternate gateways. For illustration purpose, we assume that the majority of traffic from net-a.com is malicious, that of net-b.com is legitimate, and that of net-c.com is mixed.

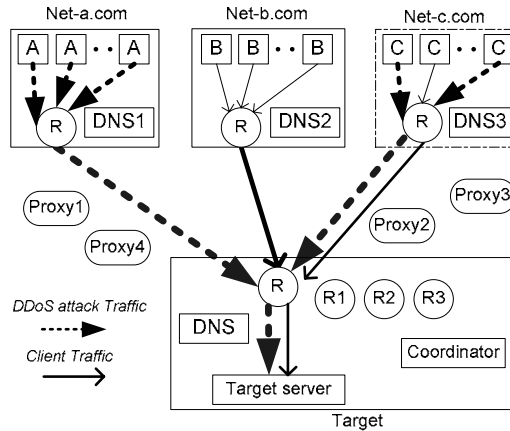


Figure 1: Target site under DDoS attack

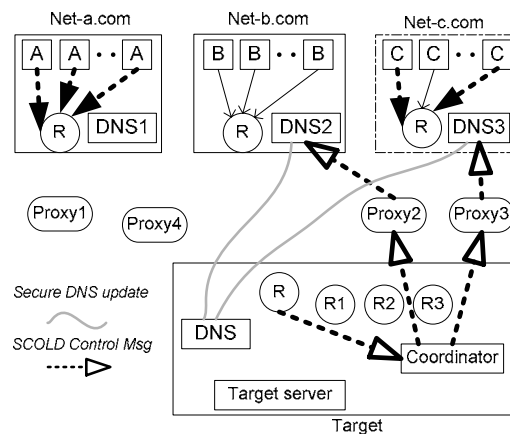


Figure 2: The control flow in SCOLD

Figure 2 shows control flow of SCOLD system. When target site is under DDoS attacks, its Intrusion Detection System (IDS) raises an intrusion alert and notifies SCOLD coordinator, who sits in the same or trusted domain of target server. The coordinator notifies some selected proxy servers (proxy 2 and 3 here) to set up indirect routes. The proxy servers notify the DNS servers of legitimate clients to perform a secure DNS update. The clients from net-b.com and net-c.com are notified with indirect route, but net-a.com is not notified due to its malicious traffic pattern.

Figure 3 shows how an indirect route is setup in SCOLD system. After a secure DNS update, client side DNS server gets new DNS entries containing designated proxy servers IP addresses. The clients query DNS server to get proxy server IP addresses. They can then set up indirect routes to target server via selected proxy servers. The proxy servers examine incoming traffic and relay to designate alternate gateway on target site.

On client side, the name resolve library needs to be enhanced to support indirect routing. In enterprise environment, internal clients go outside through an enterprise gateway (or an enterprise proxy server). Instead of modifying client resolvers, the enterprise gateway (or the enterprise proxy server) needs to be enhanced to support indirect route.

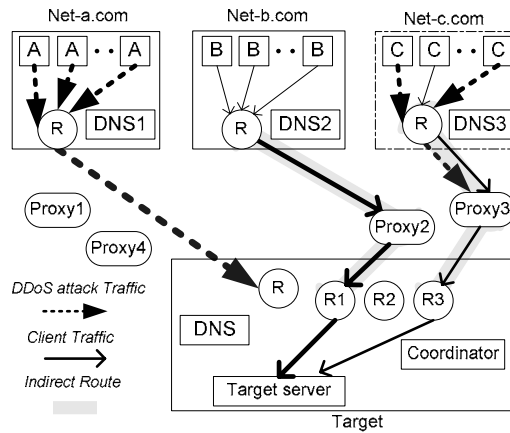


Figure 3: Indirect route in SCOLD

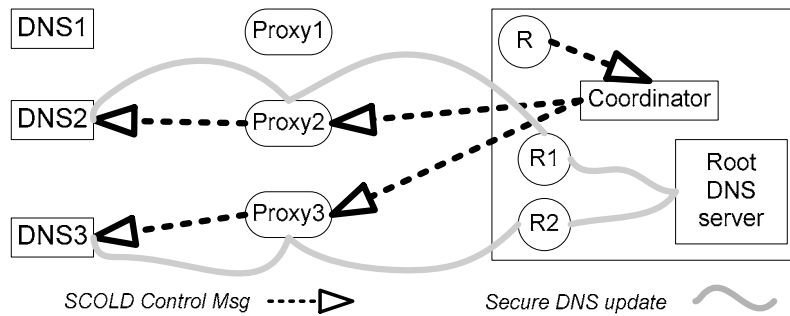


Figure 4: Protect the root DNS server

In SCOLD, the IP addresses of alternate gateways and SCOLD coordinator(s) are revealed only to trustworthy proxy servers to protect them from being attacked by malicious clients. Clients in public domain can connect to target side through designed proxy servers. To avoid traffic analysis at proxy servers by intruders, multiple proxy servers can be deployed in a chain on an indirect route. However, detecting and handling comprised proxy servers is not an easy task, and is beyond the scope of this paper.

Proxy servers in SCOLD are enhanced with IDS and firewall filters to block malicious traffic that may try to come in through indirect route. The detection of intrusion on proxy servers can provide additional information for identifying and isolating spoofed attack sources. In Figure 3, by combing the intrusion detection results from the main gateway R and the proxy server 3, attack source from net-c.com could be more accurately identified.

A proxy server itself may suffer from DDoS attacks or get congested when large volume of traffic comes through it. If there are a large collection of proxy servers available, the impact of heavy traffic can be alleviated by spreading traffic over multiple proxy servers [19, 20, 21].

The procedure for resuming normal route is similar to setting up indirect route. The proxy servers need to notify client DNS servers with another secure DNS update to restore normal DNS records. Clients query DNS server and start to resume normal direct route. We can also set an “expiration time” for indirect route so that SCOLD can automatically revoke obsolete indirect routes.

All control messages communicated in SCOLD system are encrypted using Secure Sockets Layer (SSL) and all nodes involved must be mutually authenticated. Experiments showed that this is one of the major causes of overhead in SCOLD system.

Proxy servers can be provided by participating organizations in SCOLD, or fee-based service providers, like Akamai.

Note that different proxy server selection may result in different system performance; and multiple proxy servers can be selected to enable parallel transmission or multi-path connection. This will be the future work of SCOLD.

2.3 SCOLD applications

Enhanced SCOLD proxy servers with bandwidth throttling can be used to defend large-scale DDoS attacks like MyDoom. SCOLD coordinator collects and analyzes target server system load, available network bandwidth and the statistics of client traffic. Based on the information, the coordinator can decide what is allowed maximum bandwidth on each proxy server. Proxy servers equipped with admission control and rate-limiting mechanism can enforce such bandwidth throttling. In Figure 3, the coordinator may assign different maximum bandwidths to proxy 2 and 3 depending on sever load and client behavior. This type of integrated IDS can help to control aggressive or malicious clients and reserve resources for normal operation.

A slightly revised version of SCOLD can be used to protect Root DNS servers from DDoS attacks, like an attack causing a brief service disruption on the nine of the thirteen DNS root servers in 2002 [3]. In Figure 4, DNS 1-3 are client side DNS servers, and the main gateway R of the root DNS server is under sever DDoS attacks. DNS 1-3 may experience significant delay or even failure when querying the root DNS server. Due to the current DNS querying model, end users will perceive a poor Internet performance with unbearable delay.

By utilizing SCOLD technique, we can set up indirect routes between client DNS and root DNS to ensure the normal operation of root DNS server. The IDS on the root DNS server raises alert and notifies coordinator; the coordinator notifies selected proxy servers (proxy 2, 3 here); the proxy servers notify legitimate client DNS servers with their IP addresses; those DNS servers then set up indirect routes to the root DNS via proxy servers and alternate gateways; then client DNS servers can query the root DNS server via indirect route.

In SCOLD architecture, proxy servers become the “frontline” fighting against DDoS attacks. It brings several benefits. First, with a large number of proxy servers available, target server gain more resources to defend DDoS attacks. Second, if a proxy server fails, we can quickly recruit other proxy servers without significant lost. Third, proxy servers with integrated IDS can provide powerful functionalities.

In SCOLD, there are three defense lines against DDoS attacks. First, based on the preliminary intrusion detection result from main gateway, some malicious clients will not be notified with indirect route. Second, proxy servers are equipped with IDS and firewall filters to further block malicious traffic. Third, proxy servers are equipped with admission control and rate-limiting mechanism to enforce bandwidth throttling and control the aggressive clients.

3. Secure DNS Update

In SCOLD, DNS is utilized to store and convey indirect routing information, which are proxy server IP addresses. This requires several modifications and enhancements on current DNS systems.

First, we need to redefine DNS record format for storing additional information. A sample of new DNS record in DNS zone file may look like below.

```
target.targetnet.com. 10 IN A      133.41.96.71
target.targetnet.com. 10 IN ALT   203.55.57.102
                        10 IN ALT   203.55.57.103
                        10 IN ALT   185.11.16.49
```

The first line is a normal DNS entry, containing host name and its IP address. The next 3 lines contain the IP addresses of proxy servers, as the newly defined "ALT" type (type 99).

DNS zone data needs to be securely updated from target side DNS server to client side DNS server upon request. However, in the scenario of DDoS attack, main gateway of target server domain may become unavailable or unstable. Therefore, DNS update might experience significant delay or even failure. By setting up indirect route and perform DNS update via indirect route, we can overcome the problem.

Figure 5 illustrates how the enhanced DNS update works. Step 1, the target side IDS raises intrusion alert, and notifies the coordinator. Step 2, the coordinator notifies the selected proxy server(s). Step 3, the proxy server notifies the client DNS server for a secure DNS update. Step 4, if the client DNS server decide to make a DNS update, it sends a request back to the proxy server for setting up indirect route; if the proxy server grants the permission, it notifies a selected alternate gateway and the target server for setting up indirect route; then an indirect route from the target DNS server to the client DNS server via the proxy server and the alternate gateway is set up. Step 5, client DNS server performs secure DNS update and gets DNS records from target DNS server.

4. Indirect Route

We investigate several alternatives for implementing indirect route, including SOCKS proxy, Zebedee, IPSec and IP tunnel. The main drawbacks of SOCKS are that it doesn't support UDP and FTP. Zebedee is an application to establish an encrypted and compressed tunnel between two systems. But it requires specific configuration per network application.

IP tunnel is a technique to encapsulate IP datagram within IP datagram. This allows datagram destined from one IP address to be wrapped and redirected to another IP address. IP tunnel provides what we want for indirect route.

IPSec is an extension to the IP protocol which provides security to the IP and the upper-layer protocols. Whether client traffic needs to be encrypted is a client decision. We decided to use IP tunnel to support basic indirect routing. However, the implementation using IP tunnel can be migrated to using IPSec easily. IP tunnel and IPSec have been used widely in Virtual Private Network (VPN) to set up "tunnel" between network nodes and redirect traffic.

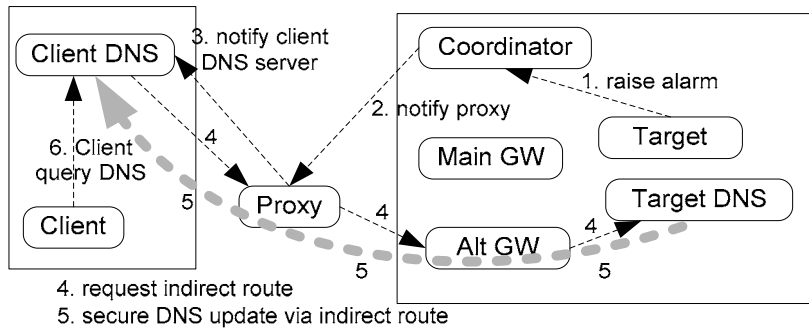


Figure 5: Secure DNS update via indirect route

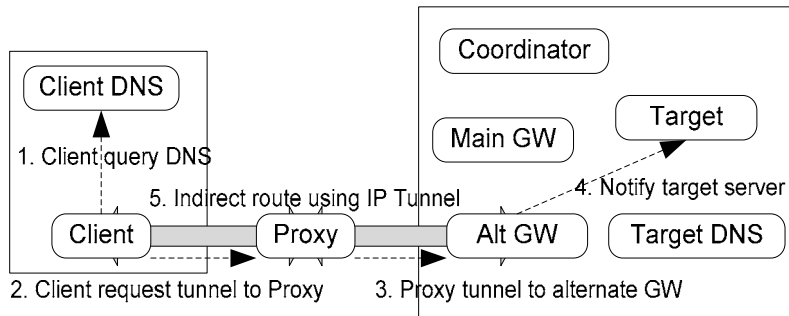


Figure 6: Indirect route by using IP tunnel

The advantages of using IP tunnel are as follows. IP tunnel is a layer three protocol. All the upper layer protocols and applications can utilize it. Second, IP tunnel is a widely used protocol and supported by most modern operating systems. Last but not the least, IP Tunnel consumes limited system resources since it is a device descriptor.

There is overhead associated with IP Tunnel due to the extra set of IP header and the reduced payload size. This can also cause fragmentation and reassembly overhead. In our experiments, the overhead in term of response time varied between 30% and 200%. But compared with the impact of DDoS attack, which may cause unbearable delay, the overhead of IP tunnel is still in an acceptable range. Fragmentation overhead can be avoided by restricting the message transfer size at sender.

Figure 6 illustrates how indirect route is set up by using IP tunnel. A client queries its DNS and get the IP addresses of proxy servers; the client sends a request to a proxy server for indirect route; if the proxy server grants permission, it notifies the designated alternate gateway; the alternate gateway notifies the target server, then an indirect route can be set up between the client and the target server via the proxy server and the alternate gateway. We set a timeout value at client side in case the communication is lost or the indirect route is broken.

5. Experimental & Simulation Result

In this section, we present experimental and simulation results on SCOLD.

5.1 Prototype implementation

Our implementation on BIND 9 and Redhat Linux 8 / 9 is summarized as follows.

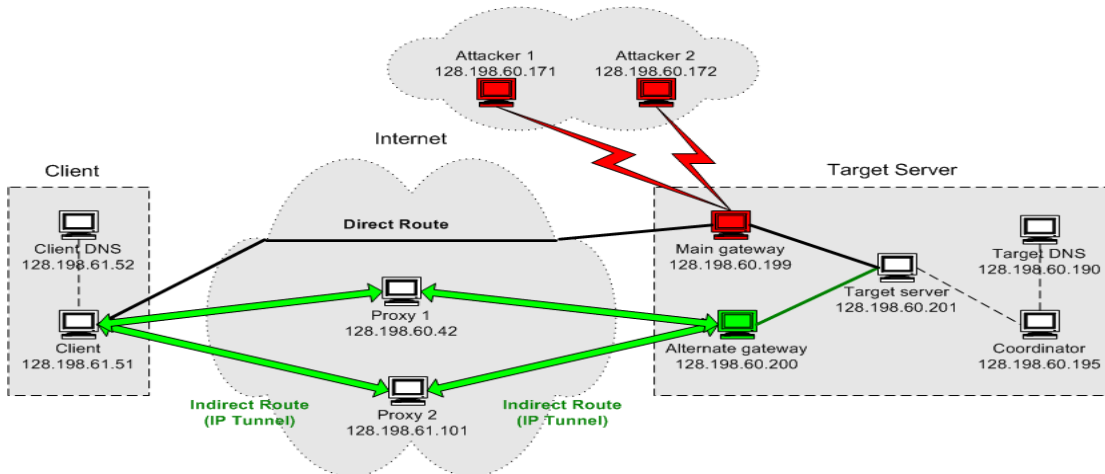


Figure 7: a SCOLD testbed

- 1) The BIND9 (v.9.2.2) DNS server was modified to support the newly defined ALT type 99 data and to enable the automated secure DNS update.
- 2) The DNS dynamic update utility (nsupdate) was enhanced to support indirect routing and the new data type. The enhanced DNS update utility was named nsreroute.
- 3) On client side, the domain name resolve library (v.2.3.2) was enhanced to support the new data type and enable the establishment of indirect route. In Redhat Linux, the resolve library is usually located in /usr/lib or /lib directory, and named as libresolv-*nnn*.so (*nnn* is the version). The routing table on the client node was modified at run time.
- 4) An agent program run on the participating nodes (client DNS server, target DNS server, proxy server, alternate gateway and target server) listening for the control message. The routing table on the participating node was modified at run time.
- 5) The indirect route was implemented by using IP Tunnel. By modifying the routing table at run time, we can utilize IP tunnel just like normal Ethernet devices. We also tested indirect route on Windows 2000 server using IP tunnel
- 6) All the control messages were encrypted using Secure Sockets Layer (SSL) and all participating nodes must be mutually authenticated. The implementation of authentication and encryption/decryption mechanism was a difficult decision, especially in large-scale distributed system. However, this was not the key focus of the paper. We utilize the most commonly-used public key cryptography and digital certificate in OpenSSL (v.0.9.6).

5.2 Experimental setup

We set up several testbeds consists of more than 20 nodes with various machine settings, including HP Vectra machines (PIII 500MHz, 256MB RAM, 100Mb Ethernet connection), HP Kayak machines (PII 233MHz, 96MB RAM, 10/100 Mb Ethernet connection), Dell machines (PIII 1GHz, 528MB RAM, 100 Ethernet connection) and virtual machines (96MB RAM, 100 Mb virtual Ethernet connection, running on a Dell machine with dual PIII 1.2GHz and 4G RAM). The operating systems are Linux Redhat 8, 9 and

Windows 2000 server. StacheldrahtV4 is used as DDoS attack tool. Figure 7 shows the network topology of a SCOLD test bed.

5.3 Analysis of experimental results

a) SCOLD initial setup overhead.

We first evaluated time taken to initially set up an indirect route in SCOLD, which is the SCOLD initial setup overhead. As discussed previously, there are three steps involved. Step 1, "IDS -> coordinator -> proxy". The overhead comes from the secure communication among nodes. Step 2, "Proxy -> client DNS -> perform secure DNS update". The overhead comes from the secure communication and the secure DNS update. Step 3, "client -> client DNS -> set up indirect route". The overhead comes from the secure communication, the client side resolve library processing overhead and the time to set up indirect route.

Table 1 shows the initial setup time in SCOLD. It is observed that the overhead comes primarily from the secure DNS update and secure communication among nodes. Table 2 further shows that the secure DNS update time increases dramatically when the number of client DNS servers increase. This suggests that there is a limit on how many client DNS servers a proxy server can handle concurrently.

Table 1: SCOLD initial setup time (second)

Step 1	Step 2	Step 3	Total
2.1	4.7	2.7	9.5

Table 2: Secure DNS update time (second)

1 DNS	10 DNS	25 DNS	50 DNS
4.7	25	96	240

b) SCOLD performance

Next we evaluated the SCOLD performance. Table 3 shows the processing overhead of using indirect route vs. the possible delay of direct route under DDoS attacks. The SCOLD processing overhead comes from the IP tunneling overhead and more Internet hops involved in indirect route. We can observe that the overhead of indirect route in term of response time is about 70%. Further experiments showed the overhead varied from 30%–200%. However, under DDoS attack, the response time of using direct route increased dramatically (15 times to infinity), while the response time of using indirect route kept the same. Table 3 also shows that the SCOLD performance is relatively independent of the application type (Ping, HTTP, FTP).

Table 3: Indirect Route processing overhead vs. Direct Route delay under DDoS attack

Test	No attack		Under DDoS attack		Direct Route Delay (c) / (a)	Indirect Route Overhead (b - a) / (a)
	Direct Route (a)	Indirect Route (b)	Direct Route (c)	Indirect Route (d)		
Ping	49 ms	87 ms	1048 ms	87 ms	21 times	77%
HTTP(100k)	6.1s	11s	109s	11s	18 times	80%
HTTP(500k)	41s	71s	658s	71s	16 times	73%
HTTP(1M)	92 s	158s	timeout	158s	infinity	71%
FTP(100k)	4.2 s	7.5s	67s	7.5s	16 times	78%

FTP(500k)	23 s	39s	345s	39s	15 times	69%
FTP(1M)	52 s	88s	871s	88s	17 times	69%

We also evaluated the performance of the enhanced secure DNS update. Table 4 shows performance comparison between an enhanced DNS update with indirect route (using nsrroute) vs. normal secure DNS update with direct route (using nsupdate). It shows that the nsrroute with indirect route is usually slower than the nsupdate with direct route by 30 - 70%. The overhead is mainly caused by the time to set up indirect route and transport DNS data via indirect route. However, when the main gateway of the target site is under DDoS attack, the nsupdate with direct route is impacted seriously, and the nsrroute with indirect route is almost not affected.

Table 4: Performance of nsrroute vs. nsupdate, with and without DDoS attack

	No attack		Under DDoS attack		nsupdate Delay (c) / (a)	nsrroute Overhead (b-a) / (a)
	nsupdate (a)	nsrroute (b)	nsupdate (c)	nsrroute (d)		
1 DNS	4.2 s	7.1s	50 s	7.1 s	12 times	70%
10 DNS	21.1 s	27.4 s	timeout	27.4 s	infinity	30%

c) Other overheads

Table 5 shows the overhead of enhanced resolver. We measured the response time to resolve a domain name by using enhanced resolver and the original resolver. It is observed that the enhanced resolve library only imposes very limited overhead compared to original resolve library.

We evaluated the overhead of the enhanced BIND DNS server. Table 6 shows the response time to answer a domain name query by using the enhanced DNS server and the original DNS server. The result shows that the overhead of the enhanced DNS server is also very limited.

Table 7 shows the overhead of IP tunnels itself. It is observed that the number of IP tunnels on network nodes doesn't affect the performance, because IP tunnel itself consumes very limited system resources.

Table 5: Performance of enhanced resolver vs. original resolver

Test	Enhanced resolver	Original Resolver
Ping	0.7 ms	0.6 ms
HTTP	0.7 ms	0.7 ms
FTP	0.7 ms	0.7 ms

Table 6: Performance of enhanced DNS vs. original DNS

Test	Enhanced DNS	Original DNS
Ping	1.2 ms	1.1 ms
HTTP	1.2 ms	1.1 ms
FTP	1.2 ms	1.1 ms

Table 7: The influence of how many tunnels exist

Test	1 tunnel	10 tunnels	50 tunnels	100 tunnels
Ping	87 ms	87 ms	87 ms	87 ms

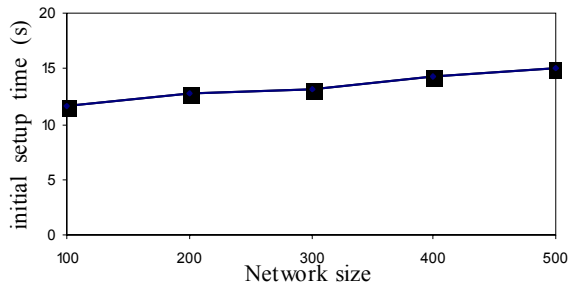


Figure 8a: average initial setup time vs. network size

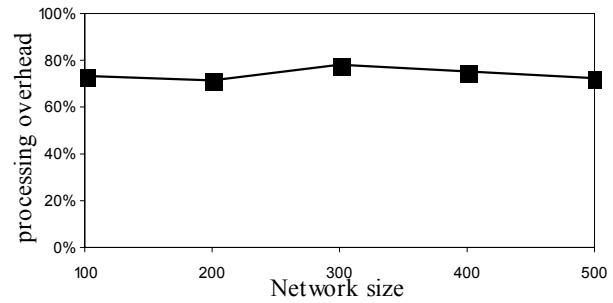


Figure 8b: indirect route processing overhead vs.

HTTP(100k)	11s	11s	11s	11s
------------	-----	-----	-----	-----

5.4 Preliminary simulation results

To further analyze the overhead in SCOLD, ns2 simulator was used to perform the simulation study for large-scale network. The topologies used in simulation are generated using GT-ITM (Georgia Tech Internetwork Topology Models). We created transit-stub graphs with 100-500 nodes. We picked nodes in the same stub for target server, target DNS server, coordinator, main gateway and 3 alternate gateways. We randomly picked 10% nodes as proxy servers, 5% nodes as DDoS attackers, 20% nodes as clients and 4% nodes as client DNS servers.

For simplicity, we set the overhead of IP tunneling and the overhead of secure communication to be a fixed percentage with a small random variance. We randomly generated background traffic whose average is 60% of the total network bandwidth. We generated DDoS attack traffic which can completely shutdown the victim.

Figure 8a shows that the average initial setup time of indirect route increases slowly when the network size increases. Figure 8b shows that the indirect route processing overhead keeps nearly constant when the network size increases. In both figures, SCOLD demonstrates good scalability with respect to the initial setup overhead and the processing overhead.

6. Related Works

6.1 DDoS defense mechanism

J. Mirkovic, et al. from UCLA presented a taxonomy of DDoS attacks and DDoS Defense Mechanisms [16]. SCOLD falls into the category of reconfiguration mechanism. Related works in reconfiguration mechanism include reconfigurable overlay networks ([17], [15]), resource replication services and attack isolation strategies ([13]).

The Resilient Overlay Network (RON) [17] is an architecture that allows distributed Internet applications to detect and recover from path outages and periods of degraded performance within several seconds. A RON is an application-layer overlay on top of the existing Internet routing substrate. The RON nodes monitor the functioning and quality of the Internet paths among themselves, and use this information to decide whether to route packets directly over the Internet or by way of other RON nodes. The XenoService [14] is a distributed network of web hosts that respond to an attack on any one web site by replicating it rapidly and widely. In [1], Christian Cachin, et al. from IBM presents an intrusion tolerance system named Secure INtrusion-Tolerant Replication Architecture1 (SINTRA). SINTRA supplies a number of group communication primitives, such as binary and multi-valued Byzantine agreement,

reliable and consistent broadcast, and an atomic broadcast channel. Atomic broadcast immediately provides secure state-machine replication. Stefan Savage, et al. [8] present a Detour project. The authors proposed to use intelligent routers spread at key access and interchange points to "tunnel" traffic through the Internet. These intelligent tunnels can improve performance and availability by aggregating traffic information, shaping bursty traffic flows, and using more efficient routes.

6.2 DNS enhancement

DNSSEC (DNS Security Extensions) is one of the major efforts to improve the DNS security. DNSSEC was designed to provide end-to-end authenticity and integrity in DNS. All zone data in DNSSEC is digitally signed with public-key cryptography. By checking the signature, a resolver can verify the validity of a DNS response.

Another major DNS enhancement is dynamic DNS update protocol, which allows an entity to update a DNS record "on the fly". Dynamic DNS update can create caching issues and additional problems. Dynamic DNS update was extended to secure DNS update by using a set of keys to authenticate an update. Digital signatures are stored in the DNS as SIG resource records and are used to encrypt and decrypt update messages for a zone.

DNS has also been extended for purposes other than name-to-address mapping and name resolution. Web server load balancing using DNS, storing IPsec key in DNS, and attribute-base naming system are some of the many examples. DNS for load balancing and traffic distribution among a cluster of web servers has been studied in [6, 7]. In [2], the author proposed a method for storing IPsec keying material in DNS. The IPSECKEY resource record is used to publish a public key that is to be associated with a domain name. It can be the public key of a host, network, or application. Intentional Naming System [5] is a resource discovery and service location system by mapping service name-attributes to name records using an intentional name language.

7. Conclusion

This paper presents the SCOLD architecture to defend against DDoS attacks. SCOLD redirects traffic between clients and servers through indirect routes via proxy servers and alternate gateways. BIND9 DNS package and its secure DNS update utility were enhanced to support indirect route. IP tunnel was utilized to implement indirect routing. The preliminary results show that SCOLD can improve network security, availability and performance. It is our hope that the research results of SCOLD can produce a valuable secure software package, and provide insights for network security and Internet cooperation.

References

1. Christian Cachin, Jonathan A. Poritz. "Secure Intrusion-tolerant Replication on the Internet", Proc. of Intl. Conference on Dependable Systems and Networks, 2002.
2. M. Richardson, "A method for storing IPsec keying material in DNS", RFC 4025.
3. Internetnews, "Massive DDoS Attack Hit DNS Root Servers", <http://www.internetnews.com/>
4. Internetweek, "SCO Moves Web Site To Battle MyDoom", <http://www.internetweek.com/>
5. W. Adjie-Winoto, E. Schwartz, H. Balakrishnan and J. Lilley, "The design and implementation of an intentional naming system", Operating Systems Review, vol.35, pp. 186-201, 1999.
6. V. Cardellini, "Redirection Algorithms for Load Sharing in Distributed Web-server Systems", Proc. of 19th IEEE Int. Conf. on Distributed Computing Systems, 1999
7. Eddie, Enhanced DNS Server, <http://eddie.sourceforge.net/lbdns.html>

8. S. Savage and T. Anderson, "Detour: a Case for Informed Internet Routing and Transport," IEEE Micro, pp. 50-59, v 19, no 1, 1999.
9. S. Savage, D. Wetherall, A. Karlin and T. Anderson, "Practical network support for IP Traceback", Proc. of 2000 ACM SIGCOMM Conference, 2000.
10. D. X. Song and A. Perrig, "Advanced and authenticated marking schemes for IP Traceback," Proc. of IEEE Infocom, 2001.
11. T. M. Gil and M. Poletto, "MULTOPS: a data-structure for bandwidth attack detection," Proc. of 10th Usenix Security Symposium, 2001.
12. Mazu Networks, "Dynamically Provisioned Monitoring, traffic master", technical report, http://www.mazunetworks.com/white_papers/provmon-toc.html
13. BBN Technologies, "Applications that participate in their own defense," <http://www.bbn.com/infosec/apod.html>
14. J. Yan, S. Early, and R. Anderson, "The XenoService – A distributed defeat for DDoS", Proc. of ISW 2000.
15. Information Sciences Institute, "Dynabone", <http://www.isi.edu/dynabone>
16. Jelena Mirkovic, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms", UCLA Technical Report
17. D. Andersen, H. Balakrishnan, F. Kaashoek and R. Morris, "Resilient Overlay Networks," In Proceedings of 18th ACM SOSP, October 2001.
18. Edward Chow, Yu Cai, David Wilkinson, and Ganesh Godavari, "Secure Collective Defense System", In Proceedings of GlobeCom 2004.
19. M. Zhang, et. al., "A Transport Layer Approach for Improving End-to-End Performance and Robustness Using Redundant Paths", In Proc. of the USENIX 2004 Annual Technical Conference. 2004.
20. H. Hsieh, et. al., "ptcp: An end-to-end transport layer protocol for striped connections", In Proceedings of IEEE ICNP, 2002.
21. H. Adiseshu, et. al., "A reliable and scalable striping protocol", In Proceedings of ACM SIGCOMM, 1996.
22. Y. Zhang, et. al., "On the characteristics and origins of Internet flow rates". In Proceedings of SIGCOMM, 2002.
23. S. Savage, et.al. "The End-to-End Effects of Internet Path Selection", Proc. ACM SIGCOMM, 1999