Proceedings of the 2004 IEEE
Workshop on Information Assurance
United States Military Academy, West Point, NY 10-11 June

Enhanced Secure Dynamic DNS Update with Indirect Route

David Wilkinson, C. Edward Chow and Yu Cai

Abstract – In this paper, we present the design and implementation of the enhanced secure dynamic DNS Update with indirect route (the IR DNS update). The existing DNS update may experience serious performance problem when the normal Internet route is unstable or unavailable due to DDoS attacks. By setting up indirect route via a set of proxy servers from the target side DNS server to the client side DNS server, the DNS zone data can be transported over Internet via the indirect routes to foil the DDoS attack. After the IR DNS update, the end users can get the indirect routing information by querying the DNS server, and set up indirect route to the target server accordingly. The IR DNS update is an essential part of the Secure COLlective Defense (SCOLD) system, and it can be utilized independently as an extension to the existing DNS update utility. This technique can also be utilized to protect the root DNS servers from DDoS attacks. The implementation of the IR DNS update on BIND 9 is presented. The experimental results show that the IR DNS update can be used to improve the network security, availability and performance.

Index terms – SCOLD, Secure DNS Update, Dynamic DNS Update, IR DNS Update, Indirect Route

I. INTRODUCTION

The Domain Name System (DNS) [1, 2] is one of the most critical and fundamental building blocks of today's Internet infrastructure. It is a hierarchically distributed database which provides the essential service of translating between domain names and IP addresses. DNS is also used to route email, store additional mapping information [13, 18], or provide other services [15, 16]. Berkeley Internet Name Domain package (BIND) [3] is the most widely deployed DNS implementation, which is openly redistributable.

In a survey conducted by the SANS Institute [4], the number-one Internet vulnerability reported by survey participants was DNS BIND weaknesses. The brief service disruption on the nine of the thirteen DNS root servers caused by DDoS bandwidth attacks on 2002 [5] is one of the most prominent attacks on DNS in recent years. In [17], D. Atkins provides an overview of existing DNS vulnerabilities and known threats.

The DNS was originally designed nearly two decades ago. Due to the increasing demands from Internet on DNS security, robustness and functionality, the DNS is undergoing a number of significant changes, and various DNS enhancements and new services have been suggested [7, 8, 9, 12, 13, 15, 16, 18].

A novel approach named SCOLD (the Secure COLlective Defense system) to defend against DDoS attack has been proposed by Edward Chow and Yu Cai in [6]. The key idea of SCOLD is to follow intrusion tolerance and network reconfiguration paradigm by providing 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. SCOLD utilizes indirect routes to keep communication channels open between clients and the attacked servers. Protocol software was developed for Linux systems. Preliminary experimental results show that SCOLD can significantly improve the network security, availability and performance.

One of the key techniques in SCOLD is the enhanced secure dynamic DNS update with indirect route. We refer to it as the IR DNS update. In SCOLD, the DNS system is utilized to store the indirect routing information, like the proxy server IP addresses; this DNS zone data is updated from target DNS server to client DNS servers through indirect route, since the normal route may be unstable or unavailable due to DDoS attacks; after the IR DNS update, the heterogeneous clients can query their DNS servers to get the indirect routing information, and use it to set up indirect routes to the target site.

Although the IR DNS update was originally designed for the SCOLD project, it can be utilized independently as a useful extension to the current dynamic DNS update and the secure DNS update [7, 8, 9, 10].

The balance of this paper is organized as follows. In Section 2, we give an overview of the SCOLD system,

Correspondence to: Yu Cai, Department of Computer Science, University of Colorado at Colorado Springs, Colorado Springs, CO 80933-7150, USA
Email: ycai@cs.uccs.edu

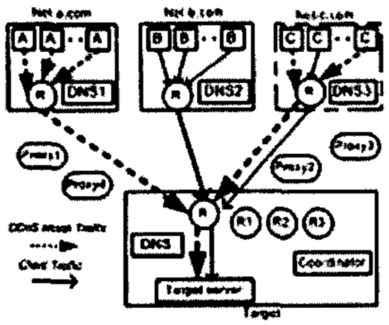

Figure 1: Target site under DDoS attack

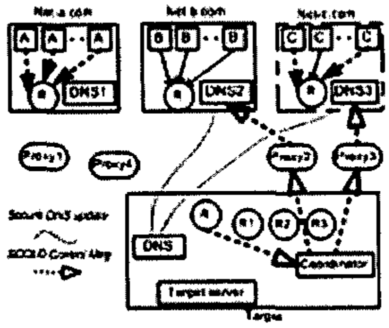

Figure 2: The control flow in SCOLD

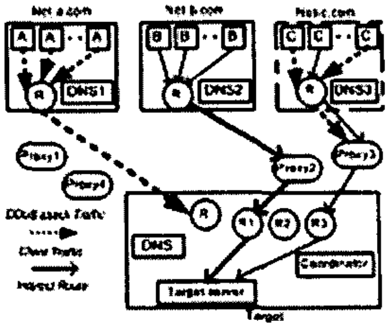

Figure 3: Indirect route in SCOLD

The balance of this paper is organized as follows. In Section 2, we give an overview of the SCOLD system, and present the related work on DNS extension. In Section 3, we present the design and implementation of the IR DNS update. In Section 4 we present the experimental results and simulation results. The conclusion is drawn in Section 5.

II. Background and Related Work

A. SCOLD overview

To better understand the IR DNS update, we first present an overview of SCOLD system.

Most organizations today deploy multiple gateways or multi-homing scheme. When the main gateway is under DDoS attack, the clients' traffic should be redirected to the alternate gateways. However, we may not want to reveal the alternate gateway IP addresses to public domain. Because once this information is exposed to public domain, the alternate gateways may become new targets of DDoS attack.

The SCOLD system defends against DDoS attacks by setting up indirect routes between clients and target server via a collection of geographically separated proxy servers and alternate gateways. The traffic between clients and target server is transported over Internet through the indirect routes.

Figure 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 the alternate gateways. In the figure the majority of the traffic from net-a.com is malicious, that of net-b.com is legitimate, and that of net-c.com is mixed.

Figure 2 shows the control flow of the SCOLD system. When the target site is under DDoS attacks, its Intrusion Detection System (IDS) raises an intrusion alert and notifies the SCOLD coordinator, which sits in the same or trusted domain of the target server. The coordinator then notifies some selected proxy servers (proxy 2 and 3 here) to set up indirect routes. The proxy servers notify the DNS servers of the legitimate clients to perform an IR 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 the IR DNS update, the client side DNS server gets the new DNS entry containing the designated proxy servers IP addresses. The clients query the DNS server, get the proxy server IP addresses, and can set up indirect routes to the target server via the selected proxy servers. The proxy servers examine the incoming traffic and relay it to a designated alternate gateway on the target site.

In SCOLD, the IP addresses of the alternate gateways and the SCOLD coordinator(s) are revealed only to the trustworthy proxy servers to protect them from being attacked by malicious clients. The clients in public domain can connect to the target side through the designed proxy servers.

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

Proxy servers can be provided by the participating organizations of SCOLD, or fee-based service providers.

B. Related work on DNS extension

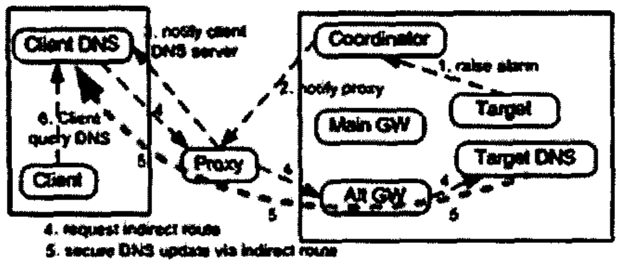

Figure 4: The IR DNS update

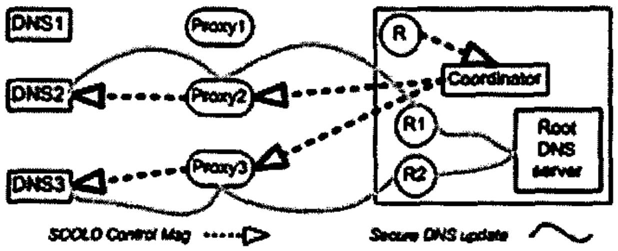

Figure 5: Protect the root DNS server

The DNS has been undergoing fundamental changes recent years. It has been noted that the current DNS is vulnerable to a wide range of attacks and faults. For example, attackers can intercept and change the DNS message; therefore, the clients' traffic will be redirected to a wrong place, or even a forged web site.

DNSSEC [7] (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.

To protect the signature key, the DNSSEC stored the key offline and use "off-line signing". However, the secure dynamic update complicates the situation by requiring frequent access to signature key and online signing process. In [19], the authors proposed a solution for distributing the signature keys through threshold cryptography. Therefore, the signature keys can be kept online and "online signing" becomes possible.

The public-key cryptography in DNSSEC may introduce significant overhead. In [20], a more efficient approach based on symmetric-key cryptography [20] has been proposed by building chains of trust from root servers to authoritative servers.

Another major DNS enhancement is dynamic DNS update protocol [8], which allows an entity to update a DNS record "on the fly". Dynamic DNS update can create caching issues and additional problems [22].

Dynamic DNS update was extended to secure DNS update by using a set of keys to authenticate an update [9, 10]. Digital signatures are stored in the DNS as SIG resource records and are used to encrypt and decrypt update messages for a zone.

Dynamic DNS update has been integrated with DHCP (Dynamic Host Configuration Protocol) [11]. DHCP is used to dynamically assign IP addresses to hosts in the network. With dynamic DNS update, a mapping between domain names and IP addresses can be maintained dynamically [12, 13].

EDNS0 (Extension Mechanisms for DNS) [21] is another DNS enhancement which allows DNS requestors to advertise the size of their UDP packets and facilitate the transfer of packets larger than 512 octets, the original DNS restriction for UDP packet size.

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 loading balancing and traffic distribution among a cluster of web servers has been studied in [15, 16]. The web servers are known by a single domain name, and DNS dynamically map the domain name to a real web server IP address based on loading balancing algorithm. Therefore, the clients' traffic will be routed to different real server.

In [23], 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 [14] is a resource discovery and service location system by mapping service name-attributes to name records using an intentional name language.

III. THE IR DNS UPDATE

A.Design and architecture

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

```
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).

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

Figure 4 illustrates how the IR 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 an IR DNS update. Step 4, if the client DNS server decide to make an IR 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, the client DNS server performs the secure DNS update and gets DNS records from target DNS server.

The IR DNS update can not only be used for zone data transfer in the scenario of DDoS attacks, but can also be used to protect the root DNS servers from large-scale DDoS attacks [5]. In Figure 5, DNS 1-3 are the client side DNS servers, and the main gateway R of the root DNS server is under sever DDoS attacks. Therefore, DNS 1-3 will experience significant delay or even fail to query the root DNS server. Due to the current DNS querying model, the end users may perceive a poor Internet performance with unbearable delay.

By utilizing the new technique we proposed, 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 the coordinator; the coordinator notifies the selected proxy servers (proxy 2, 3 here); the proxy servers notify the legitimate client DNS servers with their IP addresses; those DNS servers then set up indirect routes to the root DNS via the proxy servers and the alternate gateways; then the client DNS servers can query the root DNS server via indirect route.

In this proposed architecture, there are three tiers of defense. First, based on the preliminary intrusion detection result from the main gateway, some malicious or forged DNS servers will not be notified with indirect route. Second, the proxy servers are also equipped with IDS and firewall filters to further block malicious traffic. Third, the proxy servers are equipped with admission control and rate-limiting mechanism to enforce bandwidth throttling and control the aggressive clients.

B. Implementation

We implement the IR DNS update on BIND v. 9.2.2 and Redhat Linux v. 8 / 9. The indirect route is implemented by using IP tunnel protocol [24]. For further information on indirect route, please refer to [6].

Our implementation is summarized as follows.

1) The BIND 9 DNS server was modified to support the newly defined ALT type 99 data and to enable the automated setup of indirect route.

2) The DNS dynamic update utility (nsupdate [27]) was enhanced to support indirect routing and the new data type. The enhanced DNS update utility is called nsreroute.

3) On client side, the domain name resolve library (v.2.3.2) was enhanced to support the new data type. 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).

4) An agent program runs on each participating node (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 may be modified as needed at run time.

5) All the control messages are encrypted using Secure Sockets Layer (SSL) [26] and all participating nodes must be mutually authenticated. Experiments show that this is one of the major causes of overhead.

The authentication and encryption/decryption mechanism is a difficult decision, especially in large-scale distributed system. However, it is not the focus of this paper. Therefore, we adopt the most commonly-used public key cryptography used in OpenSSL [26] and DNSSEC [7].
The new DNS update utility "nsreroute" can be executed by a selected proxy server to initiate the IR DNS update from target DNS server to client DNS server. The command format is as follows.

```
nsreroute input_file
```

```
reroute client.clientnet1.com. victimDNSserver1.victimnet.com. victimDNSserver2.victimnet.com. <victim DNS 1 address> <victimDNS 2 address>
<proxy server address 1> <proxy server address 2> ... <proxy server address N>

reroute client.clientnet2.com. victimDNSserver1.victimnet.com. victimDNSserver2.victimnet.com. <victim DNS 1 address> <victimDNS 2 address>
<proxy server address 1> <proxy server address 2> ... <proxy server address N>
```

Figure 6. nsreroute and input file

"input_file" is a file that contains one or more entries, with a sample shown in the Figure 6. Each entry begins with the word "reroute", followed by a client domain name. The next two elements are always two of the victim's authoritative DNS servers (target DNS server). These are followed by each DNS server's respective IP address. The last items are proxy servers IP addresses as ALT type data. A carriage return separates entries.
The clocks in the participating computers must be in sync. Use of NTP (Network Time Protocol) [25] is strongly recommended.

IV. Experimental Results

A. Testbed setup

We set up a test bed consists of more than 20 nodes with various machine settings. The test bed includes 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.

B. Experimental results

A key concept in the IR DNS update and SCOLD is indirect route. We first evaluate the performance of indirect route compared to direct route, with and without DDoS attack respectively. The result is shown in Table 1. We can observe that there are overhead associated with indirect route. It mainly comes from the IP tunneling overhead and more Internet hops involved in indirect route. In Table 1, the overhead in term of response time is about 70%. Further experiments shows the overhead varies from 30%-200%. However, under DDoS attack, the response time of using direct route increases dramatically (15 times to infinity), while the response time of using indirect route keep the same (assuming no DDoS attack against proxy servers directly). Therefore, compared with the serious impact of DDoS attack, the overhead of indirect route is acceptable.

Next, we evaluate the overhead of the enhanced resolve library. Table 2 shows 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.

Table 1: Performance of Indirect Route vs. Direct Route, with and without DDoS attacks

Test	No attack			Under DDoS attack		
	Direct Route (a)	Indirect Route (b)	Indirect Route Overhead (b - a) / (a)	Direct Route (c)	Direct Route Delay (c) / (a)	Indirect Route (d)
Ping	49 ms	87 ms	77%	1048 ms	21 times	
HTTP(100k)	6.1s	11s	80%	109s	18 times	
HTTP(500k)	41s	71s	73%	658s	16 times	
HTTP(1M)	92 s	158s	71%	timeout	infinity	same as no attack
FTP(100k)	4.2 s	7.5s	78%	67s	16 times	
FTP(500k)	23 s	39s	69%	345s	15 times	
FTP(1M)	52 s	88s	69%	871s	17 times	

Table 2: 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

We evaluate the overhead of the enhanced BIND DNS server. Table 3 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.

We then evaluate the performance the IR DNS update. Table 4 shows how long it takes to set up an indirect route for an IR DNS update. This is mainly caused by the secure communication between participating nodes. We observe that it is pretty fast to set up an indirect route.

Table 5 shows the time to perform an IR DNS update after the indirect route is set up. We observe that the update time increases dramatically with the increase of the number of the authoritative client DNS servers to be updated. This suggests that there is a limit on how many client DNS servers one IR DNS update can handle concurrently. We suggest that one IR DNS update should handle no more than 10 client DNS servers concurrently.

Table 6 shows performance comparison between an IR DNS update with indirect route (using nsreroute) vs. a normal secure DNS update with direct route (using nsupdate), under and not under DDoS attacks respectively. It shows that the nsreroute 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 nsreroute with indirect route is almost not affected (assuming no DDoS attack against proxy servers directly).

Table 3: 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 4: Time to set up indirect route for IR DNS update

Time
2.4 s

Table 5: Time for an IR DNS update

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

Table 6: Performance of IR DNS update with indirect route (using nsreroute) vs. secure DNS update with direct route (using nsupdate), with and without DDoS attack

	No attack		Under DDoS attack			
	nsupdate (a)	nsreroute (b)	nsupdate (c)	nsreroute (d)	nsupdate Delay (c) / (a)	nsreroute Overhead (b-a) / (a)
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%

V. Conclusion

We present the design and implementation of the IR DNS update. It can be used to defend against DDoS attacks by redirecting the DNS update traffic between client DNS server and target DNS server through indirect routes via proxy servers and alternate gateways. It is an essential part of the SCOLD system, but can also serve as a useful extension to the existing DNS update utility. BIND 9 DNS package is modified to support IR DNS update. IP tunnel was utilized to implement indirect routing. New ALT 99 type data is defined and a new DNS update utility named nsreroute is developed. The preliminary results show that SCOLD can improve the network security, availability and performance.

VI. References

[1] P. Mockapetris, "Domain Names--Implementation and Specification", RFC 3658, Nov. 1987
[2] P. Mockapetris, "Domain Names--Concepts and Facilities", RFC 1034, Nov. 1987
[3] Internet Systems Consortium, "ISC BIND", http://www.isc.org/index.pl?/sw/bind/
[4] The SANS Institute, "How To Eliminate The Ten Most Critical Internet Security Threats" http://www.sans.org/top20/top10.php, 2001
[5] Internetnews.com, "Massive DDoS Attack Hit DNS Root Servers", http://www.internetnews.com/ent-ews/article.php/1486981
[6] Edward Chow, Yu Cai, "Secure Collective Defense system (SCOLD)", http://cs.uccs.edu/~scold/doc/SCOLD_globecom2004.doc, submitted to Globecom 2004
[7] DNSSEC, http://www.dnssec.net/
[8] Dynamic DNS update, RFC 2136, http://www.faqs.org/rfcs/rfc2136.html
[9] Secure DNS update, RFC 3007, http://www.faqs.org/rfcs/rfc3007.html
[10] Y. Shim, et al., "Extension and Design of Secure Dynamic Updates in Domain Name Systems", 5th Asia-Pacific Conference on Communications, 1998
[11] DHCP, http://www.dhcp.org/
[12] C. Park, et.al., "The Improvement for Integrity between DHCP and DNS", High Performance Computing on the Information Superhighway, pp. 511-516, 1997.
[13] "Network Management: NetID 4.1", http://www.nortelnetworks.com/products/02/datasheets/2894.pdf
[14] W. Adjie, et al., "The design and implementation of an intentional naming system", Operating Systems Review, vol.35, pp. 186-201, 1999.
[15] V. Cardellini, et al., "Redirection Algorithms for Load Sharing in Distributed Web-server Systems", Proc. of 19th IEEE Int. Conf. on Distributed Computing Systems, pp. 528-535, 1999
[16] Eddie, Enhanced DNS Server, http://eddie.sourceforge.net/lbdns.html
[17] D. Atkins, et al., "Threat Analysis of the Domain Name System", http://www.lasr.cs.ucla.edu/classes/239_1.spring03/papers/dns_threats.html, 2002
[18] M. Richardson, "A method for storing IPsec keying material in DNS",

http://www.sandelman.ottawa.on.ca/SSW/ietf/ipsec/key/draft-richardson-ipsec-rr.html, 2003
[19] Christian Cachin, et al., "Secure Distributed DNS", IBM research report, 2003
[20] G. Ateniese, et al., "A new approach to DNS security (DNSSEC)", Proceedings of the 8th ACM conference on Computer and Communications Security, 2001
[21] EDNS0, "Extension Mechanisms for DNS", ftp://ftp.isi.edu/in-notes/rfc2671.txt
[22] Yair Amir, "A new look at the old domain name system", technical report, Johns Hopkins University, 2003.
[23] M. Richardson, "A method for storing IPsec keying material in DNS", http://www.sandelman.ottawa.on.ca/SSW/ietf/ipsec/key/draft-richardson-ipsec-rr.html, 2003
[24] IP Tunnel, http://www.linuxforum.com/linux-advanced-routing/lartc.tunnel.ip-ip.html
[25] NTP, "Network Time Protocol", http://www.ntp.org/
[26] OpenSSL, http://www.openssl.org/
[27] nsupdate, http://www.linuxforum.com/man/nsupdate.8.php