# DNSSECの運用展開に関する一考察

力武 健次[†] 中尾 康二[†,††] 下條 真司[†††] 野川 裕紀[††††]

[†] 独立行政法人 情報通信研究機構 セキュリティ高度化グループ 〒184–8795 東京都小金井市貫井北町 4–2–1
[††] KDDI株式会社 技術開発本部 情報セキュリティ部 〒102–0072 東京都千代田区飯田橋 3–10–10
[†††] 大阪大学 サイバーメディアセンター 〒567–0047 大阪府茨木市美穂が丘 5–1
[††††] 東京医科歯科大学 情報医科学センター 〒113–8510 東京都文京区湯島 1–5–45
E-mail: †rikitake@nict.go.jp

**あらまし** DNSSECはDNS（ドメイン名システム）の提供する情報の詐称を防ぐためのプロトコル群である．DNSSECの正常動作は取り扱うべきUDPペイロード長の増加，およびそれに伴い発生する分割されたIPパケットの伝送保証を前提としている．しかし，この基礎的な前提はセキュリティ保持のためのパケットフィルタリング機構の設定により，インターネット上では多くの場合満たされていないのが現状である．DNSSECの正常動作のための前提条件が満たされないことは，今後のDNSSEC展開の上で無視できない障害になる．本研究では，DNSSECを実運用で展開していく際に発生するセキュリティ上の諸問題について，DNSで使うトランスポートプロトコルの問題を主に考察を行い，今後予想できる問題とその対策について提案する．

**キーワード** DNS（ドメイン名システム），DNSSEC，トランスポートプロトコル，パケット分割，ペイロード長，ネットワーク運用

# A Study of DNSSEC Operation and Deployment

Kenji RIKITAKE[†], Koji NAKAO[†,††], Shinji SHIMOJO[†††], and Hiroki NOGAWA[††††]

[†] Security Advancement Group, NICT, Japan 4-2-1, Nukui-Kitamachi, Koganei City, Tokyo 184-8795 Japan
[††] Information Security Department, KDDI Corporation
3-10-10, Iidabashi, Chiyoda-Ku, Tokyo 102-0072 Japan
[†††] Cybermedia Center, Osaka University 5-1, Mihogaoka, Ibaraki-City, Osaka 567-0047 Japan
[††††] Information Center for Medical Sciences, Tokyo Medical and Dental University
1-5-45, Yushima, Bunkyo-Ku, Tokyo 113-8510 Japan
E-mail: †rikitake@nict.go.jp

**Abstract** DNSSEC is a set of authentication protocols for preventing forgery of information provided by the DNS (Domain Name System). DNSSEC assumes the handling capability of larger UDP payloads and guaranteeing transmission of their fragmented IP packets to the software for the proper operation. These fundamental assumptions are, however, *not* satisfied over the Internet in most cases because of the configuration of the packet-filtering devices. The failure of satisfying the assumptions for the proper operation of DNSSEC will become a significant impediment to the deployment of DNSSEC. In this paper, we review and analyze the security issues on deploying DNSSEC to the real-world Internet production systems focused on the lower-layer transport protocols, and propose possible solutions to the arising problems.

**Keywords** DNS (Domain Name System), DNSSEC, transport protocol, packet fragmentation, payload length, network operation

## 1. Introduction

DNS has been one of the most critical application subsystem of the Internet systems, since every other application software depends on it, for resolving the domain names to actual resources, such as IP addresses or mail exchanging hosts. As the Internet becomes where the business and commerce activities happen, maintaining the integrity and genuineness of each resource records (RRs) represented by the DNS becomes more crucial for guaranteeing the Internet as a safe and secure place, as the other parts of the real world should be.

DNS RRs are considered to be public, i.e., open worldwide, so the integrity maintenance of DNS means preventing the RR forgeries, keeping the same content of RRs visible throughout the Internet, and the updates of RRs be accessible as soon as they are available. Under this principle, in this paper we will not focus on existing methods providing the access control of DNS Zones and the belonged RRs, such as TSIG [1] of Secure DNS Update [2].

DNS Security Extensions also known as *DNSSEC*, is designed to prevent RR forgeries by authenticating each RR with the digital signature of the authoritative origin domain and the upper-level domain authorities. Since current DNS completely lacks of any authentication, anyone can forge DNS answer RRs and can redirect a client host to a forged server, which could be used for criminal activities such as the identity theft or defamation of the contents of a Web server. DNSSEC has long been promoted by the IETF (Internet Engineering Task Force) *dnsext* and *dnsop* WGs (Working Groups), and implementations such as BIND [3] and NSD [4] have been publicly available.

One of the problem for the wide-area DNSSEC deployment which has not been widely addressed, however, is that DNSSEC does neither cover the protocol attack vulnerabilities of the DNS itself, nor compensate or augment the limitation of the lower-layer transport protocols. While DNSSEC assumes the lower-layer transport protocols provide complete transparency which is necessary for exchanging the data, this assumption often fails in the current real-world Internet environment.

In this paper, we review and analyze the security issues on deploying DNSSEC to the real-world Internet production systems, and propose possible solutions and open issues to the arising problems, as DNSSEC is going to be widely deployed. In later sections, we first discuss the general issues on DNSSEC deployment to the production-level Internet systems in Section 2. We then analyze the transport-protocol issues of DNSSEC payload exchange in Section 3. Our conclusion and future works are presented in Section 4.

## 2. DNSSEC Deployment Issues to Production Systems

DNSSEC development and standardization have been going on for 9 years as of March 2006. DNSSEC was first proposed as RFC2065 [5] in January 1997, and later updated as RFC2535 [6] on March 1999, though the whole scheme was revised later and republished on March 2005 as a set of three RFCs also informally known as *DNSSECbis*, RFC4033 [7] as the introduction, RFC4034 [8] as the definitions of RRs, and RFC4035 [9] for the protocol changes. One of the major changes of DNSSECbis from RFC2535 is the simplification of authority delegation by introducing ZSKs (Zone Signing Keys) and KSKs (Key Signing Keys).

The *dnsext* WG has its own charter Web page [10] as well as the *dnsop* WG does [11]. While the *dnsext* WG focuses on DNSSEC technical details on the application data exchange according to the past discussion records of their *namedroppers* mailing list [12], the *dnsop* WG rather focuses on the operational issues of established protocols, as shown on their *dnsop* mailing list [13].

The RFCs and Internet-drafts referred by the charter pages of the both WGs show a standard set of reference documents to know how the DNSSEC research and development are going on. Since the *dnsop* WG has so far published only one document for DNSSEC [14] as of the end of March 2006, most of the documents on DNSSEC details are published by the *dnsext* WG.

The issues addressed in the *dnsext* and *dnsop* WGs are mostly focused onto the application-level protocol extension details, as described in the introductory document [7], including:

- the realization of authentication chain between the DNS zone authorities using a public-key infrastructure (PKI) from the Root Domain to the leaf zones;
- the necessary RR type extensions and usage definitions, such as the DS, NSEC and RRSIG RRs [8];
- the necessary behavior extensions of the security-aware servers and resolvers [9];
- the operational guidelines on the key management such as key generation, rollover and storage, time management due to the introduction of absolute time in DNSSEC [14]; and
- other experimental documents, including how to conduct public DNSSEC experiments [15], evaluating transitional methods for authenticated denial of existence [16], and an opt-in-based authority delegation to the subzones of an authenticated zone [17].

On the other hand, the issues which are still *not* well-addressed in detail by the IETF DNSSEC documents are mostly on the operational issues of its deployment into the production-level Internet systems, including:

- feasibility analysis of the requirement of being able to handle larger-than-512byte UDP payloads on the real-world Internet production systems, though the notification mechanism enhancement using a pseudo RR in the DNS application-level protocol is documented as EDNS0 [18] and mandated in RFC4033 [7] Section 3 and RFC4035 [9] Section 4.1;

- quantitative evaluation of DNS traffic increase by exchanging larger payloads more frequently for verifying the digital signatures and the chain of authentication, though the issues are very briefly noted in the RFC3226 [19] Section 2.4 and 2.5; and
- quantitative evaluation of computational resource consumption by mandating PKI to the security-aware servers and resolvers, which will result in a large increase of CPU time of the hosts running DNS software.

While these issues mentioned above look trivial to find out the solutions, we have learned in the real-world production-level Internet systems, these issues are left unresolved in many cases. For example, on the lower-layer transport protocol issues, many protocol filters and proxies which are in between the two end-to-end DNS communicating hosts and servers may not provide the necessary transparency assumed by the DNSSEC.

In this paper, we address the DNS transport-protocol issues in next section which may become a significant impediment against deploying DNSSEC worldwide.

## 3. Transport-protocol Issues on DNSSEC Payload Exchange

DNSSEC requires complete transparency on the transport protocols used between the resolvers and servers. This requirement is not necessarily satisfied in the production-level Internet nowadays.

### 3. 1. Large UDP Payload Size on DNSSEC and The Possible Consequences

One of the examples which prevents DNSSEC deployment is that DNSSEC requires considerably larger size of payloads being exchanged by the resolvers (clients) and the servers. We have found that the mean value of the payload length of DNSSEC-signed RRs are 3 to 6 times larger than the unsigned original RRs, under a simulated condition that at least one RRSIG RR is attached as the signature of every returned RRset [20], which is true in security-aware exchange between the DNSSEC resolvers and servers. While the actual condition may change, additional RRSIG RRs largely contribute to increase the length of each payload of the UDP message, since each RRSIG RR has 1024bits or 128bytes in its length under the RSA/SHA1 algorithm [21].

The increase of DNS message size will break the lower-level transport reliability which assumes that each DNS message will not be fragmented, since IPv4 allows only 1472 bytes for UDP payloads in single packet, where IPv6 allows only 1232 bytes, regarding the default MTUs of both protocols (the size in bytes: 1500 for IPv4 [22], 1280 for IPv6 [23]). The link MTU values may become even smaller in a restricted environment such as through a point-to-point link.

For DNSSEC-enabled access through UDP:

```
dig +novc +norec +dnssec @b.ns.se se ns
```

(The `+novc` option forces the command to use UDP only.)

For DNSSEC-enabled access through TCP:

```
dig +vc +norec +dnssec @b.ns.se se ns
```

(The `+vc` option forces the command to use TCP only.)

Fig. 1 Command lines used for DNSSEC-enabled access

We faced an interesting case while testing DNSSEC accessibility through the public Internet. NIC-SE, the registrar for `.se` Country TLD (Top Level Domain), has started the service test of providing signed zone information [24]. While we confirmed the NS RRset for a `.se` authoritative name server was accessible from two hosts connected to two different Japanese ISPs (Internet Service Providers) via TCP, the access via UDP with EDNS0-enabled larger payloads were not successful through either of the Japanese ISPs, using the `dig` command of the BIND 9.3.2 on January 9, 2006, and on March 29, 2006. Figure 1 shows the command and the options used.

We later learned that only the first IP fragment packet of the UDP was delivered from a `.se` authoritative server and all the IP packets contained the TCP payload were with the "don't fragment" (DF) bit set, as we analyzed the answer packets by the `tcpdump` [25] packet-dump analysis software. The DF-set packet usage of TCP is an expected behavior of hosts enabled the Path MTU discovery algorithm [26], and is common among many servers such as those using FreeBSD [27].

From the above observation, we conclude that in the real-world Internet, many ISPs *may block any fragmented IP packets*, since TCP, ICMP, and *unfragmented* UDP packets such as those of non-DNSSEC DNS software and NTP [28] still work without problems, even if the fragment IP packets are blocked. While we do not have the information whether the ISPs we used for the experiment explicitly blocks the fragmented IP packets, we conclude that it is highly plausible that the assumption of DNSSEC for large fragmented UDP packets are broken in many parts of the real-world production-level Internet systems.

### 3. 2. Loss of Transparency on Large UDP Packet Delivery

As shown in Section 3. 1, the transparency of fragmented IP packets is being lost in many parts of the global Internet. We assume one of the reasons for this imposed limitation is that a vulnerability using the IP packet fragments has been widely known to forge TCP headers and packets, since Internet system administrators should block any security hole once they know it.

Ziemba et al. [29], have discovered that IP fragmentation can be used to disguise TCP packets from IP filters, using a very small size of IP fragments, called Tiny Fragment Attack. In the same document, the authors also explain another attack of rewriting the header information of a TCP packet using the existing IP-level packet reassembly capability required by IPv4, called Overlapping Fragment Attack. Miller [30] updates the details of the attack prevention scheme against Tiny Fragment Attack.

We should emphasize that on many Internet Service Provider networks *fragmented IP packets simply do not get through.* Applying this for *all* IP packets including UDP, is overkill and not necessarily the optimal practice. Nevertheless, this restriction is surely an effective prevention method against Tiny and Overlapping Fragment Attacks, *provided that no application would use the fragmented UDP packets*, which is true under the condition where only non-secure DNS is the critical UDP application for running a production-level system connected to the global Internet.

While a *dnsext* WG document, RFC3226 [19], which describes message-size requirements on DNSSEC servers shows the prediction on UDP .vs. TCP choice for DNS messages on its Section 2.4, it does not mention anything about the risks of allowing IP fragmentation on the real-world Internet.

Many firewalls and proxies which are in between the DNS communication path may block the EDNS0-enabled UDP packets. For example, NIC-SE suggests some early firewall products do not allow EDNS0-enabled UDP DNS packets to get through [31] without necessary reconfiguration. While in this document NIC-SE also suggests restricting EDNS0 maximum packet size to 512 bytes to retain downward compatibility with non-DNSSEC DNS packets, this will significantly reduce the usability of DNSSEC, because 96.5% of the signed DNS answers with additional records exceeds the 512-byte limit in our simulation [20].

### 3. 3. Congestion Control of DNSSEC UDP Usage and Timing Sequence of Falling-back to TCP

Since UDP does not have its own congestion or any timing control function, the application software must apply a proper method to prevent congestion in a wide-area network, where latency becomes a major factor of reliability for the packet delivery. In this section, we first point out the importance of historical perspective of NFS (Network File System), and show two completely-different ways of query-time control in DNS implementations.

We need to mention that in the NFS, a popular UDP-based application used over the Internet, congestion control and transport selection is a major issue when the service is provided throughout the public Internet. While each DNS servers and resolvers do not necessarily generate the large amount of traffic data as the NFS clients and servers do, the numbers of DNS resolvers and servers running on the Internet is huge, and the traffic usage caused by the protocol change should be assessed.

**a)** The function performs pseudo-exponential timeout backoff for sending UDP queries for four times, using the intervals of 1, 3, 11, 45, in seconds respectively;
**b)** if the function receives the answer with the TC bit set it tries to retrieve the answer by TCP, waiting 10 seconds for the connection and 10 more seconds for the response; and
**c)** **a)** and **b)** are sequentially repeated for each server if two or more authoritative servers have to be looked up.

Fig. 2 Timing description for the function `dns_transmit()` of djbdns.

- The minimum retry interval for the UDP queries is 5 seconds;
- the interval is expanded for $2^n/n$ times when the number of servers to lookup is $n$;
- the retry interval remains the same during the queries;
- the TCP queries are performed before the UDP queries if explicitly specified by `RES_USEVC` option, or if an answer is received with the TC bit set; and
- the TCP queries will be performed only once for each server.

Fig. 3 Timing description for the function `res_nsend()` of BIND-derived resolver in FreeBSD.

In an NFS Version 4 (NFSv4) design document RFC2624 [32] Section 7.1, the author of the document emphasizes that TCP must be used for congestion control or management, even the author admits the throughput advantage of UDP to TCP. In the later protocol specification of the NFSv4, RFC3530 [33] Section 3.1, the authors write:

> ... the supported transports between NFS and IP MUST be among the IETF-approved congestion control transport protocols, which include TCP and SCTP [34] [35].

Since DNS is a distributed database, the congestion control issue on the UDP usage should be well-defined, at least as the level of NFS, though the protocol nature might be different from NFS.

We should also note that the retransmission timing details and the algorithm to choose UDP and TCP for sending the queries from DNS resolvers are complex and not well-explained. While setting the TC bit in the DNS message header, which is defined in RFC1035 [36] Section 4.2.1 and later clarified in RFC2181 [37] Section 9, shows how to fallback from UDP to TCP when UDP DNS answer payload exceeds 512 bytes, the details of handling multiple servers and fallbacks are not thoroughly defined.

We have learned that only one detailed timing and falling-back-to-TCP specification is currently available as a written document for implementations of DNS resolvers, which is about `dns_transmit()` library function [38], representing the resolver interface for djbdns [39]. In this document, it is shown that the function `dns_transmit()` employs the algorithm described in Figure 2.

On the other hand, for BIND-Version-8-derived DNS resolver in FreeBSD Version 4, you need to refer to the actual source code files under the `/usr/src/contrib/bind` tree since the timing sequence is not documented. As far as we have learned from the source code of FreeBSD 4.11-RELEASE-p16, the resolver function `res_nsend()` employs the algorithm described in Figure 3.

The fact that the algorithms described in Figure 2 and Figure 3 are completely different from each other, suggests some sort of recommendation should be made for DNS UDP retransmission and TCP fallback sequence timings for defining the transport protocol timing of DNS in details, so that the behavior of DNS software become more robust against and aware of possible congestion happening on the large-scale Internet systems. This standardization will also help the analysis of DNS software by a network packet simulation.

We should also note that still major portion of DNS software employs the algorithm in Figure 3, since BIND and the derived software have been the most popular DNS software implementation on the Internet. With this algorithm, congestion is expected when a large number of queries are sent simultaneously to a server, since no exponential backoff is performed for resending UDP queries.

### 3.4. DNS Amplification Attack

While distributed denial-of-service (DDoS) attacks are out of scope of DNSSEC specification, DNS-specific DDoS attacks may also significantly impact the DNSSEC itself and the whole Internet. We should note that mismanagement of DNS servers have already established a remotely-controllable DDoS attack network. Since DNSSEC handles much larger legitimate packets than non-secure DNS, the impact will even become more significant.

Vaughn and Evron [40] show an *open resolver*, which accepts a recursive DNS query from *any* site, can be exploited to generate a DNS answer packet to an arbitrary host specified by the spoofed source IP address, i.e., the target's address, when the spoofed query packet contains a request for a valid domain name with a large TXT RR. Since this attack *amplifies* 60-byte length of a query packet to a response larger than 4000 bytes by the scale factor of more than 60, this is called *DNS Amplification Attack*. The authors states about 580000 open resolvers are already located. US-CERT [41] and JPCERT/CC [42] have released the warnings for this attack.

The implication of DNS Amplification Attack to DNSSEC is significant since:

- if an open DNSSEC-enabled resolver were victimized, the whole DNSKEY and RRSIG RRs could be used for the amplification results, though the possibility of inducing such an answer packet is low;
- many sites would disable EDNS0 for circumventing the attacks, though this is not a dependable solution since the reduced UDP payload length limits could be countered by increasing the request rates [41]; and
- if a badly-managed DNSSEC-enabled resolver were compromised to regenerate a legitimate query to a DNSSEC-enabled non-recursive server, the server could also be victimized by consuming the processing power for DDoS attacks.

The IP-spoofing protection methods such as ingress filtering [43] and proper access control to recursive DNS resolvers to prevent allowing the open access are effective measures against DNS Amplification Attack, as well as other reflector-based DDoS attacks [44]. The large number of 580000 open resolvers, however, suggests that the opportunity for DNS Amplification Attack is high and that more cases of the attack will be reported.

## 4. Conclusion and Future Works

In this paper, we reviewed and analyzed the security issues on deploying DNSSEC to the real-world Internet production systems, from the operational perspective and the usage of lower-level transport protocols. Since DDoS attacks have been so popular on the Internet these days, a detailed traffic volume analysis imposed by the larger payloads of DNSSEC signature and the fragmented IP packets caused by the usage of UDP large payloads should be thoroughly performed before deploying the DNSSEC to the public Internet.

We also suggest that some methods should be standardized for the congestion control of UDP packets and the fallback sequence from UDP to TCP transports of DNS so that the behavior of DNS software become more robust against congestion, and will also help the analysis of DNS software by a network packet simulation.

While DNSSEC itself is not a solution for DDoS attacks, the precaution against possible DDoS such as DNS Amplification Attack, has to be analyzed from the DNSSEC protocol design points of view.

The important issues on DNSSEC deployment which are not addressed in this paper include, but not limited to:

- DNS-resolver-implemented appliances which lack of sufficient CPU processing power to resolve DNS requests;
- DNS server hosts in the production systems which do not have sufficient source of secure random bit sequences for necessary public-key rollover sequences; and
- the migration cost for the DNS registrars and the leaf domain administrators of the management to establish the PKI chain of trust and keep the data to each security-aware hosts by themselves.

## Acknowledgements

Our thanks go to Masato Minda of Japan Registry Services, Co., Ltd., Youki Kadobayashi of Nara Advanced Institute for Science and Technology, Tsunehiko Suzuki of Chukyo University, and Peter Koch of DENIC eG (also IETF *dnsop* WG chair), for valuable suggestions on DNS protocol and security issues.

## References

[1] P. Vixie, O. Gudmundsson, D. Eastlake, and B. Wellington, “Secure Key Transaction Authentication for DNS (TSIG),” 2000. RFC2845.

[2] B. Wellington, “Secure Domain Name System (DNS) Dynamic Update,” 2000. RFC3007.

[3] Internet Software Consortium, “BIND.” `http://www.isc.org/bind/`.

[4] NLnet Labs, “Name Server Daemon (NSD).” `http://www.nlnetlabs.nl/nsd/`.

[5] D. Eastlake 3rd and C. Kaufman, “Domain Name System Security Extensions,” 1997. RFC2065.

[6] D. Eastlake, “Domain Name System Security Extensions,” 1999. RFC2535.

[7] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “DNS Security Introduction and Requirements,” March 2005. RFC4033.

[8] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “Resource Records for the DNS Security Extensions,” March 2005. RFC4034.

[9] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, “Protocol Modifications for the DNS Security Extensions,” March 2005. RFC4035.

[10] IETF dnsext WG, “DNS Extention (dnsext) Working Group Charters.” `http://www.ietf.org/html.charters/dnsext-charter.html`.

[11] IETF dnsop WG, “Domain Name System Operations (dnsop) Working Group Charters.” `http://www.ietf.org/html.charters/dnsop-charter.html`.

[12] IETF dnsext WG, “The *namedroppers* mailing list.” The archive available at `http://ops.ietf.org/lists/namedroppers/`.

[13] IETF dnsop WG, “The *dnsop* mailing list.” The archive available at `http://darkwing.uoregon.edu/~llynch/dnsop/`.

[14] O.M. Kolkman and M. Gieben, “DNSSEC operational guidelines,” 2006. INTERNET-DRAFT draft-ietf-dnsop-dnssec-operational-practices-08.txt, approved as an RFC by IESG as of March 2006.

[15] D. Blacka, “DNSSEC experiments,” 2006. INTERNET-DRAFT draft-ietf-dnsext-dnssec-experiments-02.txt.

[16] R. Arends, P. Koch, and J. Schlyter, “Evaluating DNSSEC Transition Mechanisms,” 2006. INTERNET-DRAFT draft-ietf-dnsext-dnssec-trans-03.txt.

[17] R. Arends, M. Kosters, and D. Blacka, “DNSSEC Opt-In,” 2006. INTERNET-DRAFT draft-ietf-dnsext-dnssec-opt-in-08.txt.

[18] P. Vixie, “Extension Mechanisms for DNS (EDNS0),” 1999. RFC2671.

[19] O. Gudmundsson, “DNSSEC and IPv6 A6 aware server/resolver message size requirements,” Dec. 2001. RFC3226.

[20] K. Rikitake, H. Nogawa, T. Tanaka, K. Nakao, and S. Shimojo, “An analysis of DNSSEC transport overhead increase,” IPSJ SIG Technical Reports 2005-CSEC-28, vol.2005, no.33, pp.345–350, March 2005. ISSN 0919-6072.

[21] D. Eastlake, “RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS),” May 2001. RFC3110.

[22] R. Braden (Editor), “Requirements for Internet Hosts – Communication Layers,” 1989. RFC1122.

[23] S. Deering and R. Hinden, “Internet Protocol, Version 6 (IPv6) specification,” 1998. RFC2460.

[24] NIC-SE, “`.se` is the first TLD in the world with DNSSEC – a more secure technique for name resolving on the Internet.” NIC-SE Press Release on September 14, 2005, `http://www.nic.se/english/nyheter/pr/2005-09-14?lang=en`.

[25] TCPDUMP Public Repository, “tcpdump.” `http://www.tcpdump.org/`.

[26] J. Mogul and S. Deering, “Path MTU Discovery,” 1990. RFC1191.

[27] The FreeBSD Project, “FreeBSD.” `http://www.freebsd.org/`.

[28] D. Mills, “Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI,” 1996. RFC2030.

[29] G. Ziemba, D. Reed, and P. Traina, “Security Considerations for IP Fragment Filtering,” 1995. RFC1858.

[30] I. Miller, “Protection Against a Variant of the Tiny Fragment Attack,” 2001. RFC3128.

[31] NIC-SE, “Firewall & DNSSEC.” `http://dnssec.nic.se/fw/en.html`.

[32] S. Shepler, “NFS Version 4. Design Considerations,” 1999. RFC2624.

[33] S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C. Beame, M. Eisler, and D. Noveck, “Network File System Version 4 Protocol,” 2003. RFC3530.

[34] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson, “Stream Control Transmission Protocol,” 2000. RFC2960.

[35] J. Stone, R. Stewart, and D. Otis, “Stream Control Transmission Protocol (SCTP) Checksum Change,” 2002. RFC3309.

[36] P.V. Mockapetris, “Domain names – implementation and specification,” 1987. RFC1035 (also STD13).

[37] R. Elz and R. Bush, “Clarification to the DNS specification,” 1997. RFC2181.

[38] D.J. Bernstein, “The `dns_transmit` library interface.” `http://cr.yp.to/djbdns/dns_transmit.html`.

[39] D.J. Bernstein, “djbdns.” `http://cr.yp.to/djbdns.html`.

[40] Randal Vaughn and Gavi Evron, “DNS Amplification Attacks (*Preliminary Release*).” `http://www.isotf.org/news/DNS-Amplification-Attacks.pdf`.

[41] US-CERT, “The Continuing Denial of Service Threat Posed by DNS Recursion.” `http://www.us-cert.gov/reading_room/DNS-recursion121605.pdf`.

[42] JPCERT/CC, “Distributed Denial of Service Attacks using Recursive DNS Queries (written in Japanese).” `http://www.jpcert.or.jp/at/2006/at060004.txt`.

[43] P. Ferguson and D. Senie, “Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing,” 2000. RFC2827.

[44] V. Paxson, “An analysis of using reflectors for distributed denial-of-service attacks,” Computer Communication Review, vol.31, no.3, pp.38–47, 2001.