SRI International

CSL Technical Report SRI-CSL-05-02 ● July 27, 2005

Denial of Service against the Domain Name System: Threats and Countermeasures

Steven Cheung

This research is sponsored by the Department of Homeland Security under contract number NBCHF040146. The views herein are those of the author and do not necessarily reflect the views of the sponsoring agency.

Computer Science Laboratory ● 333 Ravenswood Ave. ● Menlo Park, CA 94025 ● (650) 326-6200 ● Facsimile: (650) 859-2844

Abstract

The Domain Name System (DNS) is a core component of the Internet infrastructure. Many network services such as the Web and electronic mail rely on DNS. Thus it is critical to protect DNS from denial-of-service (DoS) attacks. This paper analyzes the DoS threats against DNS, and reviews existing and proposed countermeasures for addressing those threats.

Keywords: Denial of service, domain name system, network infrastructure, threat analysis, countermeasures

Chapter 1

Introduction

The Domain Name System (DNS) is a core component of the Internet infrastructure. Many network services rely on DNS to function. For instance, DNS is used to translate human-level (domain) names such as www.cnn.com to the IP addresses of the CNN's Web servers. If DNS is unavailable, a diverse range of network services, including the Web and electronic mail, may cease to work. Denial of service attacks (DoS) against DNS prevent DNS requests from being served, and have many forms, from corrupting the configuration of a DNS component to flooding-based distributed DoS attacks. This paper analyzes the DoS threats against DNS, and reviews security mechanisms and practice for countering these threats.

In recent years, a number of high-profile attacks against the availability of DNS have occurred, which affected various e-commerce, software company, and news Web sites, content distribution services, ISPs, and Internet infrastructure components. For example, in October 2002, a coordinated distributed DoS attack flooded all root servers simultaneously with high-volume network traffic. Although the root servers were reportedly able to keep up with the traffic, some root servers were unreachable or incurred significant increase in response time from many parts of the Internet because of the network congestion caused by the attack [10, 12, 34].

The organization of the rest of this paper is as follows. Chapter 2 presents an attack tree for characterizing the security threats against the availability of DNS. Chapter 3 describes existing and proposed countermeasures for addressing those threats. Chapter 4 presents concluding remarks.

Chapter 2

Attack Tree

In the attack tree methodology [30], a root node corresponds to an attacker's goal. A tree node may have a set of children. These child nodes correspond to way(s) to achieve the parent node. An attack tree, developed by successive refinements, represents the security threats against a system with respect to the goal. An internal (tree) node can be either an AND node or an OR node. If a node is an AND node, all its child nodes need to be achieved to achieve that node. On the other hand, an OR node can be achieved by achieving only one of its child nodes.

The leaf nodes of an attack tree can be assigned a security value (e.g., amount of resources needed to achieve this node, or whether a node is achievable). Based on these values, one can compute the security value for the goal (i.e., the root node). Using this methodology, "what if" questions can be asked—for example, when a security mechanism is added to a system, it may change the values of some nodes, and the security value of the root node may then be recomputed.

Previous papers [6,8,31,32] have described vulnerabilities of DNS. Here, we build on the existing work, and analyze the DoS threats against DNS in more detail.

In a typical DNS name resolution[1] scenario, there are several potential "attack points" for disrupting the resolution process:

[1] The process of retrieving data from DNS is called *name resolution*, or simply *resolution*. There are two modes of resolution in DNS: iterative and recursive. In the *iterative* mode, when a name server receives a query for which it does not know the answer, the server will refer the querier to other name servers that are more likely to know the answer. DNS servers are initialized with the IP addresses of root name servers. Moreover, the root servers know the authoritative servers of the top-level domains (e.g., `com`). A top-level domain server knows the authoritative servers of its second-level domains (e.g., `example.com`), and so on. By following the hierarchical structure, the querier can get "closer" to the answer after each referral. In the *recursive* mode, a server either answers the query based on the DNS data it has or finds out the answer by contacting other servers itself and then returns the answer to the querier.

1. stub resolver[2] host
2. communication between the stub resolver and the first-hop name server for recursive name resolution
3. first-hop name server
4. communication between the first-hop name server and the other name server(s), if needed, for iterative name resolution
5. other name server(s)

Based on the above, we developed an attack tree, depicted in Figure 2.1. The root node of the attack tree corresponds to the goal of causing DNS resolution to fail. Moreover, there are three subgoals for achieving it—namely, attack resolver hosts (see Section 2.1), disrupt communication (see Section 2.2), and attack name servers (see Section 2.3).

2.1 Attack Resolver Hosts

Goal: Attack resolver hosts to make them unable to contact a recursive name server (OR)

1. Exploit a vulnerability of resolvers to cause them to fail or misbehave
2. Corrupt resolver configuration settings
3. Attack the TCP/IP stack of the resolver host

Like other software pieces, one may be able to attack DNS resolvers by exploiting their vulnerabilities such as buffer overflow attacks (e.g., [22]) and cause them to misbehave or crash.

Another attack vector is modifying resolver configuration files, say through computer viruses, Trojans, or worms. Resolver configuration files may include information about the DNS servers used for name resolution. An attack that replaces the name server IP addresses in a resolver configuration file with bogus ones could cause denial of service. Also, changing other directives in resolver configuration files may affect the name resolution process. For example, a resolver may be configured to perform “name completion” by appending certain suffixes to an “incomplete” domain

[2] Stub resolvers are the most common form of DNS client. They do only the minimal job of constructing DNS queries, sending them to name servers for recursive resolution, and resending the queries if timeouts occur.

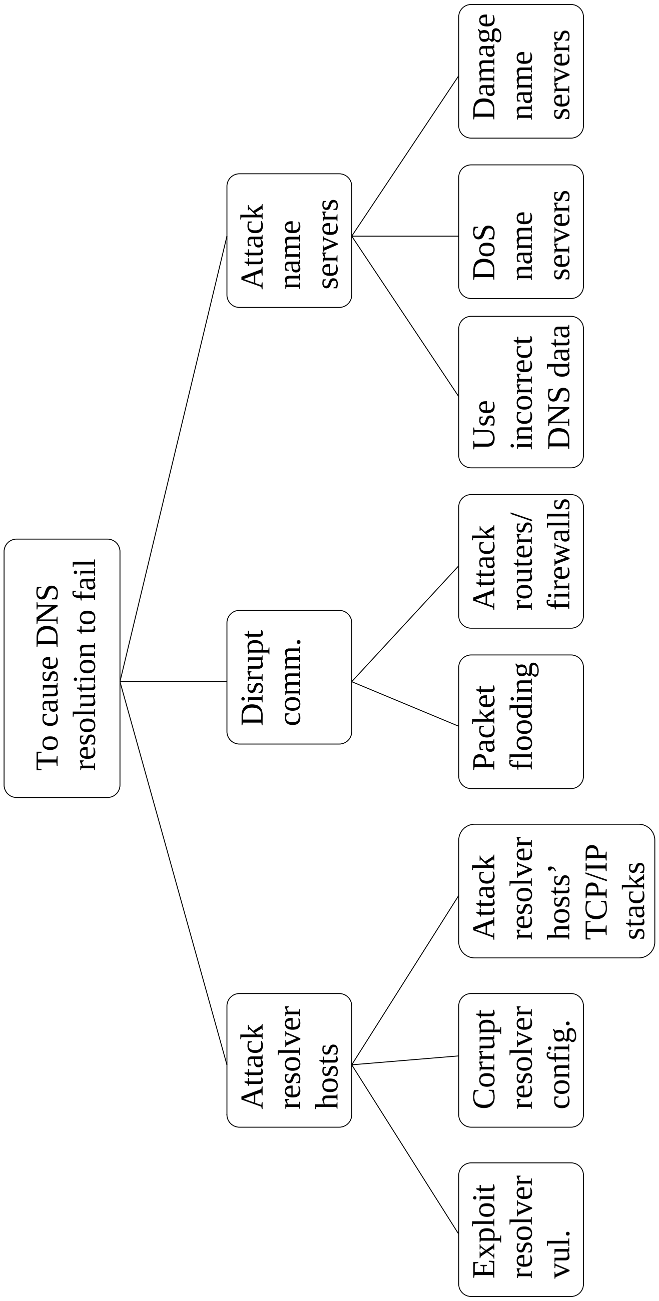

Figure 2.1: Attack Tree: DoS against DNS

name (i.e., non-fully qualified domain name). Thus, an adversary may cause name resolution involving an incomplete domain name to fail by appropriately modifying the resolver configuration settings. Another example is that DNS name resolution may be disabled or having the resolver host consult a local host table instead by changing a directive in a resolver configuration file.

An adversary may also attack the TCP/IP stack of a resolver host to confuse it about the identity or the availability of the first-hop name servers. For example, an attack may send ICMP unreachable messages to the resolver host to cause it to give up sending DNS requests to name server(s). If the adversary has access to the local area network in which the resolver host lives, other attacks such as ARP spoofing [9] may be used to change the mapping of IP addresses of the name servers to incorrect MAC addresses.

2.2 Disrupt Communication

Goal: Prevent DNS resolvers/servers from receiving queries or responses (OR)

1. Perform packet flooding to cause network congestion, which causes routers to drop DNS queries (OR)

   1.1 Direct packet flooding attacks

   1.2 Indirect packet flooding attacks

2. Exploit the vulnerabilities of routing protocols or firewalls

A brute force approach for attacking the availability of name servers is to flood the servers with packets. The 2002 attack against the root servers described in Chapter 1 is an example of this type. If the network routers at the server sites or their upstream ISP cannot handle the network traffic, the routers will have to drop certain packets. Because it may be impossible to distinguish legitimate network traffic from attack traffic, both types of traffic going to the name servers may be dropped during the attack. Depending on the number and network bandwidth of the attacking machines, a DoS attack like that may prevent the majority of the normal DNS queries from reaching the name servers. An adversary may leverage on other hosts on the Internet to increase the attack power by using those hosts as "bandwidth amplifiers" and directing the resultant network traffic to the name servers. For example, the *smurf* attack [13] sends ICMP echo request messages (with a forged source address) to broadcast addresses to generate many ICMP echo reply messages directed at the target, or an adversary takes advantage of the difference in size between DNS queries and DNS responses and sends forged DNS queries to other name servers to flood the target name servers with large DNS responses [7].

DNS queries may fail to reach the name servers if the underlying routing infrastructure is compromised. For example, a compromised router may selectively drop all DNS queries to root name servers. An in-depth threat analysis against routers and routing protocols is beyond the scope of this paper. Readers are referred to [14,19,28] for more information about routing protocol vulnerabilities and to [15] for an attack tree for the Border Gateway Protocol (BGP).

2.3 Attack Name Servers

Goal: Attack DNS servers to prevent them from resolving DNS queries (OR)

1. Use incorrect DNS data for name resolution

2. DoS DNS servers

3. Damage name servers

At the server end, one may cause the resolution process to fail by having the server use incorrect DNS data, preventing it from receiving queries, or disabling the server.

2.3.1 Use Incorrect DNS Data for Name Resolution

1. Corrupt the cache of DNS servers so that it uses bogus data for name resolution

2. Attack zone transfer between primary and secondary master name servers (OR)

   2.1 Prevent TCP connection establishment between primary and secondary master name servers

   2.2 Hijack TCP connections and feed incorrect DNS data to secondary master name servers

   2.3 Modify DNS data in transit between primary and secondary master name servers

3. Unauthorized changes of DNS data

4. Human errors or insider threats

When a name server performs recursive name resolution, it may cache the results to expedite processing for future queries.[3] If an adversary can poison the cache of a server with bogus resource records or negative responses, it may cause name resolution to fail. For example, if an adversary can corrupt the cache of a name server with incorrect NS/A resource records of the name servers for the `example.com` zone, the attacked name server may be unable to contact the `example.com` name servers, and thus may be unable to resolve names that belong to the `example.com` domain. Cache poisoning attacks may be conducted by having a name server controlled by the adversary give out incorrect DNS data, or by spoofing DNS responses via transaction ID prediction [26]. Although these attacks have been known for years [8,31,32], they are still a threat (e.g., [21]).

For robustness, one should deploy two or more name servers that are authoritative for a zone. One is the *primary master* name server, which obtains the zone updates directly, and the others are *secondary master* (or *slave*) name servers, which obtain zone data from the primary master via a process called *zone transfer*. By preventing the zone transfer process (e.g., using a TCP SYN flooding attack against the primary master), the secondary master servers will not be able to obtain the updates and their zone data will eventually expire. The zone transfer process may also be attacked by TCP connection hijacking or by a man-in-the-middle attack, which may cause a secondary master to serve incorrect DNS data.

The zone data of an authoritative name server may be compromised by exploiting a vulnerability of the name server to gain access and then modify the stored zone data, or by performing (unauthorized) dynamic updates for zone data. Also, an adversary may exploit a weakness of the registrar/registry operation procedure or communication protocol to fraudulently change the DNS data pertaining to a domain, for example, the domain hijacking attack for the ISP Panix in January 2005 [25].

Errors made by DNS operators may affect DNS availability. For example, in July 1997, corrupted zone data for the top-level domains `.com` and `.net` was distributed, which caused Internet-wide disruption for more than 4 hours [35]. A common example of operational errors is *lame delegation* in which a zone has incorrect information about the name servers of a delegated zone. This may occur when a DNS operator changes the name servers for a zone but fails to update the corresponding resource records in the parent zone or to notify the administrator of the parent zone about the change. As a result, name resolution following the DNS hierarchy will fail to

[3] Name servers may be categorized into three types: caching-only name servers, authoritative-only name servers, and name servers that perform caching and are authoritative for a zone. The authoritative-only name servers do not support recursive name resolution and are designed to respond to queries based on the DNS resource records for which they are authoritative. Thus authoritative-only name servers are immune to cache poisoning attacks.

reach the name servers that can answer the query. Although the above examples were probably accidents, a renegade DNS administrator could have initiated them.

2.3.2 DoS DNS Servers

1. Attack the TCP/IP stack of the DNS server machines to cause them to drop incoming DNS queries
2. Exhaust the resources of DNS servers

One may be able to force name servers to drop DNS queries by attacking the TCP/IP stack of name server machines, for example, by exploiting IP fragmentation reassembly vulnerabilities to exhaust memory or CPU resources. Another approach is to exhaust the CPU and memory resources of a DNS server, for example, by bombarding name servers with a lot of DNS queries so that they do not have enough resources to process all the DNS queries they receive.

2.3.3 Damage Name Servers

1. Exploit a vulnerability of DNS servers to cause them to fail or misbehave
2. Gain access to the DNS server and reconfigure/shutdown the server
3. Compromise a service on which name servers depend
4. Physical attacks against the DNS server

Name server implementations may have vulnerabilities that can be exploited to cause them to fail or misbehave; readers are referred to CERT advisories and the BIND vulnerability list [22] for more information. Name servers may also be disabled by gaining remote or physical access (e.g., password guessing or social engineering) and turning them off.

Chapter 3

Countermeasures

Denial of service has long been a difficult problem to address. A book by Mirkovic et al. [27] presents state-of-the-art attack and defense methods for denial of service. Some general defense techniques such as packet filtering and rate limiting are useful in handling DoS attacks against DNS. This section describes DNS-specific countermeasures for DoS attacks.

3.1 Existing Countermeasures

The "best current practice" RFCs (e.g., [11, 18]) and a book by Albitz and Liu [2] provide guidance on planning and operating DNS. This section describes some of the existing methods that are useful for protecting DNS from DoS attacks.

Multiple name servers To increase the robustness of DNS, one should avoid having a single point of failure. For a zone, one should deploy more than one name server and have the servers geographically and topologically distributed to reduce the likelihood that an attack or accident can knock out all authoritative name servers of a zone. More information about deploying secondary name servers is in [18].

Anycast routing A special case of deploying multiple name servers is having them share a common IP address and using anycast routing. When a client sends a DNS query to this network address, the underlying routing infrastructure will direct the query to exactly one of these servers, typically the one that is "closest" to the client. This technique is useful in increasing the number of name servers for a zone without using more IP addresses, which is important in some cases (e.g., the root zone and the top-level domains) to keep the sizes of DNS responses small. To counter flooding attacks, using anycast routing may also have the benefit of containing the damage of the attacks to the network regions in which the attack traffic originates. See [1,20,24] to learn more about using anycast routing for DNS.

Overprovisioning of host resources and network capacity By using machines and network connections that can handle significantly more traffic than the expected peak load for the DNS servers, the additional capacity can function as a buffer zone to deal with server failure and a surge in traffic due to DoS attacks. For instance, a requirement for root servers is that each server can handle three times the peak load occurring in normal conditions so that the root services can be provided even if two-thirds of the root servers are unavailable [11].

Diversity Using diverse DNS software implementations, operating systems, hardware platforms, and personnel (e.g., the root name servers are managed by different organizations independently) may reduce the risk of having a single point of failure. For example, different resolver and name server implementations tend to have different sets of vulnerabilities.

TSIG The transaction signature (TSIG) [33] scheme uses symmetric-key cryptography and one-way hash functions to protect DNS transactions. In particular, one can use TSIG to authenticate DNS responses to ensure that data received in zone transfers is authentic and not modified in transit. Moreover, name servers can use TSIG to authenticate dynamic update requests. As discussed in Section 2.3.1, if a name server uses incorrect DNS data, it may fail to resolve DNS queries.

Dedicated machines Using separate name servers for providing the services of iterative and recursive name resolution may reduce risk. Authoritative-only name servers, which support only iterative name resolution, may not be vulnerable to cache poisoning attacks. Also, one should avoid running non-DNS applications on the name server machines; otherwise, the vulnerabilities of those applications may present additional attack vectors for an adversary to compromise the name servers.

3.2 To-be-deployed or Proposed Countermeasures

This section describes countermeasures that are not yet (widely) deployed or are in the research stage.

DNSSEC The DNS security extensions (DNSSEC) [3–5] use digital signatures to secure the authenticity and integrity of DNS data and negative responses (i.e., authenticated denial of existence). DNSSEC provides strong protection against the threats of spoofing and man-in-the-middle attacks. Although, as noted in [6], DNSSEC does not help address flooding-based attacks, it may strengthen DNS against a type of DoS attacks. Ensuring integrity and authenticity of DNS data is useful not only for countering DNS attacks that return incorrect results to applications such as *pharming*, which involves attacking DNS to redirect users to malicious Web sites even though they enter the correct URLs in their browsers, but also for addressing certain DoS attacks by preventing DNS servers from using incorrect DNS data for resolution. Ongoing efforts to facilitate the large-scale deployment of DNSSEC include the IETF DNS Extensions and DNS Operations Working Groups, and the DNSSEC Deployment Initiative [17].

Peer-to-peer DNS Using a peer-to-peer network to provide the DNS service (e.g., [16, 29]) presents another approach for addressing DoS attacks. In particular, a peer-to-peer DNS does not rely on the root servers and the top-level domain DNS servers for resolution, and may dynamically adapt to an increase in the number of DNS queries for a domain by having more nodes act as servers. Thus the peer-to-peer approach may present a more scalable solution to counter flooding attacks against DNS. Note that the peer-to-peer DNS and the existing DNS are not mutually exclusive. For example, one may add a new dimension of security to the existing DNS by deploying the peer-to-peer DNS as a backup. An experimental deployment of a peer-to-peer DNS, called CoDoNS [29], has been conducted in PlanetLab.

Chapter 4

Concluding Remarks

Although the existing and proposed countermeasures appear to be useful for addressing many of the DoS threats against DNS, conducting a comprehensive analysis on them may help us better understand their effectiveness and limitations, and derive accurate security values used in the attack tree. For example, consider the apparent lack of diversity in the population of name servers deployed on the Internet (as indicated by Internet Systems Consortium's Internet domain surveys [23]). A study may be conducted on name server implementations to understand the tradeoffs between diversity and security.

We need to take into account the risk of DoS attacks in protocol design and deployment. For example, to facilitate DNS operators in deploying DNSSEC and make zone data enumeration more difficult, a solution [36] is proposed that involves generating digital signatures on-the-fly for providing a strong cryptographic proof to show that a queried domain name does not exist. As noted in [36], there is a DoS risk for that scheme because digital signature generation is a computationally intensive operation. An adversary may be able to attack a name server by flooding it with DNS queries corresponding to nonexistent domain names to cause denial of service. Based on a cost-benefit analysis for the deploying organization, precautionary measures (e.g., using faster computers or a hardware-based solution for signature generation) may be employed to mitigate the threat.

Acknowledgements

We thank Steve Crocker, Scott Rose, and Marcus Sachs for their suggestions and comments on a draft of this paper.

Bibliography

[1] J. Abley. Hierarchical anycast for global service distribution. ISC Technical Note 2003-1, 2003.

[2] P. Albitz and C. Liu. *DNS and BIND*. O'Reilly & Associates, 2001.

[3] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. DNS security introduction and requirements. RFC 4033, Mar. 2005.

[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol modifications for the DNS security extensions. RFC 4035, Mar. 2005.

[5] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Resource records for DNS security extensions. RFC 4034, Mar. 2005.

[6] D. Atkins and R. Austein. Threat analysis of the Domain Name System (DNS). RFC 3833, Aug. 2004.

[7] Australian Computer Emergency Response Team. *Denial of Service (DoS) attacks using the Domain Name System (DNS)*, Aug. 13, 1999. AusCERT Advisory AL-1999.004.

[8] S. M. Bellovin. Using the Domain Name System for system break-ins. In *Proceedings of the 5th USENIX UNIX Security Symposium*, pages 199–208, Salt Lake City, Utah, June 5–7, 1995. USENIX.

[9] S. M. Bellovin. A look back at "security problems in the TCP/IP protocol suite". In *Proceedings of the 20th Annual Computer Security Applications Conference*, pages 229–249, Tucson, Arizona, Dec. 6–10, 2004.

[10] T. Bridis. Powerful attack cripples majority of key Internet computers. The Associated Press, Oct. 22, 2002. http://www.securityfocus.com/news/1400.

[11] R. Bush, D. Karrenberg, M. Kosters, and R. Plzak. Root name server operational requirements. BCP 40, RFC 2870, June 2000.

[12] CAIDA. Nameserver DoS attack October 2002. http://www.caida.org/projects/dns-analysis/oct02dos.xml.

[13] CERT Coordination Center. *Smurf IP Denial-of-Service Attacks*, Jan. 5, 1998. CERT Advisory CA-1998-01.

[14] S. Cheung and K. N. Levitt. Protecting routing infrastructures from denial of service using cooperative intrusion detection. In *Proceedings of the New Security Paradigms Workshop*, pages 94–106, Cumbria, UK, Sept. 23–26, 1997.

[15] S. Convery, D. Cook, and M. Franz. *An Attack Tree for the Border Gateway Protocol*. Cisco, Feb. 26, 2004. Work in progress, IETF Internet-Draft draft-ietf-rpsec-bgpattack-00.txt.

[16] R. Cox, A. Muthitacharoen, and R. T. Morris. Serving DNS using a peer-to-peer lookup service. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, Cambridge, MA, Mar. 2002.

[17] DNSSEC Deployment Initiative. http://www.dnssec-deployment.org.

[18] R. Elz, R. Bush, S. Bradner, and M. Patton. Selection and operation of secondary DNS servers. BCP 16, RFC 2182, July 1997.

[19] G. G. Finn. Reducing the vulnerability of dynamic computer networks. Technical Report ISI/RR-88-201, University of Southern California, June 1988.

[20] T. Hardie. Distributing authoritative name servers via shared unicast addresses. RFC 3258, Apr. 2002.

[21] Internet Storm Center, SANS. March 2005 DNS poisoning summary. http://isc.sans.org/presentations/dnspoisoning.php.

[22] Internet Systems Consortium. BIND vulnerabilities. http://www.isc.org/index.pl?/sw/bind/bind-security.php.

[23] Internet Systems Consortium. Internet domain survey. http://isc.org/ds.

[24] D. Karrenberg. Distributing K-root service by anycast routing of 193.0.14.129. Document ripe-268, Feb. 10, 2003.

[25] J. Leyden. Panix recovers from domain hijack, Jan. 17, 2005.

[26] LURHQ Threat Intelligence Group. DNS cache poisoning - the next generation, Jan. 27, 2003. http://www.lurhq.com/cachepoisoning.html.

[27] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher. *Internet Denial of Service: Attack and Defense Mechanisms*. Prentice Hall, 2005.

[28] R. Perlman. *Network Layer Protocols with Byzantine Robustness*. PhD thesis, MIT, Aug. 1988.

[29] V. Ramasubramanian and E. G. Sirer. The design and implementation of a next generation name service for the Internet. In *Proceedings of the ACM SIGCOMM*, pages 331–342, Portland, Oregon, Aug. 30–Sept. 3, 2004.

[30] B. Schneier. Attack trees: Modeling security threats. *Dr Dobb's Journal*, 24(12):21–29, Dec. 1999.

[31] C. L. Schuba and E. H. Spafford. Addressing weaknesses in the Domain Name System protocol. Technical report, Purdue University, 1994.

[32] P. Vixie. DNS and BIND security issues. In *Proceedings of the 5th USENIX UNIX Security Symposium*, pages 209–216, Salt Lake City, Utah, June 5–7, 1995. USENIX.

[33] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington. Secret key transaction authentication for DNS. RFC 2845, May 2000.

[34] P. Vixie, G. Sneeringer, and M. Schleifer. Events of 21-Oct-2002. ISC/UMD/Cogent, Nov. 24, 2002.

[35] P. Wayner. Human error cripples the Internet. The New York Times, July 17, 1997. http://www.nytimes.com/library/cyber/week/071797dns.html/.

[36] S. Weiler and J. Ihren. *Minimally Covering NSEC Records and DNSSEC Online Signing*, May 2005. Work in progress, IETF Internet Draft draft-ietf-dnsext-dnssec-online-signing-00.