Perils of Transitive Trust in the Domain Name System

Venugopalan Ramasubramanian and Emin Gün Sirer
Dept. of Computer Science, Cornell University, Ithaca, NY 14853
{*ramasv, egs*}*@cs.cornell.edu*

Abstract

The Domain Name System, DNS, is based on nameserver delegations, which introduce complex and subtle dependencies between names and nameservers. In this paper, we present results from a large scale survey of DNS, and show that these dependencies lead to a highly insecure naming system. We report specifically on three aspects of DNS security: the properties of the DNS trusted computing base, the extent and impact of existing vulnerabilities in the DNS infrastructure, and the ease with which attacks against DNS can be launched. The survey shows that a typical name depends on 46 servers on average, whose compromise can lead to domain hijacks, while names belonging to some countries depend on a few hundred servers. An attacker exploiting well-documented vulnerabilities in DNS nameservers can hijack more than 30% of the names appearing in the Yahoo and DMOZ.org directories. And certain nameservers, especially in educational institutions, control as much as 10% of the namespace.

1 Introduction

The Domain Name System (DNS), which resolves host names to IP addresses, is critical to the integrity of Internet services and applications. Yet, the design of DNS poses security risks that are difficult to anticipate and control. DNS relies on a delegation based architecture, where resolution of a name to its IP address requires resolving the names of the servers responsible for that name. Resolving these server names, in turn, depends on additional name resolutions, creating complex interdependencies among DNS servers. Overall, the resolution of a single name is directly or indirectly affected by several servers, and compromise of any of them can severely affect the integrity of DNS and the applications that rely on it.

This paper studies the risks posed by the delegation based architecture for DNS name resolution. Our study, based on a large-scale survey of half a million domain names, answers some of the basic questions about DNS security: How many servers are involved in the resolution of a typical domain name? How easy is it to hijack domains by exploiting well known security holes in DNS servers? Which servers control the largest number of domain names, and how vulnerable are they?

Our survey exposes several new and surprising vulnerabilities in DNS. First, we find that the resolution of a domain name depends on a large trusted computing base of 46 servers on average, not including the root servers. Of this, only 2.2 servers on average are directly designated by the nameowner; the remainder is outside the control of the nameowner. Second, 30% of domain names can be hijacked by compromising just two servers each, where both servers contain publicly-known security loopholes. Finally, about 125 servers control a disproportionate 10% of the namespace. Surprisingly, 25 of these critical servers are operated by educational institutions, which may not have adequate incentives and resources to enforce integrity.

Overall, this study shows that DNS has complex dependencies, where a vulnerability in an obscure DNS server may have far reaching consequences. For example, the domain *fbi.gov* indirectly depends on a server belonging to *telemail.net*, which is vulnerable to four well-known exploits. A malicious agent can easily compromise that server, use it to hijack additional domains, and ultimately take control of FBI's namespace.[1]

The primary contribution of this paper is to expose the inherent risks involved in a basic Internet service. These risks create an artificial dilemma between failure resilience, which argues for more geographically distributed nameservers, and security, which argues for fewer centralized trusted nodes. Our study indicates that many network administrators may not be aware of this artificial tradeoff caused by the current design of DNS, and thus make an uninformed choice between failure resilience and security.

The rest of the paper is organized as follows. The next section provides some background on the delegation based architecture of DNS. Section 3 presents the findings of our survey, and Section 4 summarizes other related DNS surveys. Finally, Section 5 discusses the impact of our findings and concludes.

2 DNS Overview and Threats

DNS namespace is hierarchically partitioned into non-overlapping regions called *domains*. For example, *cs.cornell.edu* is a sub-domain of *cornell.edu*, which in turn is a sub-domain of the top-level domain *edu*, which is under the global root domain. Names within a domain

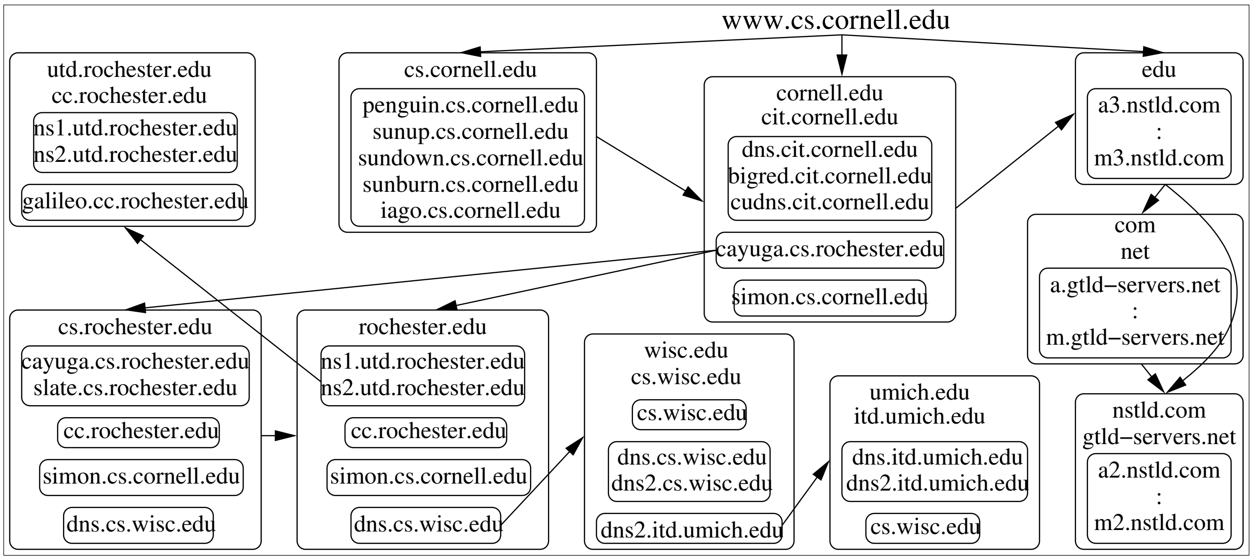

Figure 1: **Delegation Graph: DNS exhibits complex inter-dependencies among nameservers due to its delegation based architecture. For example, the domain name *www.cs.cornell.edu* depends indirectly on a nameserver in *umich.edu*. Arrows in the figure indicate dependencies. Self-loops and redundant dependencies have been omitted for clarity.**

are served by a set of nodes called the *authoritative nameservers* for that domain. At the top of the DNS hierarchy are *root nameservers* and the authoritative nameservers for *top-level domains* (TLDs). The top-level domain namespace consists of generic TLDs (gTLD), such as *.com*, *.edu*, and *.net*, and country-code TLDs (ccTLD), such as *.uk*, *.tr*, and *.in*.

DNS uses a delegation based architecture for name resolution [6, 7]. Clients resolve names by following a chain of authoritative nameservers, starting from the root, followed by the TLD nameservers, down to the nameservers of the queried name. For example, the name *www.cs.cornell.edu* is resolved by following the authoritative namesevers of the parent domains *edu*, *cornell.edu*, and *cs.cornell.edu*. Following the chain of delegations requires additional name resolutions to be performed in order to obtain the addresses of intermediate nameservers.[2] Each additional name resolution, in turn, depends on a chain of delegations. Overall, these delegations induce complex non-obvious dependencies among nameservers, and can cause unexpected nodes to exert great control over remote domains.

A domain name is said to *depend* on a nameserver if the nameserver could be involved in the resolution of that name. Similarly, a nameserver is said to *affect* a name if the name can involve that nameserver in its resolution. We represent the dependencies among nameservers that directly or indirectly affect a domain name as a *delegation graph*. The delegation graph consists of the transitive closure of all nameservers involved in the resolution of a given name. The nameservers in the delegation graph of a domain name form the *trusted computing base* (TCB) of that name.

Figure 1 illustrates the delegation interdependencies for the name *www.cs.cornell.edu*. In addition to the top-level domain nameservers, the resolution of this name depends on twenty other nameservers, of which only nine belong to the *cornell.edu* domain. Several nameservers that are outside the administrative domain of Cornell have indirect control over Cornell's namespace. In this case, *cornell.edu* depends on *rochester.edu*, which depends on *wisc.edu*, which in turn depends on *umich.edu*. While Cornell directly trusts *cayuga.cs.rochester.edu* to serve its namespace, it has no control over the nameservers that *rochester.edu* trusts.

Compromise of any nameserver in the delegation graph of a name can lead to a hijack of that name. The compromised nameserver can divert DNS requests to malicious nameservers, which effects the hijack by providing false IP addresses; clients can thus be misdirected to servers controlled by attackers and become easy victims of phishing attacks. Surely, it is not the case that all of the nameservers in the delegation graph are involved in every resolution of the name. We distinguish between a *partial hijack*, where an attacker compromises a few nameservers and diverts some queries for the targeted name, and a *complete hijack*, where an attacker compromises enough nameservers to guarantee the misdirection of all the queries for the name.

Attackers can use a combination of techniques, including systematic break-ins and denial of service attacks, to disrupt nameservice. Commonly used nameserver softwares, such as BIND, have well-documented security loopholes, which can be exploited using standard crack tools to break into the vulnerable nameservers [3]. Targeted denial of service attacks through link saturation and overloading on some nameservers further exacerbates the impact of the break-ins by increasing the number of requests passing through the exploited nameservers. Overall, the dele-

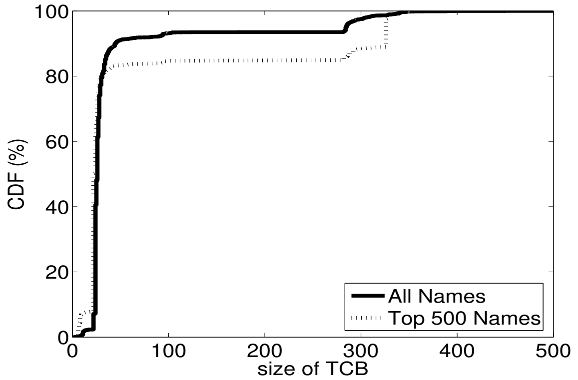

Figure 2: **Size of TCB: DNS Name resolution depends on a large number of nameservers. On average, name resolution involves 46 nameservers, while a sizable fraction of names depend on more than 100 nameservers.**

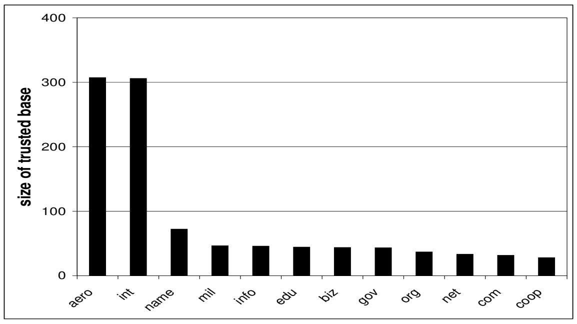

Figure 3: **Average TCB Size for gTLD Names: Names in *.aero* and *.int* have significantly larger TCBs.**

gation graph facilitates attackers to carefully select targets that maximize the impact of attacks and to take over large portions of the namespace.

3 Survey Results

We performed a large-scale survey to understand the risks posed by DNS delegations. We collected 593160 unique webserver names by crawling the Yahoo! and DMOZ.org directories. These names are distributed among 196 distinct top-level domains. Since the names were extracted from Web directories, these names are representative of the sites people actually care about. We then queried DNS for these names and recorded the chain of nameservers that were involved in their resolution. Totally, 166771 nameservers were discovered in this process. We thus obtained a snapshot of the DNS dependencies as it existed on July 22, 2004.

We study three different aspects of the dependencies to quantify the security risks in DNS. First, we examine the size of the trusted computing base for each name to determine which names are most vulnerable. Second, we study how software loopholes in DNS servers can be exploited to hijack domain names. Finally, we determine the most valuable nameservers, which affect large portions of the namespace, and explore how securely they are managed.

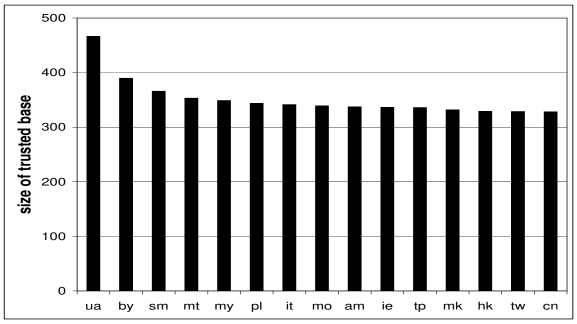

Figure 4: **Average TCB Size for ccTLD Names: Some ccTLDs rely on, and are vulnerable to compromises in, a large number of servers.**

3.1 Most Vulnerable Names

The vulnerability of a DNS name is tied to the number of servers in its trusted computing base, whose compromise could potentially misdirect clients seeking to contact that server. Larger TCBs provide attackers with a wider choice of targets to attack. Further, larger TCBs also imply more complex and deeper dependencies among nameservers making it more difficult for the nameowner to control the integrity of the servers it depends on. In this section, we characterize the TCB size of the surveyed names.

Figure 2 plots the cumulative distribution of TCB sizes not including the root nameservers, which belong to the TCBs of all the domain names. Our survey shows that TCB size follows a heavy-tailed distribution with a median of 26 nameservers, and an average of 46 nameservers; about 6.5% of the names has a TCB of greater than 200 nameservers. We computed the TCB by counting the number of distinct server names in the delegation graph. Since distinct names referring to the same machine may cause the TCB to appear larger, we also computed the number of distinct IP addresses in the delegation graphs. TCB size based on IP addresses has the same median (26), while the average decreases marginally to 44.

One might expect that the administrators of the popular websites would be better aware of the security risks and keep their DNS dependencies small. To test this hypothesis, we separately studied the TCB sizes for the 500 most popular websites reported by *alexa.org*. Figure 2 shows that these names are more vulnerable; they depend on 69 nameservers on average, and 15% of them depend on more than 200 nameservers.

Next, we study the TCB sizes for names belonging to different TLDs. Figures 3 and 4 plot in decreasing order the TCB sizes for names in the generic TLDs, and the fifteen most vulnerable country-code TLDs, respectively. Overall, ccTLD names have a much higher average TCB size of 209 nameservers than gTLD names, whose average is 87 nameservers. GTLDs *aero* and *int* have considerably larger TCBs than other gTLDs, and among the ccTLDs Ukraine, Belarus, San Marino, Malta, Malaysia, Poland and Italy, in that order, are the most vulnerable.

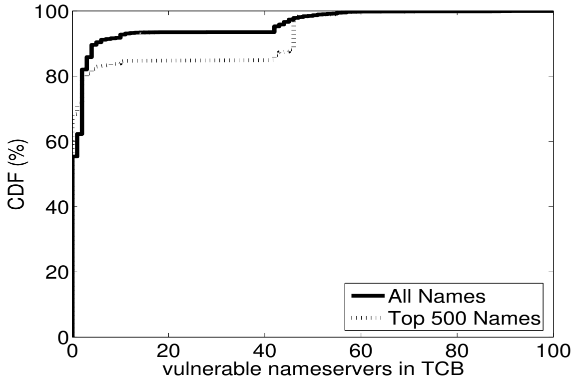

Figure 5: **Vulnerable Nameservers in TCB: 45% of the names depend on at least one nameserverver with known vulnerability.**

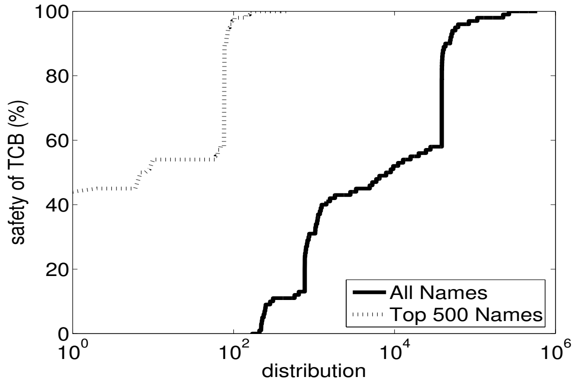

Figure 6: **Percentage of Non-Vulnerable Nodes in TCB: A few names have their entire TCB vulnerable to known exploits.**

We examined the dependencies to determine why certain domain names (e.g., names in *aero* and *int*) have much larger TCBs than others. We find that names with larger TCBs typically have authoritative nameservers distributed across distant domains. Improving availability in the presence of network outages is one of the primary reasons why administrators delegate to, and implicitly trust, nameservers outside their control. Extending trust to a small number of nameservers that are geographically distributed may provide high resilience against failures. However, DNS forces them to trust the entire transitive closure of the all names that appear in the physical delegation chains.

Sometimes even top-level domains are set up such that it is impossible to own a name in that subdomain and not depend on hundreds of nameservers. Ukrainian names seem to suffer from many such dependencies. The most vulnerable name in our survey, *www.rkc.lviv.ua*, depends on nameservers in the US including Berkeley, NYU, UCLA, as well as many locations spanning the globe: Russia, Poland, Sweden, Norway, Germany, Austria, France, England, Canada, Israel, and Australia.[3] It is likely that the Ukrainian authorities do not realize their dependency on servers outside their control. An attacker that controls a nameserver at Monash University in Australia can end up hijacking the website of Ukrainian government. DNS creates a small world after all!

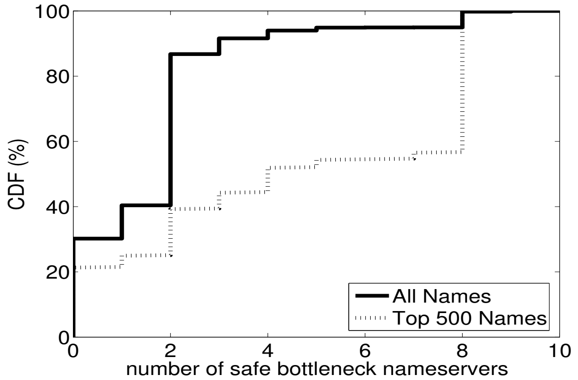

Figure 7: **DNS Nameserver Bottlenecks: 30% percentage of names can be completely hijacked by compromising a critical set of vulnerable bottleneck nameservers.**

3.2 Impact of Known Exploits

As part of our survey, we also collected version information for nameservers using BIND, the most widely-used DNS server, where possible. Different versions of BIND contain well-documented software bugs [3]. We combine known vulnerabilities with the delegation graphs of domain names to explore which names are easily subjected to compromise. For nameservers whose vulnerabilities we do not know, we simply assume that they are *non-vulnerable*; hence, the results presented here are optimistic.

Of the 166771 nameservers we surveyed, 27141 have known vulnerabilities. A naive expectation might be that, with 17% vulnerable nameservers, only 17% of the names would be affected. Instead, these vulnerabilities affect 264599 names, approximately 45%, because transitive trust relationships "poison" every path that passes through an insecure nameserver.

For example, *www.fbi.gov* is vulnerable to being hijacked, along with all other names in the *fbi.gov* domain. The *fbi.gov* domain is served by two machines named *dns.sprintip.com* and *dns2.sprintip.com*. The *sprintip.com* domain is in turn served by three machines named *reston-ns[123].telemail.net*. Of these machines, *reston-ns2.telemail.net* is running an old nameserver (BIND 8.2.4), with four different known exploits against it (libbind, negcache, sigrec, and DoS_multi) [3]. Having compromised *reston-ns2*, an attacker can divert a query for *dns.sprintip.com* to a malicious nameserver, which can then divert queries for *www.fbi.gov* to any other address, hijacking the FBI's website and services.

Figure 5 shows the cumulative distribution of the number of vulnerable nameservers in the TCBs of surveyed names. 45% of DNS names depend on at least one vulnerable nameserver, and can be compromised by launching well-known, scripted attacks. Figure 6 shows the percentage of nodes with no known bugs in the TCBs of surveyed names. Surprisingly, a few names do not have any non-vulnerable nameservers in their TCB; these names belong to the ccTLD *ws*, which relies on older buggy versions of

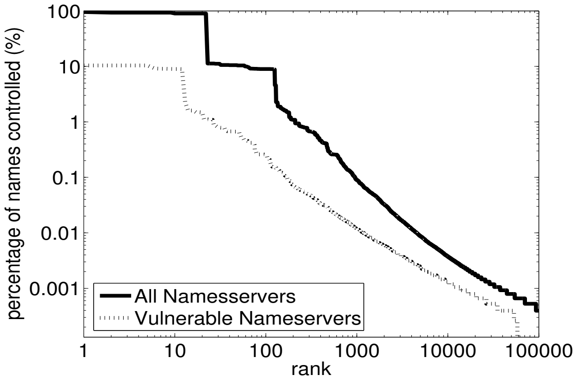

Figure 8: **Percentage of Names Controlled by Nameservers: Some nameservers with known vulnerabilities affect a large percentage of names.**

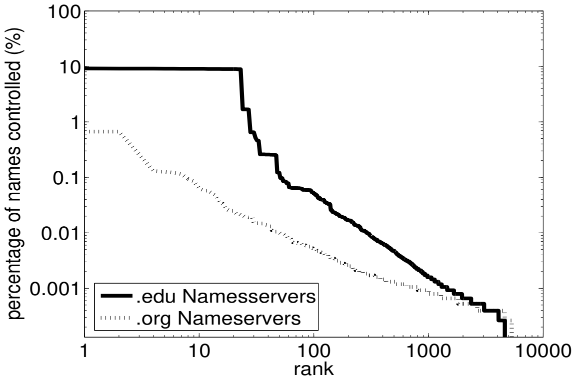

Figure 9: **Percentage of Names Controlled by Nameservers in *.edu* and *.org* Domains: Some nameservers in educational institutions and non-profit organizations affect large percentage of names.**

BIND. Overall, the average number of vulnerable servers is 4.1, about 9% of the average TCB size. The extent of vulnerability in the TCBs of the 500 most popular names is also high (7.6), about 11% of the average TCB size.

We examined the chances of a complete domain hijack by counting the minimum number of nameservers that need to be attacked in order to completely take over a domain. Such critical bottleneck nameservers can be determined by computing a min-cut of the delegation graph. Figure 7 shows the number of non-vulnerable nameservers in the min-cut of the delegation graphs.

Surprisingly, about 30% of domain names have a min-cut consisting entirely of vulnerable nameservers. The average size of a min-cut is 2.5 nameservers. This implies that these domain names can be completely hijacked by compromising less than three machines on average. Moreover, another 10% of domain names have only one non-vulnerable nameserver in their min-cut. A denial of service attack on the non-vulnerable nameserver, coupled with the compromise of the other vulnerable bottleneck nameservers, is sufficient to completely hijack these domains.

3.3 Most Valuable Nameservers

The value of a DNS nameserver is tied to the role it plays in name resolution. We model the value of a nameserver as being proportional to the number of domain names which depend on that nameserver. It is these high profile servers whose compromise would put the largest portions of the DNS namespace in jeopardy. Attackers are likely to focus their energies on such high-leverage servers; if the effort to break into a vulnerable nameserver is constant, then breaking into a nameserver that affects a large number of names provides a higher payoff.

Figure 8 shows the percentage of names affected by nameservers, ranked in the order of importance. It also gives a distribution of names affected by nameservers with known exploits. An average nameserver is involved in the resolution of 166 externally visible names, and the median is 4. This is the number of externally visible names that appear in well-known Web directories, and does not include automatically generated DHCP names or other DNS names that receive few, if any, lookups.

While an attacker targeting random nameservers would likely compromise only a few sites, a little bit of targeting can yield nameservers with great leverage. Figure 8 shows that about 125 nameservers each affect more than 10% of the surveyed names. Of these high profile nameservers, only about 30 are well-maintained gTLD nameservers. Several vulnerable nameservers affect large portions of the namespace; about 12 of the 125 high profile nameservers have well-known loopholes.

There are many valuable nameservers operated by institutions that may not be equipped to or willing to take on the DNS task. Figure 9 shows a distribution of names served by machines belonging to the *.edu* and *.org* domains. These nameservers are operated by entities such as universities, non-profit organizations, and so forth, whose primary business is not to provide networking services. These institutions, unlike ISPs, typically do not have a financial relationship with the owners of the names they serve, and thus lack the fiduciary incentives for providing correct, secure service that an ISP has. These institutions take on an additional risk by placing their servers at critical locations in the DNS hierarchy; they may be liable if their servers are taken over and used to hijack a DNS domain.

4 Related Work

Several surveys and measurement studies have been performed on DNS. However, they have typically focused on the performance and availability of DNS.

In 1988, Mockapetris and Dunlap published a retrospective study on the development of DNS identifying its successful features and shortcomings [8]. Several measurement studies since then have provided insights into the performance of the system. A detailed study of the effectiveness of caching on lookup performance is presented by Jung et al. in [4, 5]. Park et al. [10] explore the different causes for performance delays seen by DNS clients.

Huitema and Weerahandi [2] and Wills and Shang [14] study the impact of DNS delays on Web downloads. The impact of server selection on DNS delays is measured by Shaikh et al. [12].

Two recent surveys by Pappas et al. [9] and Ramasubramanian and Sirer [11] focus on availability limitations of DNS stemming from its hierarchical structure. These studies show that most domain names are served by a small number of nameservers, whose failure or compromise prevents resolution of the names they affect.

This paper studies a fundamentally different, yet crucial, aspect of DNS design: the security vulnerabilities that stem from the delegation based architecture of DNS. It exposes the risks posed by non-obvious dependencies among DNS servers, and highlights the tradeoff between availability and security.

5 Discussion and Summary

DNS is a complex system, where a vulnerability in an obscure nameserver can have far-reaching consequences, and trust relationships are hard to specify and bound. Even if the name owners are diligent and check the extent of dependencies at the time of name creation, trust relationships can change undetected.

The main culprit here is the reliance on transitive trust [13]. Nameserver delegations induce a dependency graph, and concerns, including failure resilience and independent administration, enable the resulting dependency graphs to grow large and change dynamically. It is a well-accepted axiom of computer security that a small trusted computing base is highly desirable, since smaller TCBs are easier to secure, audit and manage. Our survey finds that the TCB in DNS is large and can include more than 400 nodes. An average name depends on 46 nameservers, while the average in some top-level domains exceeds 200.

This study shows that one in three Internet names can be hijacked using publicly-known exploits. This points to the Domain Name System as a significant common vulnerability. It is highly unlikely that an attacker can break into a third of the webservers around the globe; firewalls, hardened kernels, and intrusion detection tools deter direct attacks on webservers. But DNS enables attackers to hijack one in three sites, thus gaining the ability to masquerade as the original site, obtain access to their clients, potentially collect passwords, and possibly spread misinformation. High-profile domains, including those belonging to the FBI and many popular sites, are vulnerable because of problems stemming from the way DNS uses delegations.

A better approach is required to achieve name security on the Internet. Deployment of DNSSEC [1] can help, but DNSSEC continues to rely on the same physical delegation chains as DNS during lookups. Complex dependencies in name resolution means that a much wider acceptance of DNSSEC is required for it to be effective, since every path in the delegation graph needs to be secured. And even if all nameservers support DNSSEC, attackers can exploit vulnerabilities outlined in this paper to launch DoS attacks on Web services and disrupt name resolution. As a stopgap measure, network administrators have to be aware of the vulnerabilities in DNS and be more diligent about where they place their trust.

Notes

[1]We reported this vulnerability to the Department of Homeland Security and the servers have since been upgraded; we do not know if the vulnerability has been fixed.

[2]While DNS uses glue records, which provide cached IP addresses for nameservers, as an optimization, glue records are not authoritative.

[3]A complete list of nameservers this name depends on can be found in http://www.cs.cornell.edu/people/egs/beehive/dnssurvey.html. We maintain an active website listing the results of the survey presented here.

References

[1] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose. Protocol Modifications for the Domain Name System Security Extensions. *Request for Comments 4035*, Mar. 2005.

[2] C. Huitema and S. Weerahandi. Internet Measurements: The Rising Tide and the DNS Snag. In *Proc. of ITC Specialist Seminar on Internet Traffic Measurement and Modeling*, Monterey, CA, 2000.

[3] Internet Systems Consortium. BIND Vulnerabilities. http://www.isc.org/sw/bind/bind-security.php, Feb. 2004.

[4] J. Jung, A. Berger, and H. Balakrishnan. Modeling TTL-based Internet Caches. In *Proc. of IEEE International Conference on Computer Communications*, San Francisco, CA, Mar. 2003.

[5] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and Effectiveness of Caching. In *Proc. of SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001.

[6] P. Mockapetris. Domain Names: Concepts and Facilities. *Request for Comments 1034*, Nov. 1987.

[7] P. Mockapetris. Domain Names: Implementation and Specification. *Request for Comments 1035*, Nov. 1987.

[8] P. Mockapetris and K. Dunlop. Development of the Domain Name System. In *Proc. of ACM SIGCOMM*, Stanford, CA, 1988.

[9] V. Pappas, Z. Xu, S. Lu, D. Massey, A. Terzis, and L. Zhang. Impact of Configuration Errors on DNS Robustness. In *Proc. of ACM SIGCOMM*, Portland, OR, Aug. 2004.

[10] K. Park, V. Pai, and L. Peterson. CoDNS: Improving DNS Performance and Reliability via Cooperative Lookups. In *Proc. of Symposium on Operating Systems Design and Implementation*, 2004.

[11] V. Ramasubramanian and E. G. Sirer. The Design and Implementation of a Next Generation Name Service for the Internet. In *Proc. of ACM SIGCOMM*, Portland, OR, Aug. 2004.

[12] A. Shaikh, R. Tewari, and M. Agarwal. On the Effectiveness of DNS-based Server Selection. In *Proc. of IEEE International Conference on Computer Communications*, Anchorage, AK, Apr. 2001.

[13] K. Thompson. Reflections on Trusting Trust. *Comm. of the ACM*, 27(8), Aug. 1984.

[14] C. E. Wills and H. Shang. The Contribution of DNS Lookup Costs to Web Object Retrieval. Technical Report TR-00-12, Worcester Polytechnic Institute, July 2000.