DNS Threat Analysis

Mark Santcroos*, Olaf M. Kolkman†*NLnet Labs*
www.nlnetlabs.nl

NLnet Labs document 2006-SE-01 version 1.0 May 3, 2007

*mark@nlnetlabs.nl
†olaf@nlnetlabs.nl

File information:Id: se-consult.tex 29 2007-04-20 13:16:06Z marks

Contents

CONTENTS

Executive Summary

The DNS (Domain Name System) is a critical infrastructure component of the Internet. Although invented in the early days of the Internet its design is such that it manages to be scalable to the size and the dynamics of the Internet in present days. However, the immensive growth of the Internet was not foreseen, and the scalable design did not take the abuse patterns that comes with that into account.

DNS stakeholders need to be aware of the current limitations of the protocol and corresponding implementations. The approach we take is to create an hierarchical attack tree to map DNS security threats. Based on this tree we make a full threat analysis. With the understanding of the usage of the DNS by careful monitoring and by leveraging the awareness of said threats, solutions can and should be created to preserve the DNS as a stable and critical component of the Internet.

Recent developments of DNSSEC extensions to the DNS show to be a solution to the problems surrounding data integrity of the DNS. DNSSEC allows validation of DNS data and is therefore recommended.

Part I
Introduction

In this report we provide an inventory of the threats surrounding key Domain Name System infrastructure such as top level domains. We suggest system requirements for preventing these threats and present a few tools that can assist in prevention. The approach taken is a desk study with a focus on architecture. NLnet Labs does not operate a large scale DNS service itself, therefore there is less focus on operational aspects.

For this study we have exclusively worked with information that is openly available.

This report was created on request of and sponsored by .SE, The Internet Infrastructure Foundation.

1 Server vs. Service

The DNS is ultimately designed to provide a service. Each of the individual name servers is just a small part in the full chain of the whole DNS hierarchy. Of course, some of the DNS servers play a more critical role than others, because they are higher in the hierarchy. But the definition of importance is not an absolute one as it also depends on perspective (network logically) and what part of the tree you have your interest in.

While traditionally (in the unicast era) one name server instance would be just one physical machine and one IP address. The 2nd generation name server design introduced local load balancing. Two or more machines in the same physical location and network would spread the load of the incoming queries amongst themselves. To the outside world, this was considered one machine, as the machines were all known with the same IP address. The latest generation name server design (the anycast[1] era), adds physical spread of machines again without no apparent changes to the outside world. Multiple machines spread around the globe, but with the same IP address, together service the requests. The routing protocol on the Internet (BGP4) will make the decisions to which name server instance to deliver the request.

2 Scope

In this document many recommendations are given. In practice, most of these recommendations will be a tradeoff between a number of arguments, of which many will be non-technical. In our recommendations we explicitly did not take financial considerations and consequences into account.

Part II
Attack Tree Analysis

This part describes attack tree analysis [15] as a method to analyse DNS security threats. The tree starts with three branches which identify the main categories of DNS threats. Following the structure of the tree (Figure 1), the remainder of this part is organized hierarchically.

3 Data corruption

The first main category of DNS security threats is data corruption. It is defined as all type of incidents related to the unauthorized modification of DNS data. These incidents can happen on any moment, in any part of the DNS propagation chain. For the remainder of this document we split data corruption in three sub-categories.

3.1 Repository corruption

A repository is a central place where data is stored and maintained. For DNS this is the authoritative source of the zone information. Depending on how the DNS zone is operated this can be the raw on-disk zone files or an administrative database. For the scope of this document the format is irrelevant.

3.1.1 Outdated information

We speak of outdated information when an attack has deliberately stopped or delayed the propagation of updated information in the DNS tree. Depending on the consequent usage of the (outdated) DNS information this can have far reaching security consequences.

3.1.1.1 Denial of zone-transfer DNS zones are generally serviced by multiple name servers. The DNS data served by these name servers is expected (by the standard) to be (loosely) coherent. That means that they all strive to publish the same information. Because of propagation delay there should be made no assumptions that two or more name servers have exactly the same information at a given moment in time.

In the early days of DNS there was a traditional role separation of primary vs. secondary (sometimes referred to as master vs. slave) name servers. Primary name servers contained the original and authoritative source of DNS information and distributed that with whatever available mechanism to its configured secondary name servers. The primary name server were the one listed in the MNAME[14] field in the SOA record of a domain. With the advent of additional security measures (that were needed with the growth of the Internet) and techniques like load-balancing and anycast these traditional roles disappeared. Especially because the "host" in the MNAME field does no longer need to be a

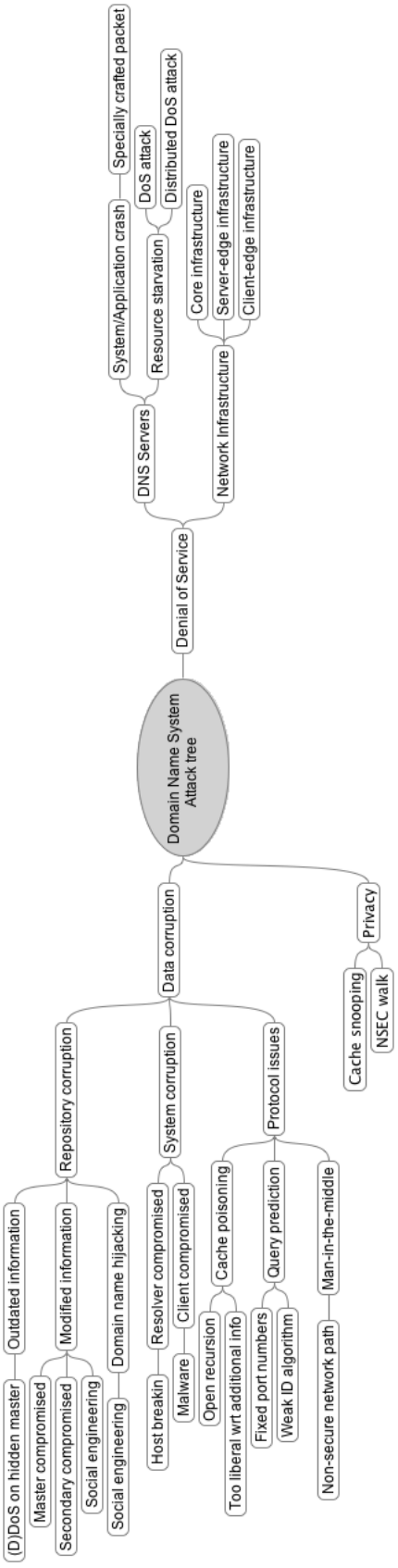

Figure 1: Attack tree

single system. These days the remaining practical use of the MNAME field is for DNS Notify[21] and DNS Dynamic Updates[23].

One of the more prominent setups where the difference between primary and secondary is relevant, is a setup with a so-called hidden primary. A hidden primary is a name server that is not listed in the zone as an authoritative name server for the domain in question. That means that the primary server will not be queried by resolvers during normal operations as it is simply not known to the outside world. Its purpose is to be a stable and reliable source of origin for the zone information. All (secondary) name servers that are listed as authoritative in the zone file will get (in this scenario) their information from the hidden primary. As per the standard, the hidden primary will be listed in the MNAME field.

An alternative scenario is the (full) mesh setup. In this situation a set of name servers authoritative for a certain domain will have an authority chain that is less dependant on a (single) primary. The name servers will be configured in such a way that they can get (and send) updates to other name servers than only the primary in the set also.

A possible incident can be if an attacker finds out the communication path between the primary and the secondary name servers he could execute a Denial of Service attack on that path and therefore disturbing the zone transfers.

Recommendation (1): ***To ensure proper replication of zones between the authoritative nameservers it is recommended to investigate Out-of-Band zone replication as an alternative synchronization method.***

Recommendation (2): ***In the case that propagation incidents do occur, it is important that the impact is carefully considered beforehand. Issues to think of are for example the tuning of SOA expiration parameters.***

3.1.2 Modified information

All unauthorized modifications to the DNS repository are described in the following sections.

3.1.2.1 Authoritative nameserver compromised

The (hidden) primary nameserver is considered the authoritative source of the information in DNS. Secondary nameservers will retrieve their information from the primary nameserver, so if the primary gets compromised this means the information on the secondaries can no longer be trusted either.

However, the secondary nameserver of a certain domain might be operated with by an organization other than the owner of the domain. This does not imply any qualitative statement about the security, but it does show the possibly dependencies on external organizations for the security of the DNS.

The secondary nameserver is vulnerable to the same types of attacks as the primary nameserver. While difficult to quantify, it might be the situation that secondary nameservers have lower profile, and therefore less administrator attention.

Possible attacks could be by exploiting bugs in the software needed for the DNS service, but can also be in other (Operating System) software running on the DNS server.

Recommendation (3): ***Secure and harden the machine(s) hosting the name server(s).***

Recommendation (4): ***Establish clear Service Level Agreements and Operating Level Agreements with the entity operating the secondary name servers.***

3.1.2.2 Social engineering

Social engineering is a method where the intruder deceives his target into complying with a request based on false pretenses and psychological manipulation. Although this attack can not be described formally, most incidents share some characteristics. These kind of attacks (well known since the black-hat days of Kevin Mitnick[13]), are generally based on a false sense of (social) trust between the victim and the attacker. Quid pro Quo, or something for something, is a variant of this situation when the attacker has something the victim might need, which makes the case stronger (and easier) for the attacker.

Recommendation (5): ***Establish clear and secure administrative procedures and educate people involved about the threats to information security.***

3.1.3 Domain Hijacking

Domain hijacking refers to the wrongful taking of control of a domain name from the rightful name holder during the registration process. The common use of the term encompasses a number of attacks and incidents including:

- impersonation of a domain name registrant in correspondence with a domain name registrar
- forgery of a registrants account information maintained by a registrar
- forgery of a transfer authorization communication from a registrant to a registrar
- impersonation or a fraudulent act that leads to the unauthorized transfer of a domain from a rightful name holder to another party
- unauthorized DNS configuration changes that disrupt or damage services operated under a domain name, including web site defacement, mail service disruption, pharming and phishing attacks

The problem of Domain Name Hijacking is technically outside the DNS scope. But because of the strong impact it has on the DNS it is covered here regardlessly. A report from ICANN [16] extensively documents threats, accidents and counter measures for all involved parties.

3.1.3.1 Typosquatting

Typosquatting is also referred to as URL hijacking. It is an attack that relies on user mistakes such as typographical errors made by Internet users when entering a website address into a browser. Should a user accidentally enter an incorrect website address, they may be led to an alternative address owned by the attacker, without the user noticing that he is led to another website than the intended one.

3.1.3.2 International Domain Names (IDN) abuse

Internationalized domain names provides a backward-compatible way for domain names to use the full Unicode character set, and this standard is already widely supported. An attacker could register a domain name that looks just like that of a legitimate website, but in which some of the letters have been replaced by homographs in another character set[7]. This creates many opportunities for phishing and other forms of fraud. For example, the attacker could send e-mail messages purporting to come from the original site, but directing people to the fake website. The fake site could record information such as username and passwords, while passing traffic through to the real website. The user may never notice the difference, until suspicious or criminal activity happens with their accounts.

3.2 System corruption

The authenticity of DNS responses is fully dependent on the trust of the whole chain of systems in the (relevant part of the) DNS tree. Generally (and by design) not all of these systems in the chain are under control of the same entity. This makes it difficult (impossible) for the owner of the DNS data to fully ensure data authenticity to the client.

3.2.1 Caching recursive name server compromised

Between the client doing the DNS request and the authoritative DNS server can be any number of (caching) recursive/forwarding DNS resolvers assisting in answering the query for the client. These DNS servers are vulnerable to the same risk as all systems on the Internet.

3.2.2 Client compromised

The last part in the DNS hierarchy is the user/client. The attacks that threaten the security of clients include computer viruses, worms, trojan horses, spyware, adware. While this part will never be under the responsibility of the owner of the domain name, it does need to be recognized that there is an ever present weakness.

3.3 Protocol issues

This category of attacks deals with incidents further down the tree (viewed from the perspective of the authoritative servers).

3.3.1 Cache poisoning

DNS cache poisoning is term that refers to abuse of deficiencies in the DNS protocol and implementations.

3.3.1.1 Open recursion While most of the DNS servers on the Internet have recursion enabled, actually few of them really need it.

Recursion on it self is not a security issue perse, but it makes the other described vulnerabilities more easy to exploit. Therefore it should be discouraged.

Recursion is exploited in source-spoofed DDoS attacks. Combined with implementation bugs, it allows outsiders to attack/crash a nameserver. Recursion makes name servers more susceptible to cache poisoning in the overall.

Recommendation (6): ***Use a fully separated name server setup. Preferably a name server is either only authoritative or it does caching, but not both.***

Recommendation (7): ***In cases that it does need to do both, it is recommended to have a split setup with an internal and an external view. It should then be configured to behave differently depending on the origin of the request.***

Recommendation (8): ***A last resort solution is to add ACLs (Access Control List) to your name server or firewall configuration to limit recursion from the outside.***

3.3.1.2 Additional info acceptance The DNS protocol provides a way for servers to add additional information to a response that it might find relevant as an addition to the authoritative answers. Because this additional info strictly does not have to be relevant in any way, it opens a door for a name server to inject information in a resolvers cache that it is not authoritative for. The resolver should therefore be sensible in what it accepts as trusted additional information

Recommendation (9): ***Use a modern DNS name server implementation that minimizes the risk of cache poisoning by being strict in what it accepts from name servers.***

3.3.2 Query prediction

Because DNS uses the connectionless UDP for its transport, DNS packets can be forged by an attacker. Whether the victim accepts the forged packet (believes it is a good packet) depends on a number of parameters:

- The question section of the (forged) reply packet matches one of the question packets been sent
- The ID field of the (forged) reply packet matches that of the question packet
- The (forged) reply packet is sent to the same network address and port number as the query was sent from

- The (forged) reply packet comes from the same network address the query was sent to

Theoretically it is possible to make correct guesses and create a "legitimate" DNS reply. However, the chance of such a situation can be heavily influenced by the way the previous four parameters are applied without the original query.

- Prevent having multiple equivalent questions outstanding
- Use the full 16 bits of the ID field
- Use a new random source port from for each outgoing query that cannot be predicted by merely having knowledge of its random number generator

Currently an Internet Draft is being written that describes the problems and the counter measures of DNS spoofing[10].

3.3.3 Man-in-the-middle

A man-in-the-middle attack is a general description for attacks that are executed when an attacker is in between two hosts (e.g. a server and a client). The attacker therefore has knowledge about the connection and can use that to eavesdrop on the connection or even inject data into it.

3.3.3.1 Non-secure network path With the wide spread usage (and further growth) of wireless network, the access to non-secure network paths is everywhere. This opens up a new dimension of man-in-the-middle attacks.

4 Denial of Service

A Denial of Service attack refers to a type of attack that renders the service unusable for legitimate users. These attacks are either aimed at a specific service (like DNS) or aimed wider to a whole part of the network (Internet).

4.1 DNS Servers

The DNS service for a domain is provided by the total set of DNS nameservers for that domain.

4.1.1 System/application crash

4.1.1.1 Specially crafted packet Nameserver software is software that openly communicates with the outside world. When there is a programming error (bug) in the software the chance is there that it can be exploited to gain more privileges or at least to crash the software. Note that there have also been incidents where specially crafted packets were even able to corrupt the network stack of the OS and therefore crash the whole machine the DNS server ran on.

Recommendation (10): ***Run diverse implementations of Operating Systems and nameserver software to spread the impact of bugs in particular implementations.***

4.1.2 Resource starvation

When a nameserver gets accidentally or intentionally too many requests, and the network is not a limiting factor, then it is possible that the server gets fully busy with receiving, and basically has no time left for answering. In case of intentional overloading the server, this will also mean that the responses it does manage to get out will not go to real users.

4.1.2.1 DoS attack A Denial of Service attack is executed from one attacking host to one victim host. The attacking host will try to consume as much resources from the victim or the infrastructure leading to the victims network, so that the service to normal users is degraded.

Because DNS servers are generally well provisioned, this kind of attack will generally not work on todays Internet because the attacker will be the bottleneck himself.

Common DoS attacks:

- TCP SYN attack put a burden on TCP socket creation on the server
- Ping of Death and teardrop attack exploits a bug in IP fragmentation reassembly
- Smurf attack is a flood attack with ICMP echo requests that congest the network
- Fraggle attack is a flood attack with UDP packets that congest the network

Recommendation (11): ***As a DoS attack is a machine-to-machine attack, the normal provisioning of high profile nameservers should be good enough to counter attacks from single hosts. (As by design it should be able to serve thousands (millions?) of clients) In case a single machine can disrupt a service, the provisioning of the name server needs to be seriously assessed.***

4.1.2.2 Distributed DoS attack Distributed DoS attacks share many characteristics with normal DoS attacks. The type of attacks are quite similar, and one can even argue that DDoS fall under the category of DoS. However, we make the distinction here that DoS attacks are executed from one host, and Distributed DoS attacks are executed from multiple 100s or 1000s of hosts. Commonly used DDoS attacks are Stacheldraht[17], TFN[18], TFN2K[19], and Trinoo[20]. Furthermore a market of even more user-friendly DDoS tools seems to be emerging.

Reflection attacks are a type of (D)DoS attack that use intermediate hosts for sending bad traffic to the victim. While the effect on the victim is the same, tracing the attacker is more difficult.

Another method that attackers use to maximize the damage are amplification attacks[4]. This is a form of reflection attack where the initiating traffic from the attacker is much less than the bad traffic from the intermediate to the victim.

DDoS attacks are very difficult to mitigate. We are not aware of any useful technique except over-provisioning and containment. Over-provisioning can be done on a node-by-node basis using local load sharing techniques.

Wide deployment of BCP38[6] and BCP84[2] is the best mitigation technique against certain DDoS attacks. TLD operators have few possibilities through enforcing the best current practices in contracts with service providers.

Anycast is a method that by design contains as well as it introduces over-provisioning.

Recommendation (12): ***Extensive system and network overprovisioning, possibly by deploying anycast***

4.2 Network infrastructure

This section describes the possibly attacks on the network infrastructure between the clients and the nameservers.

4.2.1 Core infrastructure

The core Internet infrastructure is generally not under control of the domain owner and is therefore an external dependency that is difficult to manage. It can be stated however that if the "Internet breaks down", that it can not be expected that DNS continues to work (nor would it be very useful). On the other hand if there are many "local" services, then the functioning of DNS would still be useful. This is a policy decision.

4.2.2 Server-edge infrastructure

The infrastructure at the (name) server side is of course (at least to some degree) under the control of the entity operating the name server. A document describing Root Name Server Operators Requirements [3] does into extensive detail on this matter. In general we can state that the server-edge should be well overdimensioned on all aspects.

Recommendation (13): ***It should be the goal of the name server operator to make sure that the infrastructure on the server side will not be the bottleneck in any situation, be it normal operation or during an attack.***

4.2.3 Client-edge infrastructure

It needs to be realized that, because of the nature of the Internet, and that of the DNS in particular, there is often no line-of-control between the client and the server (or the other way around).

Recommendation (14): ***The biggest win on the client side could probably be reached by education. Users need to be teached about the***

right choices concerning Internet related habits and computer security.

5 Privacy

Some problems with DNS are not so much a security issue in the sense that they change data. Instead they are privacy related issues as they allow attackers to get insight into your DNS data.

5.1 Cache snooping

DNS cache snooping [8] is the process of determining whether a given Resource Record (RR) is (or not) present on a given DNS cache. This gives information about what queries a resolver handled.

5.2 NSEC walk

DNSSEC includes the NSEC RR to provide authenticated denial of existence. Though the NSEC RR meets the requirements for authenticated denial of existence, it introduces a side-effect in that the contents of a zone can be enumerated. This property introduces undesired policy issues.

An enumerated zone can be used either directly as a source of probable e-mail addresses for spam, or indirectly as a key for multiple WHOIS queries to reveal registrant data which many registries may have legal obligations to protect. Many registries therefore prohibit copying of their zone data; however, the use of NSEC RRs renders these policies unenforceable.

A second problem is that the cost to cryptographically secure delegations to unsigned zones is high for large delegation-centric zones and zones where insecure delegations will be updated rapidly. For these zones, the costs of maintaining the NSEC record chain may be extremely high relative to the gain of cryptographically authenticating existence of unsecured zones.

The NSEC3 draft[11] presents the NSEC3 Resource Record which can be used as an alternative to NSEC to mitigate these issues.

Part III
Defense

This part breaks up the defense against attacks in three generic phases. The first phase is protection, where preventive technical measures are taken to limit the risk of attacks. The following (continuous) phase is about detection of incidents that can spawn of the next phase: (mitigating) reaction on actual attacks.

6 Protection

RFC2870 [3] describes the requirements for root name servers. Most of the requirements can also be applied to TLD operators. The requirements are split into physical, network and protocol requirements. The following section annotates the relevant parts of the RFC.

6.1 Physical

> 3.1 Physical security MUST be ensured in a manner expected of data centers critical to a major enterprise.
>
> 3.1.1 Whether or not the overall site in which a root server is located has access control, the specific area in which the root server is located MUST have positive access control, i.e. the number of individuals permitted access to the area MUST be limited, controlled, and recorded. At a minimum, control measures SHOULD be either mechanical or electronic locks. Physical security MAY be enhanced by the use of intrusion detection and motion sensors, multiple serial access points, security personnel, etc.

RFC2870 3.1.1 is both about continuity and integrity. Physical access control decreases all kind of (accidental) practical risks. Also data authenticity on the machines is better protected. Most datacenters have decent physical access control. In shared datacenters we recommend placing the machines in lockable racks.

> 3.1.2 Unless there is documentable experience that the local power grid is more reliable than the MTBF of a UPS (i.e. five to ten years), power continuity for at least 48 hours MUST be assured, whether through on-site batteries, on-site power generation, or some combination thereof. This MUST supply the server itself, as well as the infrastructure necessary to connect the server to the internet. There MUST be procedures which ensure that power fallback mechanisms and supplies are tested no less

frequently than the specifications and recommendations of the manufacturer.

3.1.3 Fire detection and/or retardation MUST be provided.

3.1.4 Provision MUST be made for rapid return to operation after a system outage. This SHOULD involve backup of systems software and configuration. But SHOULD also involve backup hardware which is pre-configured and ready to take over operation, which MAY require manual procedures.

We recommend taking close look at the MTBF of power facilities.

6.2 Network

3.2 Network security should be of the level provided for critical infrastructure of a major commercial enterprise.

3.2.1 The root servers themselves MUST NOT provide services other than root name service e.g. remote internet protocols such as http, telnet, rlogin, ftp, etc. The only login accounts permitted should be for the server administrator(s). "Root" or "privileged user" access MUST NOT be permitted except through an intermediate user account.

Servers MUST have a secure mechanism for remote administrative access and maintenance. Failures happen; given the 24x7 support requirement (per 4.5), there will be times when something breaks badly enough that senior wizards will have to connect remotely. Remote logins MUST be protected by a secure means that is strongly authenticated and encrypted, and sites from which remote login is allowed MUST be protected and hardened.

In other words, a name server should only function as name server. If needed it can be remotely administered with a protocol like ssh. Take into account possibly backdoors that are created by console servers.

3.2.2 Root name servers SHOULD NOT trust other hosts, except secondary servers trusting the primary server, for matters of authentication, encryption keys, or other access or security information. If a root operator uses kerberos authentication to manage access to the root server, then the associated kerberos key server MUST be protected with the same prudence as the root server itself. This applies to all related services which are trusted in any manner.

> 3.2.3 The LAN segment(s) on which a root server is homed MUST NOT also home crackable hosts. I.e. the LAN segments should be switched or routed so there is no possibility of masquerading. Some LAN switches aren't suitable for security purposes, there have been published attacks on their filtering. While these can often be prevented by careful configuration, extreme prudence is recommended. It is best if the LAN segment simply does not have any other hosts on it.

> 3.2.4 The LAN segment(s) on which a root server is homed SHOULD be separately firewalled or packet filtered to discourage network access to any port other than those needed for name service.

Care needs to be taken with regards to the firewall implementation. If the firewall does "deep packet inspection", it should support all extensions of the DNS protocol that the name servers also support.

> 3.2.5 The root servers SHOULD have their clocks synchronized via NTP [RFC1305] [RFC2030] or similar mechanisms, in as secure manner as possible. For this purpose, servers and their associated firewalls SHOULD allow the root servers to be NTP clients. Root servers MUST NOT act as NTP peers or servers.

> 3.2.6 All attempts at intrusion or other compromise SHOULD be logged, and all such logs from all root servers SHOULD be analyzed by a cooperative security team communicating with all server operators to look for patterns, serious attempts, etc. Servers SHOULD log in GMT to facilitate log comparison.

Especially for functions like TSIG authentication it is important that the computer clocks are synchronized. To prevent NTP based DoS attacks we recommend the use of NTP4 with authentication[12].

> 3.2.7 Server logging SHOULD be to separate hosts which SHOULD be protected similarly to the root servers themselves.

> 3.2.8 The server SHOULD be protected from attacks based on source routing. The server MUST NOT rely on address- or name-based authentication.

> 3.2.9 The network on which the server is homed SHOULD have in-addr.arpa service.

6.3 Protocol

3.3 Protocol authentication and security are required to ensure that data presented by the root servers are those created by those authorized to maintain the root zone data.

3.3.1 The root zone MUST be signed by the Internet Assigned Numbers Authority (IANA) in accordance with DNSSEC, see [RFC2535] or its replacements. It is understood that DNSSEC is not yet deployable on some common platforms, but will be deployed when supported.

3.3.2 Root servers MUST be DNSSEC-capable so that queries may be authenticated by clients with security and authentication concerns. It is understood that DNSSEC is not yet deployable on some common platforms, but will be deployed when supported.

3.3.3 Transfer of the root zone between root servers MUST be authenticated and be as secure as reasonably possible. Out of band security validation of updates MUST be supported. Servers MUST use DNSSEC to authenticate root zones received from other servers. It is understood that DNSSEC is not yet deployable on some common platforms, but will be deployed when supported.

TSIG[22] provides adequate security using a proven technology. If there are reasons to distribute the zone in a different way than AXFR/IXFR (e.g. rsync or ftp) then we recommend to apply PGP signatures for the transport or use a secure transport layer (like SSL).

3.3.4 A 'hidden primary' server, which only allows access by the authorized secondary root servers, MAY be used.

3.3.5 Root zone updates SHOULD only progress after a number of heuristic checks designed to detect erroneous updates have been passed. In case the update fails the tests, human intervention MUST be requested.

If TSIG is used, it is only necessary to confirm that the zone loaded by the master has no problems. The heuristic of a check could for example be to verify the size of the changes in the zone and compare that to the normal pattern.

3.3.6 Root zone updates SHOULD normally be effective no later than 6 hours from notification of the root server operator.

3.3.7 A special procedure for emergency updates SHOULD be defined. Updates initiated by the emergency procedure SHOULD be made no later than 12 hours after notification.

3.3.8 In the advent of a critical network failure, each root server MUST have a method to update the root zone data via a medium which is delivered through an alternative, non-network, path.

3.3.9 Each root MUST keep global statistics on the amount and types of queries received/answered on a daily basis. These statistics must be made available to RSSAC and RSSAC sponsored researchers to help determine how to better deploy these machines more efficiently across the internet. Each root MAY collect data snapshots to help determine data points such as DNS query storms, significant implementation bugs, etc.

We strongly recommend proper logging and accounting. We also extensively document that in this report.

6.4 Active relationships

The operators of name servers should develop and maintain active relationships with upstream network provider(s) and/or peering partners. We strongly recommend to do this pro-active, so that in crisis situations there are already established contacts and communication will go efficiently. Besides the network contacts we also recommend to maintain these kind of relationships with CERT organizations.

6.5 Incident Response Plan

Besides technically preparing and training people, and all other related preventive measures for attacks and crisis, we recommend to develop an incident response plan that documents all these procedures in one place. The plan should be easy to find and execute.

6.6 Stability through Variation

The DNS has been designed with resiliency in mind. At the (g/cc)TLD level it is important that service is always guaranteed even if individual server instances fail. In this section we will argue for variation in all aspects of the service.

6.6.1 Variation in building blocks

6.6.1.1 Hardware With variety in hardware we can exclude that risk that a certain production fault can cause all hardware to experience problems at more or less the same time.

6.6.1.2 Operating Systems As described earlier in this paper, there have been incidents that a single network packet sent remotely could crash a machine reproducibly over and over again. These kind of software bugs are generally OS dependent, but as many OS's have the same roots, they might also share bugs like these. When selecting OS's it might also be noteworthy to compare network stacks and see if there is variance in it.

6.6.1.3 Name server application software The nameserver software implementation is the core part of the DNS service. Also for this part variety means stability.

6.6.1.4 Networking By logically spreading the network locations of the various servers, impact of network outages can be reduced (depending on the scale).

6.6.1.5 Physical locations By physically spreading the locations of the various servers, impact of environmental disasters can be reduced (depending on the severity).

6.6.1.6 Powergrids By logically spreading the locations on the powergrid of the various servers, impact of power outages can be reduced (depending on the scale).

6.6.1.7 Procedures By using uncoordinated procedures for both the administration and the operation of the nameservers at different organizations, it will reduce the chance of duplicated procedural errors.

6.6.1.8 Organizations involved Having the nameservers operated by different (type of) organizations ensures that all of the other variations in this section will happen automatically.

6.7 DNSSEC

6.7.1 Additional requirements

By using DNSSEC there are added requirements for the DNS supporting infrastructure. These can be divided in three categories which are further described here.

6.7.1.1 DNSSEC support by all authoritative name servers Naturally it requires the name server software to comply with the DNSSEC (draft) standards. All the authoritative name servers of a domain must support DNSSEC for it to operate correctly. Also, if there is intermediate "intelligent" network equipment involved that handles DNS packets, it needs to be aware about DNSSEC to forward it correctly.

6.7.1.2 Zone signing and key management

Before a DNS zone can be published it needs to be signed. The decision need to be made where in the provisioning chain the signing will happen. The second issue is the storage and protection of the private keys. Also needs to be decided what the lifetime of the keys will be and how information about updated keys will be communicated.

6.7.1.3 Back-end systems for secure delegation

To make practical use of DNSSEC signed zones, which the client can use to authenticate DNS responses, the parent needs to include signed delegation keys from the child. The TLD operator needs to establish a mechanism to get the delegation key-set from the child to the parent in a trusted way and have it administered safely.

6.7.2 What DNSSEC solves

This section describes the (possible) solutions that DNSSEC would bring for the various DNS data integrity related problems described in this paper.

DNSSEC provides a data authentication and integrity verification mechanism on the DNS data itself. This allows DNS clients to notice that data has been tampered with or has been recorded and replayed.

Conceptually it can be compared with sending messages in a sealed transparent envelope. The seal provides a means of source authentication, the envelope and the seal help the receiver to establish integrity. In contrast to the seal being applied to the message inside the envelope the does not assert correctness of the actual message but only asserts that the data is as being put in the envelope. The data is not protected for privacy.

Once DNSSEC is deployed it is likely that security data will be distributed via the DNS. The DKIM[9] policy language is an example.

If DNSSEC is employed widely it will also be effective in combating DNS cache poisoning.

6.7.3 What DNSSEC does not solve

DNSSEC does not solve any of the problems that have to do with transport, in fact, since packets are bigger they consume more resources in the servers and on the wire and impose rules on firewalls (like 512 byte limits and packet fragementation). That may provide new vectors for (D)DOS attacks. The authors are of the opinion that not deploying DNSSEC will not be a prohibitive factor for DDOS attacks. Keeping the packets smaller means that the attackers need more machines for their DDOS attacks but since the scaling factors is a constant we feel that the benefits of DNSSEC outweigh the additional DDOS risks.

7 Detection

7.1 Monitoring

In the attack tree we have identified events that are possible to occur. This section goes into monitoring the DNS so that none of these threats stay unnoticed.

7.1.1 System monitoring

This section describes system monitoring, which is the monitoring of the DNS service on a system level.

As the key building block to the DNS as a whole, the actual server systems are a critical part. While the DNS as a service should not suffer from a single server failure, it should be the goal to have as little server outages as possible.

To ensure stable operation of DNS servers, they should be monitored on a system level in a way thats suitable to the chosen Operating System (OS) and hardware.

Likely parameters to monitor are system availability, CPU usage, disk usage, disk space, memory usage, swap/paging behavior, network throughput, etc.

7.1.2 DNS traffic monitoring

7.1.2.1 Passive monitoring Passive DNS monitoring is keeping an eye on all DNS packets coming into and going out of the server. This can either be done on the server itself or on a separate machine.

Passive monitoring on the system itself can be done in two ways. The first is to have monitoring support integrated in the nameserver software. This is probably the simplest, but not the best, and not the cleanest either. It could interfere with the way the nameserver software operates and it is our philosophy that the nameserver software should keep its features to a bare minimum. The second manner is to have some kind of network monitor running on the nameserver system that peeks at all DNS traffic. This doesn't interfere with the way the nameserver software runs but it could have influence on the performance and stability of the server as a whole.

With a network capture card connected to the same network as the nameserver the logging and monitoring could be offloaded to a separate and dedicated monitoring machine. This off-system monitoring would have minimal influence on the nameserver operations and would ensure an objective view on the on-wire activities.

Maintaining historical archives of this data allows for careful analysis and may serve as forensic data.[1]

7.1.2.2 Active monitoring With active monitoring the monitor needs to send "real" DNS queries to the nameservers to get full insight what the results are. To be useful, these monitors need to be placed at representative locations on the network (Internet). The RIPE NCC DNSMON[5] service is a good example of this type of monitoring.

7.1.2.3 DNS statistics While this section was mainly aimed at operational purposes, it needs to be recognized that statistics are valuable and should be kept as long as possible to be able to do trend analysis on any of the parameters mentioned earlier in this section.

[1] Privacy concerns need to be considered

7.2 Social networking

Besides technical monitoring it is also important to have a close watch of relevant Internet fora and user groups. As many attack may have a global character, it might well be that an attack reaches other countries first before you are under attack.

8 Reaction

In the unfortunate event that an attack is detected, either early by good monitoring or late by loss of service, action needs to be taken.

8.1 Characterize

The first step into taking action against an attack is finding out the exact nature of the attack. The base for this should be the information gathered by monitoring, and ideally the information that triggered the detection system. However, it is quite likely that more investigation is needed to fully characterize the attack.

8.2 Mitigation

In the event that the attack is not strong enough to bring the network down, but still still has a big impact on the service certain measures can be taken individually.

If the attack can be located and identified the operator can choose to drop all that traffic at the border router. When the attacking traffic is hard to identify this becomes more difficult though.

In the case that the traffic is of a certain type, the decision can be made to drop all traffic of that type so that other types of requests can still be serviced.

8.3 Escalate and cooperate

When the ((D)DoS) attack can not be dealt with within the local network it needs to be coordinated with upstream and/or peer network providers.

8.4 Post-mortem analysis

When the attack is settled and the operation goes back to normal we recommend to conduct a post-mortem analysis.

In that way it is possible determine the attack type and where it came from (if it was not discovered already). By analyzing the impact of the attack (technically, financially, etc) it may become apparent that structural improvements of the security are well justified to prevent similar impact in a possible future attack. By evaluating how the attack was noticed (did an alarm go off or did a customer complain?) the effectiveness of early warning systems can be measured. To improve the Incident Response Plan, an evaluation of what went well and what did not go well regarding the IRP is essential. In line with that the possible

cooperations with other organizations need to be evaluated. Were all contacts in place or did new contacts have to be created in the middle of the night? Finally and optionally, it may be the case that you want to analyse legal causes for yourself, but also investigate what actions can be taken against the attackers.

Part IV
Conclusions and outlook

In this document we describe various potential problems surrounding the DNS. Although it is admirable that the design of the DNS scaled so well with the growth of the Internet, the original inventors did not take security issues as serious as it needs to be taken currently.

We show that one type of weakness in the DNS, namely data integrity, can be solved by employing DNSSEC.

Transport and availability of the DNS is difficult to guarantee. (Distributed) Denial of Service attacks currently have no effective counter measures except for extreme overprovisioning and containment.

DDoS attacks are not specifically a problem for the DNS, but are a danger for all (core) Internet services (as a whole). The counter measures that can be taken are only effective if they are taken by a group of networks at large, individually there is not much that can be done against large scale attacks.

DDoS mitigation is an active research field, academic solutions have been developed, but it might take a while until they reach the operational world.

Finally we stress the importance of well maintained contacts with peer networks, local and global communities and incident handling organizations.

A Organizations

There is no magic one-stop desk for information. However, there are various forums and places that people can go to for informations and collaboration.

A.1 Computer Emergency Rescue Teams (CERT)

CERTs are organizations that monitor and maintain information about security threats. They have established a network with vendors and are involved in an early stage when vulnerabilities are discovered.

If specifically interested in DDoS preparedness membership of a CERT is probably a good thing to have.

A.2 Internet Exchange Points

The places where the organization is connected to the Internet are useful places to create contacts as other members of those exchange points may be under attack at the same time. While operators may be competing on exchange points, they certainly also have a lot common interest regarding attacks to providers on the exchange point. That creates a good basis for professional relationships.

A.3 Operator groups

(Network) operator forums like RIPE (Europe), NANOG (North-America), AfNOG (Africa), APRICOT (Asia-Pac), DNS-Operations (IETF) and OARC (ISC) are a useful source of practical technical information.

A.4 Software communities

We also recommend to follow the ongoings of the operating systems and the software used on the name servers. Generally this can be done by subscribing to mail lists of the respective products. Many projects have a security announcement mail list that you can subscribe to for notifications of security threats and software updates. Also fora where users share experiences about the software they use.

A.5 Research

The above mentioned OARC group also includes researchers. OARC acts as a clearinghouse for the exchange of knowledge, expertise, and data (like packet traces) between operators and researchers.

B About NLnet Labs

NLnet Labs was founded in 1999 by Stichting NLnet to develop, implement, evaluate and promote new protocols and applications for the Internet.

The goal of NLnet Labs is to contribute knowledge to the Internet. This can be achieved by software development, and also by educating people to develop software or deploy protocols. NLnet Labs' staff therefore not only focuses on software development defined in projects, but also on collaboration with other organizations. The budget of NLnet Labs is based on long term investment for development with a staff of five to six people.

C About the sponsor of this paper

.SE (The Internet Infrastructure Foundation), founded as a non-profit organisation in 1997, is responsible for the top-level Internet domain for Sweden, .se. .SE's core operations are registration of domain names and the administration and technical operation of the national domain name register.

Within the framework of these operations, .SE works to ensure the positive development of the Internet in Sweden over the long term, for the benefit of users, operators, businesses, authorities, universities and others. This way, .SE wants users of domain-name services to have access to high-quality, robust services on reasonable terms.

References

[1] J. Abley and K. Lindqvist. *Operation of Anycast Services*. RFC 4786 (Best Current Practice), December 2006. http://www.ietf.org/rfc/rfc4786.txt.

[2] F. Baker and P. Savola. *Ingress Filtering for Multihomed Networks*. RFC 3704 (Best Current Practice), March 2004. http://www.ietf.org/rfc/rfc3704.txt.

[3] R. Bush, D. Karrenberg, M. Kosters, and R. Plzak. *Root Name Server Operational Requirements*. RFC 2870 (Best Current Practice), June 2000. http://www.ietf.org/rfc/rfc2870.txt.

[4] DNS Amplification Attacks. http://www.isotf.org/news/dns-amplification-attacks.pdf.

[5] *RIPE NCC DNSMON web pages*. RIPE NCC DNSMON pages. http://www.ripe.net/dnsmon/.

[6] P. Ferguson and D. Senie. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. RFC 2827 (Best Current Practice), May 2000. http://www.ietf.org/rfc/rfc2827.txt, (Updated by RFC 3704).

[7] Evgeniy Gabrilovich and Alex Gontmakher. *The Homograph Attack*. http://www.cs.technion.ac.il/~gabr/papers/homograph%_full.pdf.

[8] Luis Grangeia. *DNS Cache Snooping*. http://www.sysvalue.com/papers/dns-cache-snooping/files/dns_cache_snooping_1.1.pdf.

[9] T. Hansen, D. Crocker, and P. Hallam-Baker. *DomainKeys Identified Mail (DKIM) Service Overview*, October 2006. http://www.ietf.org/internet-drafts/draft-ietf-dkim-overview-03.txt, (Internet Drafts are subject to change and have a limited lifetime; this draft has expired).

[10] A. Hubert and R. van Mook. *Measures for making DNS more resilient against forged answers*, January 2007. http://www.ietf.org/internet-drafts/draft-ietf-dnsext-forgery-resilience-00.txt, (Internet Drafts are subject to change and have a limited lifetime; this draft has expired).

[11] B. Laurie, G. Sisson, R. Arends, and D. Blacka. *DNSSEC Hashed Authenticated Denial of Existence*, January 2007. http://www.ietf.org/internet-drafts/draft-ietf-dnsext-nsec3-09.txt, (Internet Drafts are subject to change and have a limited lifetime; this draft has expired).

[12] David Mills. NTP4 Authentication options. http://www.eecis.udel.edu/~mills/ntp/html/authopt.html.

[13] Kevin D. Mitnick and William L. Simon. *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons, Inc., New York, NY, USA, 2003.

[14] P.V. Mockapetris. *Domain names - implementation and specification*. RFC 1035 (Standard), November 1987. `http://www.ietf.org/rfc/rfc1035.txt`, (Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 2137, 2845, 3425, 3658, 4035, 4033).

[15] Bruce Schneier. *Secrets & Lies*. John Wiley& Sons, Inc, 2000. isbn 0471253111.

[16] ICANN Security and Stability Advisory Committee (SSAC). *DOMAIN NAME HIJACKING: INCIDENTS, THREATS, RISKS, AND REMEDIAL ACTIONS*. `http://www.icann.org/announcements/hijacking-report-12jul05.pdf`.

[17] The "stacheldraht" distributed denial of service attack tool. `http://staff.washington.edu/dittrich/misc/stacheldraht.analysis`.

[18] The "Tribe Flood Network" distributed denial of service attack tool. `http://staff.washington.edu/dittrich/misc/tfn.analysis`.

[19] TFN2K - An Analysis. `http://packetstormsecurity.org/distributed/tfn2k_analysis.htm`.

[20] The DoS Project's "trinoo" distributed denial of service attack tool. `http://staff.washington.edu/dittrich/misc/trinoo.analysis`.

[21] P. Vixie. *A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)*. RFC 1996 (Proposed Standard), August 1996. `http://www.ietf.org/rfc/rfc1996.txt`.

[22] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington. *Secret Key Transaction Authentication for DNS (TSIG)*. RFC 2845 (Proposed Standard), May 2000. `http://www.ietf.org/rfc/rfc2845.txt`, (Updated by RFC 3645).

[23] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. *Dynamic Updates in the Domain Name System (DNS UPDATE)*. RFC 2136 (Proposed Standard), April 1997. `http://www.ietf.org/rfc/rfc2136.txt`, (Updated by RFCs 3007, 4035, 4033, 4034).