

Beveiliging van DNS

Information Security - 2IF30

Mark Bergsma
483608

6 november 2003

1 Inleiding

DNS (Domain Name System) is een van de meest elementaire services die gebruikt worden op het Internet. Onzichtbaar op de achtergrond doet het zijn werk bij vrijwel alle handelingen die verricht worden tussen verschillende entiteiten. Doordat dit protocol van zulk groot belang is en vele systemen ervan afhankelijk zijn om elkaar te kunnen vinden, is het ook een aantrekkelijk en effectief doelwit voor aanvallen en misbruik. Dit artikel beschrijft enkele middelen om het DNS-protocol beter te beschermen tegen dit soort misbruik en onderzoekt de effectiviteit en realiseerbaarheid hiervan, en de verschillen tussen de gebruikte methoden.

2 De werking van DNS

De gedetailleerde werking van het DNS-systeem is gecompliceerd, en valt buiten het kader van dit artikel. Voor meer informatie hierover wordt verwezen naar [1], [2] en uiteraard RFC 1034 [3] en 1035 [4]. Omdat echter de werking van DNS van groot belang is voor dit artikel, wordt hieronder een korte uitleg gegeven.

DNS is een gedistribueerde *database* of *directory*. Het is hiërarchisch onderverdeeld in verschillende zones, die records van verschillende typen bevatten. Deze records bevatten de DNS-gegevens. Er wordt onderscheid gemaakt tussen *resolvers* die informatie in deze database opzoeken, en *authoritative servers* die deze informatie aanbieden. Een DNS-server (nameserver) kan elk of beide van deze taken vervullen. Om het IP-adres van bijvoorbeeld `www.win.tue.nl` op te zoeken, wordt door de resolver aan de statisch geconfigureerde *root nameservers* (.) gevraagd naar de juiste authoritative nameservers om deze vraag (query) te beantwoorden. De root nameservers zullen middels een zogenaamd *NS-record* verwijzen naar de authoritative nameservers voor de `nl.`-zone, welke vervolgens zullen verwijzen naar de authoritative nameservers voor de `tue.nl.`-zone. Deze servers (onder beheer van de Technische Universiteit Eindhoven) zullen op hun beurt verwijzen naar de nameservers voor `win.tue.nl.` van de faculteit Wiskunde & Informatica, die de oorspronkelijke vraag kunnen beantwoorden met een *A-record*.

Alle door de resolver gevonden informatie, zoals nameserver-verwijzingen (NS-records) en IP-adressen behorende bij bepaalde hostnames (A-records) wordt opgeslagen in een *cache*, zodat reeds bekende informatie niet voor elke query opnieuw hoeft te worden opgezocht. Dit is voor de schaalbaarheid van het DNS-systeem van groot belang, omdat anders alle queries via de root nameservers zouden moeten verlopen. De informatie wordt in de cache opgeslagen en gebruikt voor een periode van ten hoogste de tijd (TTL, Time To Live) die door de authoritative nameservers als attribuut is meegegeven bij het opgevraagde record.

3 Aanvallen

Aanvallen op het DNS-systeem hebben vrijwel altijd het doel gebruikers te misleiden door het vervalsen van de DNS-data, danwel het geheel verhinderen van het gebruik van betreffende services.

In het eerste geval, vervalsing van DNS-data, ontvangen gebruikers – of dit mensen of geautomatiseerde systemen zijn is in dit geval irrelevant – andere informatie dan gewenst. Dit kan er bijvoorbeeld toe leiden dat er communicatie zal plaatsvinden tussen andere systemen dan de bedoeling was, wat misbruikt kan worden voor bijvoorbeeld *Man-in-the-middle-attacks*.

In het tweede geval is er in feite sprake van een *Denial of Service (DoS)*. Door er bijvoorbeeld voor te zorgen dat de gebruikte DNS-resolver denkt dat een bepaald DNS-record niet bestaat, of de informatie zodanig te vervalsen dat communicatie met de gewenste service eenvoudigweg niet zal lukken, wordt effectief verhinderd dat gebruik kan worden gemaakt van de betreffende service.

Er zijn een aantal veel gebruikte methoden bij aanvallen op DNS. Hieronder zullen er een paar kort worden besproken. Voor een uitgebreidere behandeling van DNS-aanvallen wordt verwezen naar [5].

3.1 Cache-vergiftiging

De waarschijnlijk meest gebruikte DNS-aanval is *cache-vergiftiging* (Engels: *cache poisoning*). Zoals eerder genoemd wordt reeds opgezochte informatie gecached door resolvers voor een betere schaalbaarheid en reactietijd. De cache-vergiftiging-techniek is erop gericht deze cache te vervuilen met valse informatie, zodanig dat deze valse informatie ook gebruikt wordt voor uitgifte van informatie aan clients (gebruikers van de opgevraagde informatie) en andere nameservers.

Cache-vergiftiging is mogelijk doordat het DNS-protocol mogelijkheden biedt aan DNS-servers om extra informatie te verstrekken naast de daadwerkelijk gevraagde informatie. Een DNS-packet bevat namelijk drie secties die antwoorden kunnen bevatten: de *Answer*-sectie, de *Authority*-sectie en de *Additional*-sectie. De *Additional*-sectie is bedoeld voor informatie die weliswaar niet het directe antwoord op een query bevat, maar de vragende client kan helpen bij het vinden van de juiste informatie, of op andere manier van nut kan zijn bij de verwerking ervan. Zo zou de *Additional*-sectie bijvoorbeeld de IP-adressen (A-records) van nameservers waarnaar verwezen wordt in de *Authority*-sectie kunnen bevatten, zodat de vragende client dit niet in een aparte query hoeft op te zoeken. Het probleem hiermee is dat een nameserver in de *Additional*-sectie informatie kan verstrekken waarvoor deze niet authoritative is. Als bijvoorbeeld een authoritative nameserver voor het `tue.nl`-domein A-records met IP-informatie betreffende de nameservers van het `nedworks.org`-domein zou meegeven bij een query voor het `win.tue.nl`-domein, en de client resolver deze informatie klakkeloos zou opnemen in zijn cache, dan kan de resolver-cache vergiftigd worden indien deze IP-informatie vals is. Aanvallers kunnen zo gericht de cache van een resolver proberen te vervuilen door deze te verleiden een query te doen naar een nameserver onder beheer van de aanvaller, door middel van een query voor een domein waar de nameserver van de aanvaller authoritative voor is. Bij het beantwoorden van de query door de doelwit-resolver aan de nameserver van de aanvaller, kan de laatste vergiftigde informatie verstrekken. Deze en andere methoden van cache-vergiftiging worden met al hun subtiliteiten behandeld in [5] en [6].

3.2 Client flooding

Een andere techniek van aanval is *client flooding*. De meeste DNS-queries worden afgehandeld met behulp van het stateless UDP-protocol. Door gebruik te maken van *IP-spoofing*-technieken kan een aanvaller het IP-adres (en daarmee de identiteit) van een willekeurige authoritative nameserver aannemen. Zodra een client resolver een query doet bij de authoritative nameserver, kan de aanvaller – gebruikmakende van het bron-IP-adres van de authoritative nameserver – een DNS UDP-packet sturen naar de client en deze laten denken dat het antwoord afkomstig is van de echte authoritative nameserver. Dit vereist natuurlijk dat de aanvaller weet dat de client resolver een verzoek heeft gestuurd naar de authoritative nameserver. Dit zou de aanvaller bijvoorbeeld kunnen bewerkstelligen door middel van *packet sniffing* en/of de client verleiden tot het doen van queries zoals beschreven in 3.1.

Uiteraard zal de echte authoritative nameserver ook een antwoord sturen. Afhankelijk van de implementatie zal de client resolver één van deze antwoorden selecteren en gebruiken. De aanvaller kan het antwoord van de authoritative nameserver ook proberen te onderdrukken door deze te overvloeden met packets, bijvoorbeeld met een *Distributed Denial of Service-attack* (DDoS).

Het DNS-protocol biedt wel enige bescherming tegen deze methode. Elk DNS packet bevat een *Transaction ID* [4] dat gebruikt kan worden om een antwoord-packet te koppelen aan een bepaalde query. De resolver kan dan packets met een onjuist ID verwerpen. Oude DNS-implementaties gebruikten hiervoor echter een sequentieel oplopende nummering [7], waardoor de gebruikte ID's eenvoudig te voorspellen waren. Door de client te 'flooden' met enkele antwoord-packets met verschillende ID's die in een reeks met te voorspellen nummers liggen, is dit voor de aanvaller geen echte belemmering.

3.3 Vervalsing van authoritative DNS data

Een van de meest elementaire aanvallen op DNS is uiteraard inbreuk op de authoritative nameservers zelf via wegen buiten het DNS-protocol om (zoals via exploits op services draaiende op de betreffende nameservers), gevolgd door aanpassingen van de authoritative DNS-data. Het spreekt voor zich dat zonder uitgebreide maatregelen m.b.t. het DNS-protocol er geen mogelijkheid is voor buitenstaanders om deze vervalsing te ontdekken.

4 Beveiligingstechnieken

Sommige van de bovengenoemde aanvallen zijn met betrekkelijk eenvoudige aanpassingen in de DNS-implementaties te verhelpen of op z'n minst in effectiviteit te reduceren. Andere vereisen grootschalige aanpassingen van het DNS-protocol. Van beide soorten zullen enkele mogelijkheden worden besproken.

4.1 Conventionele technieken

Een belangrijk beginsel van het voorkomen van cache-vergiftiging en client-flooding is het controleren van de geldigheid (met betrekking tot de originele query) van beschikbare gegevens in query-antwoorden als bron-IP-adressen en (cryptografisch willekeurige) transaction ID's. Ook dienen resolver-implementaties uitsluitend records in hun cache op te nemen van bronnen die autoritatieve zijn voor de betreffende records. [9] Oude DNS-implementaties, met name oude Bind [8] versies (Bind is sinds het ontstaan van het DNS-systeem de meest gebruikte DNS-implementatie op het Internet), waren hier duidelijk minder strict in. Dit is verholpen in latere versies/implementaties (in het geval van Bind 9 met een volledige rewrite).

[10] geeft een formele benadering van deze principes.

Ook meer algemene maatregelen buiten het DNS-protocol om, zoals *egress filtering* op de border routers van ISP-netwerken, helpen om DNS-aanvallen (client spoofing met IP-spoofing in het bijzonder) tegen te gaan.

4.2 TSIG

TSIG [11] is een uitbreiding op het DNS-protocol dat authenticatie en data-integriteit biedt tussen twee communicerende nameservers. Het werkt volgens een eenvoudig principe: *hashing*. Een DNS-packet bestaat in feite uit een header gevolgd door een of meerdere DNS-records van verschillende typen. TSIG voegt aan het einde van een conventioneel DNS-packet een extra *TSIG-meta-record* toe. Dit record bevat een cryptografische hash (doorgaans HMAC-MD5) van de raw data van het gehele voorafgaande dns-packet, gecombineerd met een statisch geconfigureerde *cryptografische sleutel* specifiek voor de ontvangende nameserver, en de huidige tijd. De cryptografische sleutel, die specifiek is voor een bepaald paar nameservers, wordt toegevoegd als bewijs dat het packet inderdaad van de juiste nameserver afkomstig is. De huidige tijd wordt gebruikt in de hash om *replay-attacks* te voorkomen.

De nameserver waarmee gecommuniceerd wordt heeft de beschikking over dezelfde cryptografische sleutel, de huidige tijd en dezelfde raw data van het DNS-packet, en kan daarmee de authenticiteit en integriteit van de transactie controleren.

Een aantal elementen kunnen worden opgemerkt. Allereerst valt op dat TSIG enkel zorg draagt voor *authenticatie* en *integriteit*, maar niet voor *geheimhouding*. Het gaat uit van het principe dat DNS-data publiekelijk is. Dit lijkt een redelijke veronderstelling, zeker gezien het feit dat geheimhouding verdergaande aanpassingen aan het DNS-berichtenformaat zou vergen, waar dit nu niet het geval is. De TSIG-standaard is dan ook backward-compatible met de DNS-standaarden in [3] en [4], en is vrij *lightweight* in de zin dat zowel resolvers als authoritative nameservers geen ingrijpende wijzigingen hoeven te ondergaan om TSIG te kunnen implementeren. Zoals veel andere beveiligingsprotocollen maakt het gebruik van de huidige tijd om replay-attacks te kunnen

verslaan. Hierdoor is het van belang dat de twee communicerende nameservers over dezelfde notie van huidige tijd beschikken. Dit is in het algemeen bij security van belang, en kan bijvoorbeeld worden bewerkstelligd middels NTP [12]. Het TSIG-record bevat echter een veld dat het aantal seconden specificeert waarmee de tijd van de ontvangende nameserver mag afwijken — dit is ook van belang doordat vertragingen kunnen optreden bij het versturen van het packet over het netwerk. TSIG draagt niet zorg voor de distributie van de cryptografische sleutels, en is daardoor zonder aanvullende maatregelen slechts geschikt voor gebruik tussen nameservers in een beperkt administratief domein. Het authenticaceert bovendien slechts de nameserver (of eigenlijk, het *paar* nameservers waartussen een bepaalde sleutel geconfigureerd is) waarvan de gegevens afkomstig zijn, en niet de gegevens zelf.

4.3 DNSSEC

Een uitgebreide poging om het DNS-beveiligingsprobleem grotendeels op te lossen wordt ondernomen door *DNSSEC*, DNS Security Extensions. Door gebruik te maken van *public-key cryptografie*, het signeren van afzonderlijke DNS-records en het creëren van een *chain of trust* zouden veel van de huidige problemen moeten worden opgelost.

DNSSEC is een zeer uitgebreid onderwerp, met veel implementatie-specifieke details. In dit essay kan slechts het concept erachter worden besproken. Voor meer informatie wordt verwezen naar [13], [14] en natuurlijk de op het moment van schrijven nog geldende specificatie in RFC 2535 [15].

Om authenticiteit en integriteit te kunnen waarborgen, maakt DNSSEC gebruik van public-key cryptografie. Voor iedere DNS-zone dient een tweetal cryptografische sleutels te worden gegenereerd: een *private key* en een *public key*. De private key dient geheim te blijven, en daartoe op een zo veilig mogelijke plaats bewaard. De public key dient bekend te worden gemaakt aan eenieder die gegevens uit de betreffende DNS-zone wil kunnen controleren, en dient daarom te worden gepubliceerd in bijvoorbeeld een publieke directory. DNS is zelf een grote gedistribueerde publieke directory, en kan daarom uitstekend gebruikt worden voor dit doel. DNSSEC definieert hiervoor het *KEY-record*, dat de publieke sleutel voor gebruik met een specifiek cryptografisch algoritme voor het betreffende domein zal bevatten.

Het daadwerkelijk signeren van de DNS-records wordt gedaan door voor elke set DNS-records van hetzelfde *type* en dezelfde *naam* (bijvoorbeeld: alle A-records met de naam *www.tue.nl.*) een *SIG-record* in de zone op te nemen. Dit SIG-record bevat als data een *hash* van de bijbehorende record-set, versleuteld met de *private zone-key*. Door alle DNS-records in de zone op deze wijze te signeren, kan een resolver met behulp van de gebruikte public key – uit een bovengenoemd KEY-record – verifiëren dat een bepaald record afkomstig is van de juiste eigenaar van de zone, en authentiek is.

De resolver dient daarvoor uiteraard wel te kunnen beschikken over de public key. Aangezien deze zich in dezelfde DNS-zone bevindt waarin de te controleren DNS-gegevens staan, en deze dus mogelijk tegelijk met de overige DNS-data gewijzigd zou kunnen zijn door een aanvaller, dient er een manier te zijn om de integriteit van de KEY-records te controleren. Hiervoor wordt een *chain of trust* gecreëerd, door gebruik te maken van de boomstructuur van de DNS-directory. Elke public zone-key wordt – bij nameserver-verwijzingen van een zone op een hoger niveau naar een zone op een lager niveau – gesigneerd met behulp van een SIG-record en de private key van de zone in een hoger niveau. Bijvoorbeeld: het public zone KEY-record van het domein *tue.nl.* wordt vergezeld van een SIG-record dat het KEY-record signeert met behulp van de *private key* van de *nl.-zone*. De public key van de *root* (.)-zone wordt statisch geconfigureerd in elke resolver.

Dit laat echter nog één probleem open. De authenticiteit en integriteit van bestaande DNS-records kan nu geverifieerd worden. De authenticiteit van het *ontbreken* van een record (een zogenaamde NXDOMAIN-response) echter niet, aangezien deze melding eenvoudig door elke aanvaller te genereren is. Om dit probleem op te lossen wordt in een met DNSSEC beveiligde zone een uniforme ordening aangebracht tussen de DNS-records. Deze uniforme ordening (die op het alfabet gebaseerd is) wordt in de zone gespecificeerd door de toevoeging van *NXT-records*. Zo specificeert het

record `a.dnssec.example.` NXT `c.dnssec.example.` bijvoorbeeld dat het record met de naam `a.dnssec.example.` zich direct vóór het record `c.dnssec.example.` in de ordening bevindt. Ook dit NXT-record wordt gesigneerd met een SIG-record op de hierboven beschreven methode.

Indien een resolver nu een query doet voor het niet bestaande record `b.dnssec.example.`, dan ontvangt het in plaats van een NXDOMAIN-response het hierboven genoemde NXT-record. Met behulp van het corresponderende SIG-record kan de resolver verifiëren dat het NXT-record authentiek is, en er dus geen 'tussenliggend' record voor `b.dnssec.example.` kan bestaan.

Opnieuw valt op dat ook DNSSEC geen poging doet tot geheimhouding van de DNS-gegevens, en – net als TSIG – enkel *authenticatie* en *integriteit* biedt. Sterker nog, door de noodzaak van het gebruik van NXT-records ligt de inhoud van de zone compleet open omdat men eenvoudigweg de geordende reeks NXT-records één voor één kan aflopen. Het verbieden van zonetransfers (AXFRs) in de nameserver-configuratie heeft met DNSSEC dan ook geen zin.

Het gebruik van DNSSEC heeft vrij veel effect op de werking van authoritative nameservers en met name resolvers. Allerlei kleine implementatie-specifieke details dienen (met behulp van experimenten uit de praktijk) uitgewerkt te worden, met behoud van achterwaartse compatibiliteit. Dit is één van de oorzaken waardoor de DNSSEC-standaard nog niet uitgekristalliseerd is.

Een voorbeeld van een probleem dat het gebruik van DNSSEC oplevert, is de maximale bericht-grootte van een DNS-packet: 512 bytes [4]. Voor normale, onbeveiligde DNS-transacties is deze packet-grootte geen probleem, maar het gebruik van DNSSEC vermenigvuldigt de grootte van een DNS-transactie vaak met een factor 3 of meer. Er is dan ook een voorstel [16] om de maximale packet-grootte voor DNSSEC-conforme implementaties te vergroten.

Het gebruik van de chain of trust vanuit de root-zone zorgt ervoor dat globale medewerking en overeenstemming noodzakelijk is voordat het gebruik van DNSSEC echt van de grond kan komen. Concreet betekent dit dat zowel de root-server operators (en ICANN [17], dat de root-zone verzorgt) en alle top-level domain registries medewerking zullen moeten verlenen. De Nederlandse (nl.) registry, SIDN [18] is als eerste gestart met een experiment met betrekking tot het gebruik van DNSSEC. [19] De ervaringen opgedaan in dit experiment zijn te vinden in [20].

Op het moment van schrijven is er een IETF Draft [21] in de maak die uiteindelijk RFC 2535 zal vervangen. Deze draft is incompatible met de hierboven beschreven standaard, maar berust wel op dezelfde concepten.

5 Conclusie

De beschreven beveiligingstechnieken benaderen de problemen met betrekking tot de beveiliging van DNS op verschillende manieren. De conventionele technieken (4.1) gebruiken het bestaande DNS-protocol, en verbeteren de implementaties ervan. Deze technieken zijn dan ook reeds in gebruik in de meeste DNS-implementaties.

Betere beveiliging bieden echter standaarden als TSIG en DNSSEC. Zij breiden het DNS-protocol uit, en zorgen voor meer zekerheid tegen aanvallen door middel van cryptografische middelen.

Welke van de twee beschreven technieken, TSIG of DNSSEC, gebruikt zou moeten worden is niet de vraag. Ze beconcurreren elkaar niet, maar completeren elkaar. TSIG is eenvoudig in implementatie, en beveiligt slechts de transactie tussen nameservers. DNSSEC is uitgebreid en gecompliceerd, en beveiligt de integriteit en authenticiteit van de gegevens zelf. Door de private keys van de DNS-zones offline te bewaren kan DNSSEC bijvoorbeeld zelfs bescherming bieden tegen inbreuk op de authoritative DNS-servers zelf.

Een voorbeeld waarin TSIG en DNSSEC elkaar kunnen aanvullen is bij het gebruik van zogenaamde stub-resolvers. Veel resolver libraries, die gebruikt worden door elk programma dat gebruik maakt van DNS, bevatten niet de volledige resolver-functionaliteit. In plaats daarvan maken ze slechts contact met een statisch geconfigureerde nameserver op het lokale netwerk, die als volwaardige resolver fungeert. De volledige implementatie van DNSSEC zal in dit geval enkel op de (door het lokale netwerk vertrouwde) resolver/nameserver kunnen plaatsvinden. Omdat dit echter

ruimte overlaat voor aanvallen op DNS op het lokale netwerk, kan het eenvoudige TSIG gebruikt worden voor de communicatie tussen de stub-resolver-library en de lokale resolver/nameserver.

Welke van de genoemde beveiligingsstandaarden wanneer daadwerkelijk op grote schaal gebruikt zullen gaan worden blijft echter vooralsnog onduidelijk.

Referenties

- [1] DNS and Bind, Albitz & Liu, 4e editie (2001), O'Reilly & Associates
- [2] How Domain Name Servers Work, <http://www.howstuffworks.com/dns.htm>
- [3] RFC 1034, Domain Names - Concepts and Facilities, P. Mockapetris, <http://rfc.sunsite.dk/rfc/rfc1034.html>
- [4] RFC 1035, Domain Names - Implementation and Specification, P. Mockapetris, <http://rfc.sunsite.dk/rfc/rfc1035.html>
- [5] Domain Name System (DNS) Security, Diane Davidowicz, <http://compsec101.antibozo.net/papers/dnssec/dnssec.html>
- [6] DNS Cache Poisoning - The Next Generation, Joe Stewart, <http://www.securityfocus.com/guest/17905>
- [7] CERT Advisory CA-1997-22 Bind - the Berkeley Internet Name Daemon, <http://www.cert.org/advisories/CA-1997-22.html>
- [8] ISC Bind, Berkeley Internet Name Daemon, <http://www.isc.org/products/BIND/>
- [9] Notes on the Domain Name System, D.J. Bernstein, <http://cr.yip.to/djbdns/notes.html>
- [10] A Formal-Specification Based Approach for Protecting the Domain Name System, Steven Cheung & Karl N. Levitt, University of California, http://seclab.cs.ucdavis.edu/papers/Cheung_LevittDNS.pdf
- [11] RFC 2845, Secret Key Transaction Authentication for DNS (TSIG), P. Vixie & O. Gudmundsson & D. Eastlake & B. Wellington, <http://rfc.sunsite.dk/rfc/rfc2845.html>
- [12] NTP: The Network Time Protocol, <http://www.ntp.org>
- [13] DNS and Bind, Albitz & Liu, 4e editie (2001), O'Reilly & Associates, Hoofdstuk 11: Security, online versie op <http://www.oreilly.com/catalog/dns4/chapter/ch11.html>
- [14] DNSSEC.NET - DNS Security Extensions, J. Tünnissen, <http://www.dnssec.net>
- [15] RFC 2535, Domain Name System Security Extensions, D. Eastlake, <http://rfc.sunsite.dk/rfc/rfc2535.html>
- [16] RFC 3226, DNSSEC and IPv6 A6 aware server/resolver message size requirements, O. Gudmundsson, <http://rfc.sunsite.dk/rfc/rfc3226.html>
- [17] ICANN, The Internet Corporation for Assigned Names and Numbers, <http://www.icann.org>
- [18] SIDN, Stichting Internet Domeinregistratie Nederland, <http://www.sidn.nl>
- [19] SECREG, DNSSEC experiment by NLNetLabs in collaboration with SIDN, <http://secreg.nlnetlabs.nl>
- [20] DNSSEC in NL, Intermediate report, R. Gieben, <http://miek.nl/publications/dnssecnl/secreg-report.pdf>
- [21] draft-ietf-dnssec-protocol-02, Protocol Modifications for the DNS Security Extensions, R. Arends & M. Larson & R. Austein & D. Massey & S. Rose, <http://www.ietf.org/internet-drafts/draft-ietf-dnssec-protocol-02.txt>