Ausarbeitung

Introduction to DNS (RFC 1034, RFC 1035)

DNS vulnerabilities

Referent: Florian Oerterer

28.06.2004

Inhaltsverzeichnis

1. Einleitung

Das Ziel dieses Vortrages ist es, eine Einführung in die Funktionsweise des „Domain Name Systems“ (DNS) zu geben und seine Sicherheitsproblematik zu erläutern. Des Weiteren werden einige Maßnahmen zur Verbesserung der Sicherheit des bestehenden Systems gezeigt.

2. Einführung in das DNS

Die folgenden Abschnitte beschreiben die grundlegenden Hintergründe und die Funktionsweise des DNS.

2.1. Wozu braucht man das DNS?

Das Internet besteht, wie ein reales Netz, aus Knoten und den dazwischen liegenden Verbindungen. Jeder Knoten, der permanent aktiv ist, besitzt eine eindeutige Kennung, die ihn direkt ansprechbar macht. Diese binäre Kennung ist seine Adresse, die so genannte IP-Adresse. Da es für einen Anwender allerdings sehr umständlich ist, eine Zahlenfolge als Adresse anzugeben, ist man dazu übergegangen, einer IP-Adresse eine ASCII-Zeichenkette (z.B.: www.fh-nuernberg.de) zuzuordnen. Da diese ASCII-Zeichenkette für das Netz unverständlich ist, muss diese nach der Eingabe wieder in die binäre Netzadresse umgewandelt werden, damit die Verbindung zwischen den beiden Parteien aufgebaut werden kann. Dies ist die Aufgabe des „Domain Name Systems“.
Das DNS wird im OSI-Schichtenmodell in die Schicht 7, die so genannte „Anwendungsschicht“, eingeordnet.

2.2. Geschichte des DNS

Frühere Systeme unterstützen nur Punkt zu Punkt Verbindungen zwischen Rechnern anhand von Hardwareadressen.
Der nächste Schritt in der Entwicklung war die Vergabe von Namen anstelle der Hardwareadressen. Alle Computernamen und die dazugehörigen IP-Adressen wurden vom Network Information Center (NIC) in der Datei „host.txt“ verwaltet. Diese Datei wurde jede Nacht an alle Rechner des damaligen ARPANETs[1] verteilt. Das Hauptproblem war hierbei der steigende Verwaltungsaufwand durch die steigende Netzwerkgröße.
1986 wurde schließlich das DNS geschaffen und damit eine Art verteilte Datenbank zur Namensauflösung eingeführt. Diese wurde durch mehrere NICs verwaltet. 1987 wurde das

[1] Das **ARPANET** wurde ursprünglich im Auftrag der US-Luftwaffe in Erwartung eines Atomkrieges ab 1962 von einer kleinen Forschergruppe unter der Leitung von Paul Baran entwickelt. Es ist der Vorläufer des heutigen Internets.

DNS durch die RFCs 1034 und 1035 festgeschrieben. Diese beiden RFCs bilden heute noch die Grundlage für die aktuellen Implementierungen des DNS.

2.3. Funktionsweise

Im Folgenden werden alle für das DNS relevanten Komponenten mit ihren einzelnen Aufgaben erläutert.

2.3.1. Domänen

Das Internet ist hierarchisch gegliedert und weist logisch, aber nicht physikalisch, eine umgedrehte Baumstruktur auf. Jeder Knoten stellt hierbei eine Domäne oder Zone dar. An der Spitze steht die Wurzel („root“). Für ihre Verwaltung sind weltweit zurzeit 13 so genannte „Root-Server“ im Einsatz. Diese verwalten auch die nächste Hierarchiestufe der Domänen, die „Top Level Domains“ wie zum Beispiel .com, .de, .org, .int usw. mit. Die unterste Ebene (Blätter) dieser Baumstruktur, die so genannte „Lowest Level Domain“, beinhaltet schlussendlich die Hostnamen.

Im Prinzip ist jede Domäne für ihre untergeordneten Domänen verantwortlich. So unterliegen z.B. die deutschen Domänen (.de) der Kontrolle des „Deutschen Network Information Centers“ (DeNIC), und damit auch die Vergabe der Domäne „fh-nuernberg.de“ seiner Genehmigung. Alle weiteren untergeordneten Domänen, wie „www.fh-nuernberg.de“ oder „informatik.fh-nuernberg.de“ werden nicht mehr vom DeNIC verwaltet. Dies übernimmt ein separater DNS-Server.

Dieser hierarchische Aufbau ist für das ganze System von entscheidender Bedeutung. Es garantiert Übersichtlichkeit und technische Realisierbarkeit. Leider schafft es auch Abhängigkeiten, welche zu seinem Nachteil ausgenutzt werden können.

Für eine Domäne ist folgendes zu beachten: Eine Adresse muss mindestens aus 3 Teilen bestehen, wobei jedem Teil eine Maximallänge von 63 Zeichen zugeordnet wird. Jeder Adresse ist eine Maximallänge von 255 Zeichen zugeordnet. Außerdem darf ein Teil nur aus alphanumerischen Zeichen und dem Bindestrich, welcher nicht am Anfang stehen darf, aufgebaut werden.

2.3.2. Serverhierarchie

DNS-Nameserver sind, wie der DNS-Namensraum, hierarchisch organisiert. Ein Server muss nicht alle Domänen kennen, aber er muss weitere Server kontaktieren können, welche für diese über- oder untergeordneten Domänen verantwortlich sind. Eine bestimmte Ebene der Namenshierarchie kann unter verschiedene Server aufgeteilt werden. Der autoritive DNS-

Server verwaltet die Übersetzungstabelle. Ein verantwortlicher Server besitzt entweder selbst diese Übersetzungstabelle oder kennt einen Server, der sie verwaltet.

2.3.3. Abfragetypen

Es gibt drei verschiedene DNS-Abfragetypen:

1. Die „inverse“ DNS-Abfrage:
Hierbei nutzt man den Dienst der speziellen Domäne „in-addr.arpa“. Diese dient zur Auflösung von IP-Adressen in Namen (inverses DNS), wobei jedes Oktett einer IP-Adresse eine Subdomain bildet. Bei einer Anfrage dreht man die Reihenfolge der Bytes der IP-Adresse um und gibt diese mit der Endung „in-addr.arpa“ in einen Browser ein. Als Ergebnis erhält man den Namen dieser Domäne.

2. Die „rekursive“ DNS-Abfrage:
Die Anfrage wird von Nameserver zu Nameserver weitergeleitet, bis der autoritive DNS-Server gefunden ist.

3. Die „iterative“ DNS-Abfrage:
Der Nameserver erhält auf seine Frage jeweils eine Liste mit Nameservern zurück, die er als nächste anfragen muss.

2.3.4. DNS-Caching

Jeder Nameserver nutzt das „Caching“ zur Kostenersparnis und Effizienzsteigerung. Dafür besitzt der Server einen „Cache“ für kürzlich verwendete Namen und Informationen darüber, woher sie stammen bzw. wann sie veralten. Nameserver geben *nonauthoritative answers*[2] und den Namen des Servers zurück, von dem diese Informationen stammen.

2.4. Ressource Record (RR)

Mit jeder Domäne können mehrere Ressourcensätze („Resource Records“) in Verbindung stehen, unabhängig davon, ob es sich um einen Host oder eine Domäne der obersten Ebene handelt. Der übliche Ressourcensatz für einen einzelnen Host ist seine IP-Adresse. Daneben gibt es weitere Arten von Ressourcensätzen. Gibt ein Resolver[3] einen Domänennamen an das

[2] Nonauthorative answers werden von DNS-Servern gesendet, welche nicht selbst die Übersetzungstabelle mit den gewünschten Informationen verwalten.
[3] Bibliothek von Routinen, welche Anfragen von einem Nutzer an einen Nameserver formuliert.

DNS weiter, erhält er Ressourcensätze in Verbindung mit diesen Namen. Die wirkliche Funktion von DNS ist also die Abbildung von Domänennamen auf Ressourcensätze.
Ein Ressourcensatz setzt sich aus fünf Komponenten zusammen. Diese werden zwar der Effizienz halber binär kodiert, jedoch werden Ressourcensätze in den meisten Expositionen als ASCII-Text dargestellt, und zwar je ein Ressourcensatz auf einer Zeile. Das Format lautet wie folgt:

Domain_name Time_to_live Type Class Value

Domain_name bezeichnet die Domäne, auf die sich der Satz bezieht. Normalerweise liegen für eine Domäne viele Sätze vor und jede Kopie in der Datenbank enthält Informationen über mehrere Domänen. Dieses Feld ist der primäre Suchschlüssel für Datenbankabfragen. Die Reihenfolge der Sätze in der Datenbank hat keine Bedeutung. Erfolgt eine Abfrage über eine Domäne, werden alle passenden Sätze der betreffenden Klasse ausgegeben.

Das Feld **Time_to_live** gibt einen Hinweis darauf, wie stabil der Satz ist. Sehr stabile Informationen erhalten einen hohen Wert, z.B. 86400 (Anzahl von Sekunden eines Tages). Sehr flüchtigen Informationen wird ein kleiner Wert zugewiesen, z.B.: 60 (Anzahl von Sekunden einer Minute).

Das Feld **Type** sagt aus, um welche Art von Satz es sich handelt (Siehe Abb.1).

Type	**Bedeutung**	**Wert**
SOA	Start of Authority	Parameter für die betreffende Zone
A	IP-Adresse eines Hosts	32-Bit Ganzzahl
MX	Mail Exchanger	Priorität, in der die Domäne E-Mails annimmt
NS	Name Server	Name eines Servers der betreffenden Domäne
CNAME	Canonical Name	Übersetzt einen Alias[4] der Domäne in den echten Namen
PTR	Pointer	Alias für eine IP-Adresse
HINFO	Host description	CPU und Betriebssystem in ASCII
WKS	Well Known Service	List der Dienste, die der Rechner anbietet
TXT	Text	Uninterpretierter ASCII-Text

Abbildung 1: Typen der Ressource Records

Ein *SOA*-Satz kennzeichnet den Beginn einer Zone. Hier werden die Verwaltungsparameter für die Domäne festgelegt. Der SOA-Satz gibt den Namen der primären Informationsquelle über die Zone des Nameservers, die E-Mail-Adresse des Verwalters, eine eindeutige Seriennummer und verschiedene Flags und Timeouts aus.

Der wichtigste Satztyp ist *A*. Er enthält eine 32 Bit lange IP-Adresse für einen bestimmten Host. Jeder Internet-Host muss mindestens eine IP-Adresse besitzen, damit andere Maschinen mit ihm kommunizieren können. Einige Hosts haben jedoch zwei oder mehr Netzanschlüsse. In diesem Fall besitzen sie einen Ressourcensatz vom Typ A pro Netzanschluss (und damit pro IP-Adresse).

Der zweitwichtigste Satztyp ist *MX*. Er spezifiziert den Namen der Domäne, welche bereit ist, E-Mails für die betreffende Domäne anzunehmen.

Die *NS*-Sätze spezifizieren Nameserver. Jede DNS-Datenbank hat normalerweise einen NS-Satz für jede Domäne der obersten Ebene.

[4] Zusätzlicher Name, mit dem ein bereits anders benannter PC ebenfalls angesprochen werden kann.

CNAME-Sätze ermöglichen das Erstellen von Alias-Namen.

Wie CNAME zeigt *PTR* auf einen anderen Namen. Im Unterschied zu CNAME, bei dem es sich im Grunde nur um eine Makrodefinition handelt, ist PTR ein regulärer DNS-Datentyp, dessen Interpretation vom Kontext abhängt.

Mit *HINFO*-Sätzen kann man herausfinden, um welche Maschine und welches Betriebssystem es sich bei der Domäne handelt.

WKS-Sätze beschreiben eine Liste aller für diesen Rechner zur Verfügung stehenden Dienste.

TXT-Sätze ermöglichen den Domänen, sich selbst auf beliebige Weise zu identifizieren.

Das vierte Feld eines Ressourcendatensatzes ist **Class**. Für Internet-Informationen ist dieses immer „*IN*“.

Im **Value**-Feld soll schließlich der gesuchte Wert gespeichert werden. Hierbei kann es sich um eine Zahl, einen Domänennamen oder eine ASCII-Zeichenkette handeln.

Beispiel für einen RR:

Domain_name	**Time_to_live**	**Type**	**Class**	**Value**
flits.cs.vu.nl.	86400	A	IN	192.31.231.165

Hierbei handelt es sich um eine Internet-Information zum Domänennamen „flits.cs.vu.nl.“. Das Time_to_live-Feld ist auf 86400 (Gültigkeit für einen Tag) gesetzt. Der RR ist vom Typ „A“ und beinhaltet die IP-Adresse 192.31.231.165, unter welcher die Domäne erreichbar ist.

2.5. DNS-Message

Die DNS-Kommunikation wird durch das Wechselspiel zwischen DNS-Request (Resolver) und DNS-Response (Server) charakterisiert. Die Datenpakete sind im DNS-Message Format beschrieben und umfassen die Abschnitte *Header, Question, Answer, Authority und Additional Information Sections* (Siehe Abb.2).

0	15 16	31
IDENTIFICATION	PARAMETER	
No. OF QUESTIONS	No. OF ANSWERS	
No. OF AUTHORITY	No. OF ADDITIONAL	
QUESTION SECTION		
……..		
ANSWERS SECTION		
……..		
AUTHORITY SECTION		
……..		
ADD. INFO SECTION		
……..		

Abbildung 2: DNS Message Format (eine Zeile entspricht 32 Bit)

Wie bei zahlreichen höheren Protokollen bilden ein MAC-Header, ein IP-Header und schließlich der DNS-Header die Grundlage für die DNS-Message.

Der DNS-Header umfasst folgende Felder, welchen jeweils 16 Bit zugeordnet werden:

Identifikation ist zur Antwort-Frage-Zuordnung erforderlich.

In **Parameter** sind verschiedene Teilparameter hinterlegt.

No. Of Questions, Answers, Authority und **Additional** geben die Anzahl der Anfragen, Antworten, Quellen und Zusatzinformationen an.

Die weiteren Felder der DNS-Message sind hauptsächlich für die Formulierung der Anfragen und den Empfang der Antworten zuständig:

Die **Question Section** (Anfrage-Information) beinhaltet den Domänennamen, die Art und die Klasse der Anfrage.

Die **Answer Section** (Antwort-Information) beinhaltet „Ressource Records“ (siehe 2.4.).

Die **Authority Section** (Quellen-Information) enthält den Namen des Servers, der letztlich die Auflösungsinformationen geliefert hat.

Die **Add. Info Section** beinhaltet zusätzliche Informationen.

Jeder Domänenname wird auch in einer DNS-Message als eine Folge von Labels dargestellt. Jedes Label beginnt mit einem Längenbyte und endet mit dem Byte „0“. Um Speicherplatz zu sparen wird ein Längenfeld teilweise als Pointer-Feld genutzt (mehrfaches Auftreten der gleichen Domäne). Es verzweigt dann an die Stelle, wo der Rest der Informationen steht.

2.6. Berkeley Internet Name Domain (BIND)

BIND ist die populärste Implementierung des DNS-Protokolls und besteht aus folgenden drei Komponenten:

- Einem Resolver
- Server Routinen, welche Namen in Adressen auflösen und
- Verschiedene Tools wie NSLOOKUP

3. Sicherheitsaspekte des DNS

Warum ist die Sicherheit in DNS-Servern besonders wichtig und wird doch so vernachlässigt? Das Problem bei den DNS-Servern ist, dass sie keine Daten enthalten, die man im eigentlichen Sinn als besonders schützenswert betrachtet. Der Server wird lediglich als ein triviales „Telefonbuch“ betrachtet. Dass ohne dieses Telefonbuch allerdings kein einziges Byte der gut gesicherten Daten auf irgendeinem Server zu erreichen ist, wird dabei oft nicht bedacht.

Neben der Erreichbarkeit ist die Verlässlichkeit ein weiterer wichtiger Faktor. Bei allen Verbindungen innerhalb des Internets muss sich der Initiator sicher sein können, dass die ihm übermittelten Informationen auch die sind, welche er erwartet. Sollte es einem Angreifer gelingen, die DNS-Informationen zu seinen Gunsten zu verfälschen, könnte er damit beträchtlichen Schaden anrichten. Wird zum Beispiel ein Anwender mittels einer gefälschten IP-Adresse auf eine andere Seite geführt, kann dies zu großen Problemen wie dem Verlust sensitiver Daten und Passwörter führen.

4. Sicherheitslücken

Im Folgenden werden Schwachstellen des Konzeptes erläutert, welche einen bewussten Angriff ermöglichen oder erleichtern.

4.1. Zone Transfers

Bei einem Zone Transfer gibt der primäre DNS-Server seine Datenbank an einen sekundären Nameserver weiter. Dieses dient zur Entlastung des primären DNS-Servers. Wird nach einem Zone Transfer ein sekundärer Nameserver durch einen Angreifer angefragt, kann er die Informationen des primären DNS-Servers erlangen. Daher dürfen keine sicherheitsrelevanten Informationen (wie Hosts), welche später für Angriffe auf Hosts dieser Domäne genutzt werden können, mit sensitiven Daten an den sekundären Nameserver übergeben werden.

4.2. DNS-Spoofing / Cache Poisoning

DNS-Spoofing, oder auch Cache Poisoning genannt, nutzt falsch konfigurierte Server für seinen Angriff. Ein Hacker mit Authority-Zugriff zu einer Zone kann eine rekursive DNS-Anfrage nach einem Host in seiner Zone starten. Wenn der Zielserver den DNS-Server des Hackers befragt, kann dieser in seiner Antwort RRs mit falschen und unnützen Informationen zurückliefern, welche anschließend im Cache des anfragenden Servers gespeichert werden.

4.3. Masquerading

In diesem Fall versucht der Angreifer, sich als jemand anderes auszugeben. Wenn nach einem Cache Poisoning Angriff ein RR eine falsche IP-Adresse enthält, die den Nutzer auf eine andere Website führt, nennt man das „Masquerading“.

4.4. Denial of Service (DoS)

Ziel des „Denial of Service“-Angriffes ist es, einem Nutzer den Zugriff auf einen bestimmten Service eines Servers zu verwehren. Diese Form betrifft dabei keinesfalls ausschließlich DNS-Server. Der Bereich der „Denial of Service“ – Angriffe ist ein sehr weites Feld und eine eigene Ausarbeitung wert. Wichtig sind folgende zwei Bereiche:

1. Absturz des Servers: Ziel ist es hierbei, den Server durch Absturz zu deaktivieren.
2. Belegung der Bandbreite: Die Kommunikationswege des Servers werden durch eine Flut von Daten überlastet, so dass der Server externe Anfragen nicht mehr bearbeiten kann.

5. Maßnahmen zur Sicherung des DNS

Wie jedes Computersystem, welches sensitive Informationen enthält, ist es auch beim DNS wichtig, den physikalischen Zugriff auf den DNS-Server restriktiv zu handhaben. Grundsätzlich ist der Zugriff mit Passwörtern zu schützen und die Server sind regelmäßig zu überwachen.

Die Methoden, welche bei den oben erwähnten Angriffen verwendet werden, können mit einer korrekten Zonen-Aufteilung der Nameserver vermieden werden. Für die Organisation, die ihre Domäne sowohl gegen Cache Poisoning, als auch gegen die ungewollte Verbreitung sicherheitsrelevanter Informationen mittels Zone Transfers schützen möchte, sind mindestens zwei Nameserver notwendig:

Einer, der DNS-Anfragen über seine Zone vom Internet beantwortet und auf rekursive Fragen nicht eingeht, und ein Anderer, welcher die restlichen Anfragen (aus seiner Zone kommend ans Internet) rekursiv beantwortet.

Eine weitere Maßnahme, um Zone Transfers nur an Hosts, die für einen Zone Transfer gewünscht sind, zu erlauben, ist die Installation einer aktuelleren Version von „BIND“[5] (siehe [6]), welches diese Option und die entsprechende Konfiguration unterstützt. Mit dem BIND-tool „dig“ kann ein DNS auf seine Sicherheit untersucht werden.

Sind zwei DNS-Server wie oben beschrieben vorhanden, kann der Nameserver, welcher für Anfragen von seiner Zone ans Internet zuständig ist, von der Firewall geschützt werden, indem UDP-Zugriffe vom Internet auf den Port 53 des Nameserver nur dann erlaubt sind, wenn zuvor vom Nameserver eine Anfrage „nach außen“ an einen Nameserver verschickt wurde.

[5] Version: 4.9.11, 8.2.7, 8.3.4 oder 9

Anhang A: Abbildungsverzeichnis

Anhang B: Quellenverzeichnis

Literatur:

[1] Andrew S. Tanenbaum „Computernetze“, Verlag: Pearson Studium, 3.Auflage, 2000

[2] Gerhard Lienemann: „TCP/IP-Grundlagen: Protokolle und Routing“, Verlag: Heise, 2. Auflage, 2000

[3] P. Mockapetris RFC 1034: CONCEPTS AND FACILITIES, 1987

[4] P. Mockapetris RFC 1035: IMPLEMENTATION AND SPECIFIKATION, 1987

[5] Skript zur Vorlesung „Kommunikationsnetze“ (Prof. Dr. Schwenk)

Internetadressen:

[6] http://www.golem.de/0211/22635.html

[7] http://www.linuxinfo.de/de/db/lnxi-cont.php3?start=12&step=12&bereich=doku

[8] www.linuxfibel.de/dns_srv.htm

[9] www.netplanet.org/adressierung/dns.shtml

[10] www.selflinux.org/selflinux/html/dns.html

[11] www.net.informatik.tu-muenchen.de/teaching/WS02/security/securityUeb/09Aausarbeit.pdf

[12] http://www.ifi.unizh.ch/ikm/Vorlesungen/Sem_Sich01/Daetwyler.pdf

[13] http://www.fz-juelich.de/zam/mathe/tm/information/net/NetSecMan2003-DNS.pdf

[14] http://www.ietf.org/rfc/rfc1034.txt

[15] http://www.ietf.org/rfc/rfc1035.txt