Segurança no DNS

Rafael R. Obelheiro
rro@das.ufsc.br

19/12/2002
Revisão 1.1.1.1

Resumo

Este documento discute aspectos de segurança do DNS (*Domain Name System*) e as especificações DNSSEC.

1 Introdução

Nos últimos anos, tornou-se visível o rápido crescimento da Internet. Com essa popularização, a rede, antes eminentemente acadêmica, passou a ser usada para uma infinidade de aplicações e transações comerciais, criando-se toda uma indústria em torno dela. Essa mudança na utilização da Internet impôs novos requisitos aos protocolos que formam o seu sustentáculo. O fato dela ter sido capaz de se adaptar às novas demandas é um testemunho da robustez desses protocolos, a maioria deles desenvolvidos muito tempo antes da grande explosão de crescimento da rede. Entretanto, é inevitável que alguns protocolos não consigam atender a todos os requisitos que hoje se impõem, uma vez que foram projetados em um contexto radicalmente diferente do atual.

O DNS (*Domain Name System*) é o sistema distribuído que implementa o serviço de nomes da Internet [Moc87a, Moc87b]. Ele permite que os recursos disponíveis na rede sejam localizados através de nomes inteligíveis para seres humanos, o que o coloca como um dos protocolos de aplicação mais importantes hoje em dia. Sem resolução de nomes, a Internet torna-se bastante inconveniente para os usuários. Apesar disso, o DNS é suscetível a uma série de ameaças de segurança, muitas delas imprevisíveis na época em que o protocolo foi concebido.

Face à relativa fragilidade do DNS, a IETF (*Internet Engineering Task Force*), a organização que define os padrões usados na Internet, está desenvolvendo e propondo medidas para melhorar a segurança do protocolo. Estas propostas são conhecidas, no seu conjunto, como DNSSEC (*DNS Security Extensions*).

Neste documento, os fundamentos do DNS e uma análise das ameaças às quais ele está sujeito são apresentados na seção 2. Esse conhecimento é indispensável para a compreensão das especificações DNSSEC, discutidas na seção 3.

2 DNS

Um conceito fundamental em sistemas distribuídos é o de **nome**, algo utilizado para identificar recursos no sistema e permitir a sua localização. Por exemplo, na Internet um *host* é identificado por um endereço IP, que é um número de 32 bits. Nomes formados exclusivamente por números (tais como endereços IP) são convenientes para os computadores que compõem o sistema distribuído, mas não o são para os seus usuários, ainda que existam formas simplificadas de representá-los textualmente (tais como `192.0.2.1`). Isso leva à necessidade de nomes que sejam inteligíveis para os usuários e de um mecanismo para mapear esses nomes inteligíveis nos endereços de baixo nível usados para a comunicação no sistema.

Nos primórdios da Internet, o mecanismo utilizado era um arquivo (chamado `HOSTS.TXT`) que continha todos os nomes de *hosts* da Internet e seus respectivos endereços. Esse arquivo era mantido, de

forma centralizada, pelo SRI-NIC (*Stanford Research Institute Network Information Center*), e distribuído a todos os outros *hosts* na Internet através de protocolos de transferência de arquivos [MD88].

Esse primeiro mecanismo possuía, claramente, sérios problemas de escalabilidade. De fato, à medida em que a Internet foi crescendo, tornou-se impraticável manter todas as associações nome-endereço de forma centralizada, em um único arquivo. Era necessária uma solução melhor para o problema da resolução de nomes, e a resposta encontrada foi o DNS, adotado a partir de 1984. A especificação original do DNS é descrita pelas RFCs 882 e 883 [Moc83a, Moc83b]. Essa especificação foi revisada em 1987, dando origem às RFCs 1034 e 1035 [Moc87a, Moc87a]. Alguns aspectos das RFCs 1034 e 1035 foram atualizados desde então (mais notadamente pela RFC 2136 [VTRB97] e pela RFC 2181 [EB97]), mas o cerne da especificação permanece o mesmo.

A apresentação do DNS nesta seção concentra-se nos aspectos principais do protocolo, omitindo os detalhes que não têm relação com o restante deste trabalho. Informações mais completas podem ser encontradas nas RFCs citadas acima e em [AL01], uma obra que discute extensivamente a configuração e operação de servidores DNS.

2.1 Critérios de Projeto do DNS

Durante o desenvolvimento do DNS, diversos critérios foram identificados como essenciais em um sistema de nomes para a Internet [Moc87a]:

- **Consistência**: Os recursos deveriam ser identificados de forma consistente, sem depender de informações de roteamento ou topologia de rede.
- **Eficiência**: O sistema deveria ser eficiente. As formas encontradas para permitir isso foram a hierarquização do espaço de nomes e o uso intensivo de *caching* local.
- **Natureza distribuída**: Cada rede deveria ser responsável pela administração da sua porção do espaço de nomes. Essa natureza distribuída envolve também o uso de múltiplos servidores para prover um grau aceitável de disponibilidade do serviço DNS.
- **Generalidade**: O sistema deveria ser geral o suficiente de forma a não impor restrições desnecessárias sobre o seu propósito nem sobre a sua aplicabilidade.
- **Independência**: O sistema deveria ser independente não só de plataforma de *hardware* e *software* como também do sistema de comunicação utilizado.

2.2 Componentes do DNS

O DNS é formado por diversos componentes: espaço de nomes, registros de recurso, mensagens DNS, clientes (resolvedores) e servidores DNS. Esta seção apresenta cada um destes componentes e discute o papel que eles desempenham no DNS.

2.2.1 O Espaço de Nomes do DNS

O **espaço de nomes** do DNS é hierárquico, estruturado em forma de árvore invertida. A raiz da árvore é o **domínio raiz**, e seus filhos são os domínios de primeiro nível, que podem conter vários níveis de subdomínios, conforme pode ser visto no exemplo da figura 1. Um **nome de domínio completo** (também conhecido como FQDN, de *fully qualified domain name*) é representado pela concatenação dos rótulos de cada nó da árvore, desde a folha (que representa a entidade nomeada) até a raiz. Um ponto é usado como separador entre os rótulos, sendo que o domínio raiz é representado por um rótulo vazio. Um domínio é uma sub-árvore desse espaço de nomes. No exemplo, `das.ufsc.br.` é um nome de domínio. O ponto ao final do nome separa o domínio `br` do domínio raiz (rótulo vazio); muitas vezes ele é omitido. Como mostra a figura 1, o domínio raiz, quando isolado, também é representado por um ponto.

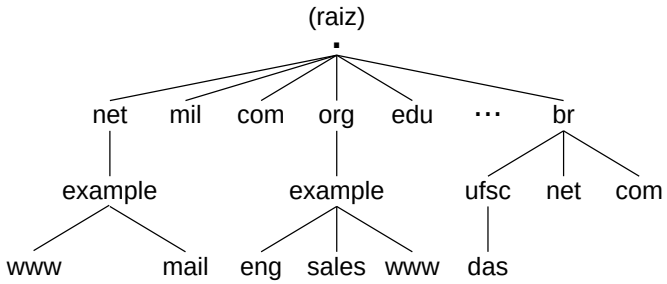

Figura 1: Espaço de nomes do DNS

A porção do espaço de nomes administrada por um servidor de nomes é chamada de **zona**. Quando um servidor de nomes controla todos os nomes de um domínio, a zona e o domínio são idênticos. Entretanto, o mais comum é que algumas porções de um domínio sejam delegadas para outros servidores, caso em que a zona contém apenas as partes não delegadas desse domínio. Voltando ao exemplo da figura 1, pode-se assumir, hipoteticamente, que o domínio `eng.example.org` é controlado por um servidor de nomes diferente do que controla o domínio `example.org`, de modo que ele não faz parte da zona `example.org`. Por sua vez, `sales.example.org` não é administrado de forma independente do domínio `example.org` e, portanto, faz parte da zona `example.org`, conforme ilustra a figura 2.

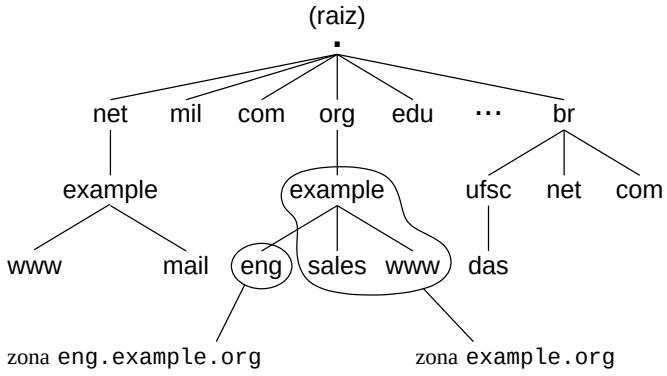

Figura 2: Divisão de domínios em zonas

2.2.2 Registros de Recurso (RRs)

Um **registro de recurso** (*resource record*), ou simplesmente **RR**, é um item de dados associado a um elemento da árvore DNS. Um RR é identificado por uma tripla $< nome, tipo, classe >$. O **nome** determina a sua localização na árvore DNS. O **tipo** determina qual a espécie de informação codificada no RR. Os principais tipos estão listados na tabela 1.[1]

Tipo	Definição	Utilização
`A`	endereço	mapeia um nome em um endereço
`NS`	servidor de nomes	designa o servidor de nomes com autoridade sobre uma zona
`SOA`	início de autoridade	define diversos parâmetros administrativos de uma zona
`PTR`	ponteiro	mapeia um endereço em um nome
`CNAME`	nome canônico	define um *alias* (apelido) para um nome
`MX`	servidor de *mail*	especifica o servidor que deve receber *mail* endereçado a um *host* ou domínio

Tabela 1: Principais tipos de RRs

[1] A alocação de tipos, classes e outros valores do DNS é feita pela IANA (*Internet Assigned Numbers Authority*). A tabela de alocação corrente pode ser obtida em `http://www.iana.org/assignments/dns-parameters`.

A **classe** possibilita que RRs de mesmo tipo tenham formatos distintos caso se refiram a famílias de protocolos diferentes. Por exemplo, todos os protocolos de rede possuem a noção de endereço (registros `A`), mas o formato dos endereços podem diferir bastante de um protocolo para o outro. As classes atualmente definidas são mostradas na tabela 2. Na Internet se utiliza apenas a classe `IN`.

Classe	Utilização
`IN`	Internet
`CH`	CHAOS
`HS`	Hesiod

Tabela 2: Classes de RRs

O formato de um registro de recurso é definido de acordo com o mostrado na figura 3. Além de `NAME`, `TYPE` e `CLASS` (que correspondem, respectivamente, a nome, tipo e classe), um RR possui os seguintes campos [Moc87b]:

- `TTL` (*time-to-live*): um campo de 32 bits que determina por quantos segundos um RR pode ser armazenado em *cache* (o papel dos *caches* no DNS será detalhado na seção 2.3.3).
- `RDLENGTH`: um campo de 16 bits que determina o tamanho em bytes de `RDATA`.
- `RDATA`: uma seqüência de bytes que descreve o recurso; o formato de `RDATA` varia de acordo com `TYPE` e `CLASS`.

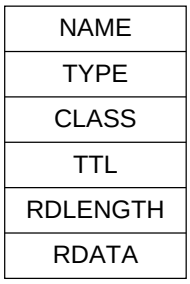

Figura 3: Formato de um RR

A figura 4 mostra alguns exemplos de RRs para uma zona fictícia `example.org`. Por esses exemplos é possível perceber que RRs de tipos distintos possuem diferentes formatos de informação: registros `A` carregam endereços IP, registros `NS` carregam nomes e registros `MX` carregam um nome e uma preferência[2] (`10` neste exemplo).

```
example.org.              IN   NS      ns.example.org.
alpha.example.org.        IN   A       192.0.2.1
www.example.org.          IN   CNAME   beta.example.org.
example.org.              IN   MX  10   alpha.example.org.
1.2.0.192.in-addr.arpa.   IN   PTR     alpha.example.org.
```

Figura 4: Exemplos de RRs para a zona `example.org`

O conjunto de RRs com o mesmo nome, tipo e classe porém dados (`RDATA`) distintos é chamado **RRset** [EB97]. A figura 5 mostra um exemplo de RRset para $<$ `example.org`, `IN`, `NS` $>$.

[2]Um nome pode ter vários registros `MX`, cada qual com uma preferência; os *hosts* com menor preferência são contactados primeiro para a entrega de *email*.

```
example.org.    IN      NS      ns.example.org.
example.org.    IN      NS      ns.example.net.
```

Figura 5: Exemplo de RRset

2.2.3 Mensagens DNS

Mensagens DNS são os elementos de dados trocados entre clientes e servidores DNS, e são definidas pela RFC 1035 [Moc87b]. O formato de uma mensagem DNS é mostrado na figura 6. Ela contém um cabeçalho (*Header*) e até quatro seções: pergunta (*Question*), resposta (*Answer*), autoridade (*Authority*) e adicional (*Additional*).

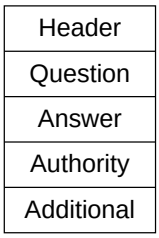

Figura 6: Formato de uma mensagem DNS

O cabeçalho possui vários campos, dentre os quais:

- `ID`, um identificador de 16 bits criado pelo programa que realiza uma consulta DNS;
- `OPCODE`, um valor de 4 bits que sinaliza qual o código da operação;
- `QR`, um *flag* de 1 bit que indica se a mensagem é uma consulta (`QR`=0) ou uma resposta (`QR`=1);
- `AA`, um *flag* de 1 bit usado apenas em respostas para indicar que o servidor de nomes que está respondendo possui autoridade sobre o domínio consultado;
- `TC`, um *flag* de 1 bit usado para indicar se a mensagem foi truncada;
- `RD`, um *flag* de 1 bit que indica se quem realiza a consulta deseja recursão;
- `RA`, um *flag* de 1 bit que indica se o servidor oferece recursão;
- `RCODE`, um valor de 4 bits que sinaliza o código de resposta do servidor;
- quatro campos de 16 bits indicando o número de entradas nas seções de pergunta, resposta, autoridade e adicional que se seguem.

As quatro seções que se seguem ao cabeçalho se apresentam sempre na mesma ordem, embora em uma mensagem possa haver seções vazias.

A seção de pergunta contém o nome (`QNAME`), o tipo (`QTYPE`) e a classe (`QCLASS`) que definem o objeto de uma consulta. Os tipos válidos em uma consulta são todos aqueles definidos para um registro de recurso (como os listados na tabela 1), além de alguns adicionais, como `*` (que significa todos os tipos) e `AXFR` (que sinaliza uma transferência de zona).[3] Os valores válidos para `QCLASS` são aqueles da tabela 2 e mais `*`, que representa todas as classes.

A seção de resposta contém o RRset que responde diretamente à consulta. A seção de autoridade contém RRs que descrevem outros servidores com autoridade sobre o domínio consultado. Quando um servidor de nomes tenta resolver um nome e descobre que existe um servidor com autoridade sobre o domínio ao qual este nome pertence, ele coloca o nome deste servidor na seção de autoridade da mensagem DNS.

[3] Transferências de zona são discutidas na seção 2.2.5.

A seção adicional contém registros de recurso que não foram explicitamente solicitados mas que muito provavelmente serão necessários caso RRs contidos nas outras seções venham a ser utilizados. Um desses casos é quando o nome de um servidor é incluído na seção de autoridade (conforme descrito no parágrafo acima): se o servidor que está montando a resposta souber, além do nome, o(s) endereço(s) do servidor com autoridade, ele deve incluir este(s) endereço(s) na seção adicional.

Exemplo. Supõe-se que se deseja descobrir o endereço IP de `alpha.example.org`, que vem a ser `192.0.2.1`. A figura 7(a) mostra a mensagem DNS com a consulta, e a figura 7(b) mostra a mensagem DNS com a resposta (o formato dos RRs retornados foi simplificado, com a omissão dos campos `TTL` e `RDLENGTH`). Este exemplo ilustra o preenchimento das seções de autoridade e adicional em respostas DNS.

Header	`ID=123`
Question	`QNAME=alpha.example.org.`, `QTYPE=A`, `QCLASS=IN`
Answer	
Authority	
Additional	

(a) Consulta

Header	`ID=123`
Question	
Answer	`alpha.example.org. IN A 192.0.2.1`
Authority	`example.org. IN NS ns.example.org.` `example.org. IN NS ns.example.net.`
Additional	`ns.example.org. IN A 192.0.2.3` `ns.example.net. IN A 10.4.8.25`

(b) Resposta

Figura 7: Exemplos de mensagens DNS

2.2.4 Clientes DNS

Os clientes no DNS são os chamdos resolvedores (*resolvers*), que são os componentes de *software* responsáveis por montar consultas DNS e processar as respostas dos servidores. Muitas vezes o resolvedor é implementado como um conjunto de funções de biblioteca, e as aplicações que necessitam do serviço de nomes então fazem uso dessa biblioteca.

Tipicamente, um resolvedor oferece as seguintes funcionalidades:

- resolução de um nome em um endereço;
- resolução de um endereço em um nome;
- consultas DNS genéricas.

Conforme será visto em detalhes na seção 2.3, a resolução de nomes no DNS é um processo que pode envolver múltiplas interações entre resolvedor e servidor DNS para se chegar à resposta solicitada por uma aplicação. Isso leva à classificação dos resolvedores em duas categorias:

1. **Resolvedor completo** (*full resolver*): aquele que faz todas as consultas e processa todas as respostas intermediárias para obter a resposta final desejada pela aplicação.
2. **Resolvedor *stub*** (*stub resolver*): aquele que monta a consulta inicial de acordo com a requisição da aplicação, repassa-a a um servidor de nomes previamente configurado para que este faça o processamento da consulta, processa a resposta final enviada por este servidor e a repassa à aplicação.

2.2.5 Servidores DNS

Para falar de servidores DNS, é importante resgatar o conceito de zona definido na seção 2.2.1. Uma zona é a porção do espaço de nomes administrada por um servidor DNS. Um servidor de nomes responsável por um determinada zona é dito **com autoridade** (*authoritative*) sobre essa zona. A divisão de subdomínios em zonas é feita através da **delegação** do servidor com autoridade sobre o domínio pai para os servidores com autoridade para o subdomínio delegado.

Voltando ao exemplo da figura 2, o servidor com autoridade sobre o domínio `org` delegou o subdomínio `example.org` ao servidor `ns.example.org`; este servidor, por sua vez, pode delegar o subdomínio `eng.example.org` para outro servidor (por exemplo, `ns.eng.example.org`, que teria então autoridade sobre a zona `eng.example.org`.[4]

Uma zona deve possuir mais de um servidor DNS respondendo com autoridade sobre ela. Isto é feito para garantir um mínimo de replicação dos dados da zona e permitir que o serviço DNS continue disponível mesmo com a falha de um ou mais servidores. O servidor principal (aquele que possui a versão original dos dados da zona) é chamado de **mestre** ou **primário**, enquanto que os servidores de *backup* são chamados de **escravos** ou **secundários**. É recomendável que pelo menos um dos servidores escravos esteja localizado de forma a que uma falha no mestre (tal como uma pane de energia ou perda de conectividade) dificilmente o afete também.

A sincronização entre servidores mestres e escravos é feita através de uma operação de **transferência de zona**. Na forma mais convencional, um secundário envia ao mestre, de forma periódica, uma consulta pelo registro `SOA`, que contém, entre outras informações, um número de série. O secundário compara o número de série no `SOA` do mestre com o que ele tem armazenado em *cache*; se o número do mestre for maior, o secundário inicia a transferência de zona, enviando ao mestre uma consulta com `QTYPE=AXFR`.

A RFC 1996 [Vix96] define um mecanismo para que o servidor mestre possa notificar os escravos quando uma zona foi alterada e deve ser atualizada. Esse mecanismo (chamado de NOTIFY) permite reduzir a janela de tempo onde os dados de zona servidos pelos escravos são diferentes dos dados servidos pelo mestre.

2.3 Resolução de Nomes

A **resolução de nomes** é o processo pelo qual um *host* obtém determinadas informações DNS sobre um dado nome. Conforme mostrado na seção 2.2.4, ela envolve um resolvedor e um ou mais servidores DNS, e pode ser feita de duas maneiras, iterativa ou recursiva. Esta mostra como é esse processo, qual a diferença entre iteração e recursão, qual o papel dos *caches* locais na resolução de nomes e como o DNS pode ser usado para resolver um nome a partir de um endereço IP (a chamada resolução reversa).

2.3.1 Iteração

A **iteração**, ou resolução iterativa de nomes, é quando um resolvedor completo interage com todos os servidores que devem ser consultados para obter a resposta solicitada por quem o invocou. Ela pode ser melhor explicada através de um exemplo. Supõe-se que o *host* `alpha.example.org` (que possui um resolvedor completo) deseja saber o endereço IP de `foo.example.net`, sem conhecer o endereço do servidor de nomes para a zona `example.net`.

A figura 8 mostra como isso é feito. Todos os resolvedores completos são pré-configurados com os endereços dos servidores para o domínio raiz. Assim, em (1), `alpha` envia para um desses servidores uma consulta pelo registro `A` de `foo.example.net`. O servidor raiz (que não possui autoridade sobre a zona `example.net`) não pode responder a esta consulta, mas ele responde com o RRset `NS` do domínio `net`, juntamente com o RRset `A` destes servidores. Em (2), o resolvedor reenvia a consulta, desta vez para um dos servidores referidos pelo servidor raiz; o servidor do domínio `net` responde com o endereço dos servidores com autoridade para `example.net`. Finalmente, em (3) o resolvedor envia a consulta para o servidor de `example.net` e obtém como o resposta o RRset `A` de `foo.example.net`.

[4] Os servidores DNS não estão mostrados na figura.

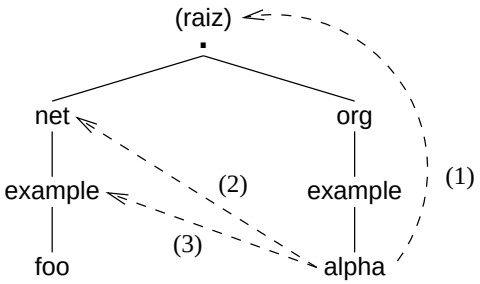

Figura 8: Resolução iterativa de nomes

2.3.2 Recursão

Na prática, poucos *hosts* em uma dada rede possuem um resolvedor completo, capaz de resolver um nome de forma iterativa. A maioria dos sistemas possui apenas um resolvedor *stub*, que se limita a montar a consulta inicial (por exemplo, pelo endereço de `foo.example.net`) e processar a resposta final. O processo no qual um servidor de nomes (que implementa um resolvedor completo) recebe uma consulta de um resolvedor *stub* e obtém uma resposta para ela é chamado **recursão**, ou resolução recursiva de nomes.

Na recursão, o servidor de nomes pode tanto resolver a consulta de forma iterativa quanto repassá-la para outro servidor recursivo. Normalmente, cada rede possui pelo menos um servidor capaz de atender a consultas recursivas.

Voltando ao exemplo da seção 2.3.1, supõe-se agora que `alpha.example.org` possui apenas um resolvedor *stub*, e que a rede `example.org` possui um servidor de nomes recursivo local. O processo de resolução recursiva é ilustrado na figura 9. A única diferença em relação ao processo iterativo da figura 8 são os passos (1) e (5), onde `alpha` se comunica com seu servidor de nomes local (que é quem interage com os outros servidores que precisam ser contactados).

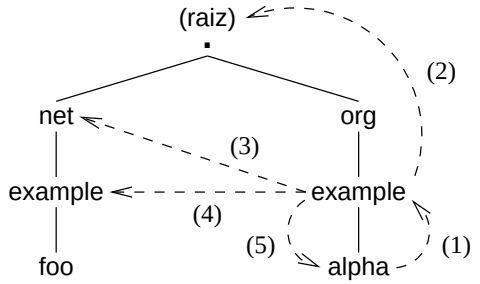

Figura 9: Resolução recursiva de nomes

2.3.3 *Caching*

Os processos de resolução de nomes mostrados nas figuras 8 e 9 não levam em conta a existência de *caches*. Conforme foi mencionado na seção 2.1, uma das chaves para se alcançar eficiência no DNS é o uso intensivo de *caches* locais.

2.3.4 Resolução Reversa

Como a árvore DNS da figura 1 é ordenada por nomes, ela permite descobrir facilmente o endereço correspondente a um dado nome. O mapeamento de um endereço em um nome, por sua vez, não é direto (uma busca exaustiva na árvore definitivamente não é uma opção). Para resolver este problema, adotou-se um artifício: foi reservado um domínio (chamado `in-addr.arpa`) cujos subdomínios são octetos de um endereço IP. Por exemplo, o endereço IP `192.0.2.1` é representado pelo nome `1.2.0.192.in-addr.arpa`. A árvore `in-addr.arpa` pode ser considerada uma árvore paralela à do DNS [Bel95].

2.4 Ameaças ao DNS

3 As Especificações DNSSEC

Referências

[AL01] Paul Albitz and Cricket Liu. *DNS & BIND*. O'Reilly, Sebastopol, CA, USA, 4th edition, 2001.

[Bel95] Steven M. Bellovin. Using the Domain Name System for system break-ins. In *Proceedings of the Fifth USENIX UNIX Security Symposium*, pages 199–208, Salt Lake City, UT, USA, June 1995.

[EB97] Robert Elz and Randy Bush. Clarifications to the DNS specification. RFC 2181, July 1997.

[MD88] Paul V. Mockapetris and Kevin J. Dunlap. Development of the Domain Name System. In *Proceedings of the ACM SIGCOMM'88*, pages 123–133, Stanford, CA, USA, August 1988.

[Moc83a] Paul V. Mockapetris. Domain names—concepts and facilities. RFC 882, November 1983.

[Moc83b] Paul V. Mockapetris. Domain names—implementation and specification. RFC 883, November 1983.

[Moc87a] Paul V. Mockapetris. Domain names—concepts and facilities. RFC 1034, November 1987.

[Moc87b] Paul V. Mockapetris. Domain names—implementation and specification. RFC 1035, November 1987.

[Vix96] Paul Vixie. A mechanism for prompt notification of zone changes (DNS NOTIFY). RFC 1996, August 1996.

[VTRB97] Paul Vixie, Susan Thomson, Yakov Rekhter, and Jim Bound. Dynamic updates in the Domain Name System (DNS UPDATE). RFC 2136, April 1997.