CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DA PARAÍBA
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE

Extensões de segurança para o DNS

George Luiz Cardoso Buriti

João Pessoa
2006

George Luiz Cardoso Buriti

Extensões de segurança para o DNS

Trabalho apresentado como requisito parcial para aprovação no Curso de Especialização em Gestão da Segurança da Informação do Centro Federal de Educação Tecnológica da Paraíba.

Orientador: Prof.Dr. Dênio Mariz Timóteo de Sousa

João Pessoa
2006

George Luiz Cardoso Buriti

Extensões de segurança para o DNS

TRABALHO DE CONCLUSÃO DE CURSO APROVADO EM __ / __ / __

_____
PROF.DR. DÊNIO MARIZ TIMÓTEO DE SOUSA
Orientador

_____
PROF.DR. JOSÉ ANTÃO BELTRÃO MOURA
Coordenador pela UFCG

_____
PROF.MSC. RICARDO LIMA E SILVA
Coordenador pelo CEFETPB

João Pessoa
2006

Dedicatória

Agradecimentos

Resumo

O Sistema de Nomes de Domínios (*Domain Name System* - DNS) é um dos principais mecanismos que compõe a Internet. Funciona como um repositório distribuído que permite o mapeamento entre nomes de serviços e sua correspondente localização na rede. O DNS tem algumas vulnerabilidades conhecidas e se exploradas, podem trazer sérias conseqüências para os serviços da Internet, tais como negação de serviço e roubo de informações. O DNSSec, um conjunto de extensões de segurança proposto pelo IETF para o DNS, utiliza criptografia de chaves públicas e assinaturas digitais, disponibiliza autenticação e integridade dos dados, além de distribuição de chaves e certificação de transações e requisições. Este trabalho tem como objetivo discutir o funcionamento, os problemas e as vulnerabilidades conhecidas do DNS, bem como discutir como o DNSSec se propõe a minimizar esses problemas. Uma análise crítica sobre os potenciais problemas do DNSSec e sobre as dificuldades da sua adoção em escala na Internet também serão discutidos.

Palavras chave: DNS, DNSSec, Internet, vulnerabilidade, ataques, criptografia, assinatura digital, chaves públicas.

Abstract

The Domain Name System (DNS) is one of the main components of the Internet infrastructure. It works as a distributed repository and query system that allows applications mapping between service names and their correspondent network address. However, DNS has some well known vulnerabilities which, if fully exploited, can bring serious problems to the Internet services and applications, such as denial of service attacks, eavesdropping and thief of information. DNSSec is a new proposed standard from IETF, which extends DNS to use public key cryptography, key distribution and digital certificates, enabling it to offer authentication, confidentiality, data integrity and non-repudiation. This work is aim at presenting how DNS works, its main problems and vulnerabilities. We also present a discussion about what solutions DNSSec is offering to DNS problems, what are the main difficulties for deploying it in large scale over the Internet , and a critical analysis of its potential problems.

Keywords: DNS, DNSSec, Internet, vulnerabilities, attacks, cryptography, public key infrastructure.

Lista de ilustrações

Lista de tabelas

Sumário

1. Introdução

Um estudante quer fazer uma pesquisa na Internet. A maneira mais fácil é acessar algum *site* de buscas e realizar sua pesquisa. Então, o estudante digita o nome do *site* em seu navegador, por exemplo www.google.com.br, e assim chega até o *site* de buscas. Este mesmo estudante está acessando a Internet através de uma rede de uma universidade, e deseja trocar arquivos com seu colega que está em um outro PC. Basta digitar o nome do PC com o qual ele deseja se conectar, através de um serviço de FTP (*File Transfer Protocol*), por exemplo, e iniciar a transferência do arquivo.

É bem simples assim acessar algum *site* ou máquina remota a partir de uma máquina local. Basta digitar o nome do destino, como algum *site* da Web ou um *host* em alguma rede para realizar conexões entre dois computadores. Mas o que acontece por trás dessa facilidade é um processo bem mais complexo, que será explicado ao longo deste trabalho.

“Em uma rede baseada no protocolo TCP/IP, cada *host* é identificado por um endereço numérico de 32 bits, que são separados em quatro octetos e cada octeto é representado normalmente em números decimais. Os quatro números são separados por um caractere ponto (“.”). Quatro números decimais são bem mais fáceis de lembrar do que trinta e dois zeros e uns, como números de telefones, e fica muito difícil para uma pessoa lembrar de muitos endereços IP´s sem precisar de algum diretório de assistência. Esse diretório nomeia nomes de *hosts* em endereços IP [1]”.

Na verdade, ao acessar uma página na Internet ou um sistema remoto,a aplicação que fez a conexão inicial precisa conhecer o endereço IP da página na Internet ou do sistema de destino, pois as informações serão enviadas em pacotes IP, que tem um campo de origem e destino que devem ser números IP e não "nomes". Acontece que a aplicação que faz essa requisição utiliza um sistema que associa o endereço IP a um nome, tornando o acesso à página na Internet possível através do nome dela e não do endereço IP. Este processo chama-se *resolução de nomes*, e é realizado por um sistema chamado *Domain Name System* (Sistema de Nomes de Domínio), ou DNS.

“O DNS, ou Sistema de Nomes de Domínio, é uma das bases de sustentação da teia que compõe a Internet moderna. Ele realiza uma função simples e direta: mapear nomes para endereços IP e vice-versa. Quando o sistema não está funcionando, as conseqüências são graves e generalizadas [2]”.

Alguns dos problemas causados pelas falhas do DNS podem levar à poluição do *cache* do servidor DNS, redirecionando as consultas feitas ao servidor DNS (*cache poisoning*), à indisponibilidade do servidor DNS, causadas por várias consultas recursivas (s*erver compromising*) e à falsificação do IP na resolução de nomes DNS (*DNS spoofing*) [9]. Estas vulnerabilidades serão discutidas neste trabalho.

Estes problemas podem ser causados por intrusos, que se aproveitam das falhas e vulnerabilidades do sistema para enganar as pessoas ou para parar o funcionamento do sistema. O objetivo deste trabalho é explicar o funcionamento do DNS, seus componentes, suas falhas e vulnerabilidades, e apresentar uma solução que está atualmente em desenvolvimento, o que ela pode e o que ela não pode resolver, bem como seus componentes e seu funcionamento: o DNSSEC.

DNSSEC é o nome dado às extensões de segurança que estão sendo propostas para o protocolo DNS definidas pelas RFC´s (*Request for Comments*) 4033, 4034 e 4035, que estabelecem os detalhes das alterações propostas, para a implantação de um DNS seguro. “A característica principal deste novo modelo do DNS é prover a autenticação da origem dos dados das zonas (partes do espaço de nomes, que será explicado mais adiante) e a integridade destes dados usando criptografia de chave pública [1]”.

2. O Sistema de Nomes de Domínios (DNS)

Durante a década de 1970, a ARPAnet (*Advanced Research Projects Agency Network*), uma área experimental de redes de computadores que conectava importantes organizações de pesquisa dos Estados Unidos, criado pelo Departamento de Defesa norte-americano (mais tarde DARPA), formava uma pequena comunidade, com algumas centenas de *host*s. Para mapear os nomes de cada *host* conectado à ARPAnet para seus respectivos endereços IP, era utilizado um único arquivo, HOSTS.TXT, que continha todas as informações necessárias para esses *hosts*.

O arquivo HOSTS.TXT era atualizado pelo *Network Information Center* do SRI (antigo *Stanford Research Institute*, na Califórnia), e distribuído a partir de um único *host*, SRI-NIC. Todas as mudanças feitas pelos administradores da ARPAnet eram enviadas por e-mail para o NIC, que as compilavam em um novo HOSTS.TXT, que por sua vez era enviado para o SRI-NIC para que os administradores capturassem o atual HOSTS.TXT. Este esquema se tornou impraticável à medida que a ARPAnet crescia, pois o arquivo precisava crescer proporcionalmente ao crescimento da rede. Quando a ARPAnet mudou para os protocolos TCP/IP, a população da rede cresceu ainda mais, fazendo surgir vários problemas com o HOSTS.TXT:

- Tráfego e carga: a taxa na SRI-NIC, cobrada pelo tráfego da rede e carga de processadores envolvidos na distribuição do arquivo começava a ficar insustentável.

- Colisões de nomes: não podia haver dois *hosts* no HOSTS.TXT como mesmo nome. Isto poderia trazer várias conseqüências, como a parada de um serviço de e-mail, por exemplo.

- Consistência: a alteração de um *host* na rede ou o surgimento de um novo *host* tornava o arquivo inconsistente, em uma rede em expansão.

Estes problemas tornaram o mecanismo falho, sendo necessário um mecanismo sucessor. "A meta era criar um sistema de tabelas de *host*s unificados. O novo sistema deveria permitir a administração local de dados, e ainda disponibilizar os dados globalmente[3]", eliminando o gargalo de um *host* único e liberando o problema de tráfego. "E o gerenciamento local realizaria a tarefa de manter os dados atualizados mais facilmente. O novo sistema deveria usar um espaço de nomes hierárquico para nomear *hosts*, garantindo assim a singularidade dos nomes [3]."

O responsável pelo projeto de arquitetura do novo sistema foi Paul Mockapetris,do Instituto de Ciência da Informação da Universidade da Califórnia. Ele liberou em 1984 o *Domain Name Server*, descrito pelas RFCs 882 e 883, que foram suplantadas pelas RFCs 1034 e 1035, que são as especificações atuais do sistema.

2.1. Funcionamento e Estrutura do DNS

Basicamente, o DNS serve para mapear nomes em endereços IP e vice-versa. Sintaticamente, cada nome de computador consiste em uma seqüência de segmentos alfanuméricos separados por pontos. Por exemplo, um computador no departamento de informática da empresa FICTICIA tem o nome de domínio: **metron.inf.ficticia.com.br**

Lembrar e trabalhar com nomes é bem mais fácil para as pessoas do que com números, mas para o computador isso é inconveniente. “Por ser mais compacta que um nome simbólico, a forma binária de um endereço IP requer menos processamento para manipulação (por exemplo, para comparar). Além disso, um endereço ocupa menos memória e exige menos tempo para ser transmitido através de uma rede do que um nome [4]”. Assim, mesmo que o software aplicativo permita ao usuário digitar nomes, os protocolos exigem endereços numéricos, que deverá ser traduzido por um aplicativo antes de usá-lo na comunicação.

Este exemplo caracteriza uma interação cliente-servidor, que é o fato de um aplicativo necessitar de outro para a tradução de nomes em endereços IP. O cliente envia uma mensagem de requisição para um servidor de nomes, que pode responder com a mensagem de resposta ou se tornar cliente de um outro servidor de nomes, até que seja encontrada a resposta correta para a requisição e esta ser passada para o aplicativo que a requisitou. Este banco de dados de nomes está distribuído em um conjunto de servidores localizados em diversos lugares no mundo, tornando o DNS um sistema de banco de dados distribuído. “Isso permite um controle local dos segmentos do banco de dados global, embora os dados em cada segmento estejam disponíveis em toda a rede através de um esquema cliente-servidor. Desempenhos resistentes e adequados são adquiridos por meio de reprodução e *cache* [4]”.

Os nomes de domínio obedecem a uma hierarquia, sendo lida da esquerda para direita o nível hierárquico. Quanto mais à direita, maior o nível. Os nomes à esquerda representam os nomes das máquinas e os da direita os subdomínios e domínios a qual a máquina pertence. O DNS especifica valores para os níveis mais altos, chamados TLDs (*Top Level Domains*). Todos os TLD´s são mostrados na Tabela 1 (atualizado de acordo com o ICANN, em 05-Jan-2006):

Tabela 1 – Top Level Domains (TLD)

TLD	Data de introdução	Propósito	Responsável
.aero	2001	Indústria de transportes aéreos	Societe Internationale de Telecommunications Aeronautiques SC, (SITA)
.biz	2001	Negócios	NeuLevel
.cat	2005	Comunidade lingüística e cultural Catalã	Fundació puntCAT - Espanha
.com	1995	Irrestrito, mas com intenções de registros comerciais.	VeriSign, Inc., USA
.coop	2001	Cooperativas	DotCooperation, LLC
.edu	1995	Organizações educacionais	EDUCAUSE, USA
.gov	1995	Organizações governamentais	US General Services Administration
.info	1998	Uso irrestrito	Afilias Limited, Irlanda
.int	1998	Organizações estabelecidas em tratados governamentais	Internet Assigned Numbers Authority, USA
.jobs	2005	Comunidade Internacional de gerência de recursos humanos.	Employ Media LLC, USA
.mil	1995	Organizações militares	US DoD Network Information Center, USA
.museum	2001	Museus	Museum Domain Management Association, (MuseDoma)
.name	2001	Para registros individuais	Global Name Registry, LTD, UK
.net	1995	Irrestrito, mas voltado para organizações que de alguma forma colaboram ou administram a Internet	VeriSign, Inc., USA
.org	1995	Organizações não-governamentais e sem fins lucrativos	Public Interest Registry. Until 31 December 2002, .org was operated by VeriSign Global Registry Services, USA.
.pro	2002	Profissionais liberais	RegistryPro, LTD, USA
.travel	2005	Comunidade de viagens e turismo	Tralliance CorporationTralliance Corporation, USA

Cada nome de domínio é um caminho em uma grande árvore invertida, denominada *espaço de nomes de domínio*. Os nomes de domínio obedecem à hierarquia dessa árvore, que possui uma

única raiz no topo, chamada *raiz*, que pode se ramificar de várias maneiras em cada ponto de interseção ou *nó*.

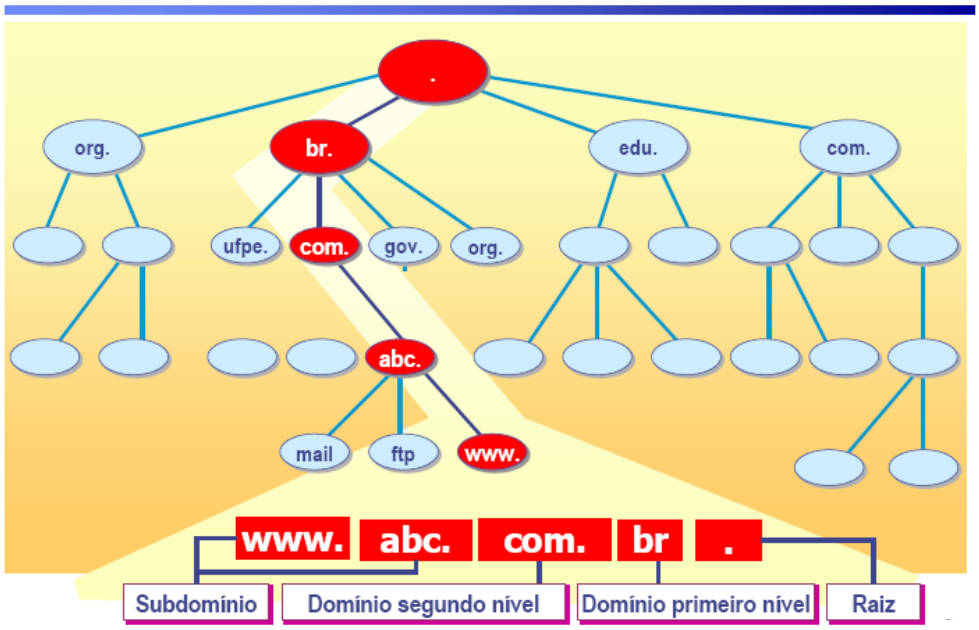

Figura 1: Exemplo da hierarquia de domínios

Cada nó da hierarquia na árvore possui um nome. O caminho completo desses nomes desde um nó até a raiz chama-se *nome de domínio*. Eles são sempre lidos do nó para a raiz, com pontos separando os nomes no caminho. Ao nó raiz é reservado um nome vazio, representado por um ponto (“.”). “Alguns softwares interpretam um ponto subseqüente em um nome de domínio para indicar que o nome do domínio é absoluto. O nome de domínio absoluto é escrito em relação à raiz e especifica sem ambigüidade a localização do nó na hierarquia [3] ”, também conhecido como *nome de domínio totalmente qualificado* (FQDN – *Fully Qualified Domain Name*).

Para garantir que cada nó seja identificado de forma única dentro de um domínio, o DNS requer que os nós filhos do mesmo pai tenham nomes diferentes. Essa restrição é apenas para nós filhos, e não para todos os nós da árvore. “O número de segmentos em um nome de domínio corresponde à hierarquia de nomes. Não existe nenhum padrão universal porque cada organização pode escolher a forma como estruturar nomes em sua hierarquia. Além disso, nomes dentro de uma organização não precisam seguir um padrão uniforme porque grupos individuais dentro da organização podem escolher uma estrutura hierárquica que é apropriada ao grupo [4]”.

Um domínio é uma parte do espaço de nome de DNS que inclui um nome de domínio e todos os nomes abaixo dele. Um domínio pode ter diversas sub-árvores próprias, denominadas sub-

domínios. Um domínio é um sub-domínio dentro de outro domínio se a raiz do sub-domínio estiver dentro do domínio.

2.2. Componentes da resolução DNS

Toda pesquisa DNS é composta por três componentes básicos: os clientes DNS, os registros de recursos e os servidores DNS. A seguir será mostrado como cada um desses componentes participam do processo de uma pesquisa DNS.

2.2.1. Os clientes DNS

Os clientes DNS, também chamados de *resolvers*, ou solucionadores, são programas instalados nas estações de trabalho e são responsáveis por detectar sempre que um programa precisa de resolução de um nome e repassar esta consulta para um servidor DNS. Os solucionadores consultam o servidor de nomes, interpretam as respostas retornadas pelo servidor e retornam as informações aos programas que as solicitaram.

Quando uma aplicação precisa resolver um nome, ela é endereçada pelo ao solucionador, e então é iniciado o processo de resolução de nomes. Primeiro, o endereço é passado ao cliente DNS, que tenta resolver o nome utilizando o *cache* DNS local ou o arquivo *host*.

Sempre que um nome é resolvido com sucesso, o resultado da resolução é mantido na memória, para que o cliente DNS utilize essas informações para resolver as solicitações (*cache*). Quando o cliente DNS não encontra a resposta no *cache*, o cliente consulta as entradas no arquivo *host*, que é um arquivo texto e possui dados relacionando endereços IP a nomes.

Caso a consulta não seja respondida após estes processos, o cliente DNS contacta o servidor de nomes local, enviando na mensagem o nome a ser resolvido, o tipo de pesquisa a ser realizado e a classe associada ao nome DNS. Então, espera que o servidor de nomes envie de volta uma mensagem de resposta DNS que contém a resposta da requisição. O solucionador então interpreta essa resposta, que pode ser registros de recursos ou um erro, e a envia para o programa que originou a requisição.

2.2.2. Registros de Recursos

"Os dados associados com os nomes de domínios estão contidos em **registros de recursos**, ou RRs [3]" **.**Os registros são as entradas do banco de dados do DNS. Em cada entrada existe um mapeamento entre um determinado nome e uma informação associada ao nome. Pode ser desde um simples mapeamento entre um nome e o respectivo endereço IP, até registros mais

sofisticados para a localização de DCs (controladores de domínio do Windows 2000 ou Windows Server 2003) e servidores de *e-mail* do domínio.

Os registros de recursos definem os tipos de dados no DNS, e são armazenados em formato binário, chamado mensagem, mas são enviados pela rede em formato texto, enquanto executam a transferência de zonas. Esta mensagem é dividida em 5 seções, onde cada seção define os nomes do domínio dentro do espaço do de nomes de domínios.

A seção *cabeçalho* está sempre presente, inclui campos que especificam quais das seções restantes estão presentes e se o pacote DNS é uma pesquisa ou uma resposta. "Os nomes das seções depois do cabeçalho são derivados do seu uso em uma pesquisa básica. A seção *question* contém campos que descrevem a pesquisa para o nome do servidor. Estes campos são do tipo pesquisa (QTYPE), tipo classe (QCLASS) e tipo nome de domínio (QNAME). A seção *answer* contém RRs com a resposta da pesquisa; a seção *authority* contém RRs que apontam para o servidor de nomes autoritativo e a seção *additional records* contém RRs que são relacionados com a pesquisa, mas não são estritamente respostas para a pesquisa [4]". Abaixo segue o esquema das seções da mensagem DNS:

Tabela 2 - Seções de uma mensagem DNS

Cabeçalho	Descreve as outras seções e possui os 16 bits identificadores da consulta, o *query ID.*
Pergunta	Pesquisa para o servidor de nomes, contendo o nome, a classe e o tipo do conjunto de registros de recursos buscados, formando a resposta da consulta (RRset).
Resposta	RRs com a resposta da pesquisa. Sempre vazia ns consultas, contém o RRset que casa com a consulta, ou está vazia se o nome não existe, se foi feita uma consulta não recursiva ou se a consulta resulta numa referência (*referral*).
Autoridade	RRs apontador para servidor de nomes autoritativo. É sempre vazia nas consultas. Pode estar vazia em respostas. Se não estiver vazia, contém os RRs, NS e SOA da zona consultada. Isto é chamado, geralmente, de *referral data.*
Adicional	RRs informações adicionais. Também é sempre vazia em consultas e pode estar vazia em respostas.

De acordo com a RFC 1035, registro de recursos do DNS tem 6 (seis) campos, a seguir:

- NAME: contém o nome DNS, e também uma referência ao seu próprio nome, para o qual o RR pertence.

- TYPE: é o tipo do RR. Este campo é necessário, pois não é comum para um nome DNS ter mais de um tipo de RR. Os tipos mais comuns de RRs são os seguintes:

Tabela 3 - Principais registros de recursos

Tipo de Registro	Descrição	Uso
A	Registro tipo endereço	Mapeamento de FQDN em endereço IP.
PTR	Registro tipo ponteiro	Mapeamento de endereço IP em FQDN.
SOA	Registro tipo *Start of Authority*	Especifica atributos referente à zona, como nome do domínio, contato administrativo, número de série da zona, TTL, etc.
CNAME	Registro tipo nome canônico	Nome de domínio canônico para um *alias* (nome alternativo).
MX	Registro tipo *Mail Exchange*	Nome do servidor de correio eletrônico para o domínio.
NS	Registro tipo *name server*	Define o nome do servidor da zona

- CLASS: cada classe pertence a um tipo de rede ou software. Atualmente existem classes para internets, redes baseadas nos protocolos Chaosnet e redes que utilizam o software Hesiod. Por exemplo, utiliza-se IN para internets e CH para redes Chaosnet[3].
- TTL: especifica o intervalo do tempo que o registro do recurso pode armazenar informações antes que a fonte da informação deva outra vez ser consultada.
- RDATA: o formato desse campo varia de acordo com o tipo e a classe do RR.
- RDATA Length: tamanho em octetos do campo RDATA.

"Os registros de recursos são agrupados em conjuntos chamados RRSets, que contém 0 ou mais RRs. Estes conjuntos contém o mesmo nome DNS, classe e tipo, mas o campo RDATA é diferente. Se o nome, classe, tipo e informação (RDATA) são os mesmos para dois ou mais registros, então o registro duplicado existe para o mesmo nome DNS [1]".

2.2.3. Servidores DNS

"Os programas que armazenam informações sobre o espaço de nomes de domínio são chamados de servidores de nomes (*name server*) [3]". Os servidores de nomes são responsáveis por receber a solicitação enviada pelos clientes DNS (*resolver*) e responder a esta solicitação.

Um servidor de nomes tem autoridade sobre uma zona quando este servidor possui informações completas sobre uma parte do espaço de nomes de domínio (zona), que carregam a partir de um arquivo no próprio servidor ou de outro servidor de nomes.

Uma organização que administra um domínio pode subdividir este domínio em subdomínios e atribuir a uma outra organização a responsabilidade pela administração deste subdomínio. Por exemplo, a organização administradora do domínio ***.br*** atribui a responsabilidade de administrar o subdomínio "sitecnet" à empresa SitecNet Informática[1], que solicitou à organização a criação deste subdomínio. Este processo chama-se **delegação de zonas**. Então, um servidor de nomes que contém informações completas sobre esta zona tem autoridade sobre esta zona. Um servidor de nomes pode ter autoridade sobre uma ou mais zonas.

Atualmente existem 13 servidores de nomes raiz espalhados pelo mundo, que contém as informações dos servidores de nomes autorizados para cada uma das zonas de nível mais alto, e gerenciam mais de 9 bilhões de consultas todos os dias [2]. Estes servidores são responsáveis pela zona raiz, que está hierarquicamente acima do espaço de nomes DNS. Os servidores raiz estão descritos na Tabela 4.

Tabela 4 – Localização dos servidores raiz do DNS

Letra	Operador	Localização
A	VeriSign	Dulles, Virginia, USA
B	ISI	Marina Del Rey, California, USA
C	Cogent	(Distribuído, usando anycast)
D	University of Maryland	College Park, Maryland, USA
E	NASA	Mountain View, California, USA
F	ISC	(Distribuído, usando anycast)
G	U.S. DoD NIC	Columbus, Ohio, USA
H	U.S. Army Research Lab	Aberdeen Proving Ground, Maryland, USA
I	Autonomica	(Distribuído, usando anycast)
J	VeriSign	(Distribuído, usando anycast)
K	RIPE NCC	(Distribuído, usando anycast)
L	ICANN	Los Angeles, California, USA
M	WIDE Project	Tokyo, Japan

[1] na verdade, o domínio br atribui a responsabilidade ao domínio "com", "org", "edu", etc. Só que existe a coincidência de que quem gerencia "br" também gerencia os demais (com, org...) que é a Fapesp-SP.

Existem dois tipos de servidores de nomes autoritativos (que tem autoridade sobre a zona): o servidor de nomes mestre primário, que armazena dados de uma zona em um arquivo de zona, que define os *host*s/servidores existentes na zona, e o servidor de nomes mestre secundário, que também armazena informações sobre a zona, mas recupera estas informações a partir do servidor mestre primário ou de outro secundário. Quando um servidor mestre secundário carrega informações de zona de um servidor primário ou de outro secundário, acontece a *transferência de zona*.

Os servidores de nomes buscam informações em suas zonas e respondem a consultas de outros servidores de nomes. Uma consulta enviada ao servidor de nomes pode ser resolvida de duas maneiras: de modo recursivo ou de moto iterativo.

Quando o servidor de nomes tem autoridade sobre a zona pesquisada (o servidor é primário para a zona), diz-se que o servidor respondeu com autoridade (resposta do tipo *authoritative*) à requisição.

Quando o servidor de nomes não possui autoridade sobre a zona pesquisada, então ele vai pesquisar a resposta em seu *cache*, pois à medida que as requisições são resolvidas, o servidor DNS armazena o nome do endereço IP mapeado, para poder resolver o nome de forma mais rápida da próxima vez que houver uma requisição para este mesmo nome. Estas informações são mantidas no servidor por um tempo, chamado TTL (*Time to Live*), configurado no arquivo de zona pelo administrador do DNS.

Caso o servidor de nomes não resolva a requisição com autoridade, o servidor fará consultas a outros servidores de nomes para resolver a consulta enviada pelo cliente. Este processo é chamado *recursão*.

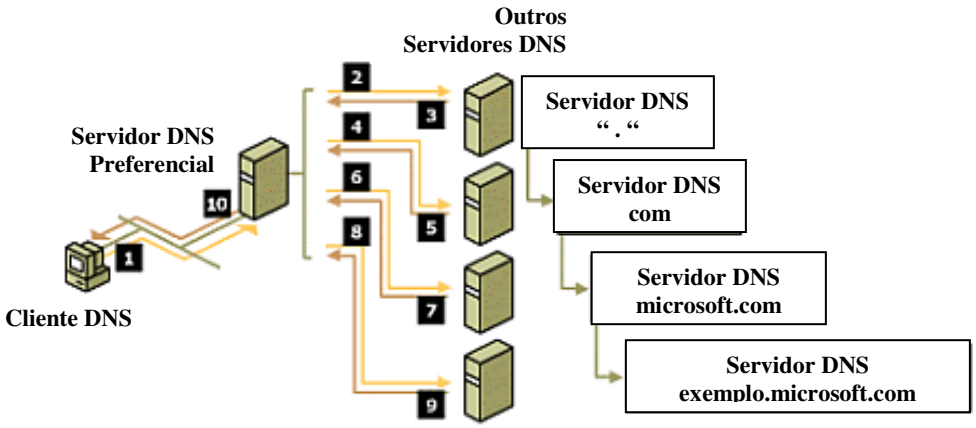

Figura 2: Resolução Recursiva

Por exemplo, uma aplicação deseja acessar o site **exemplo.microsoft.com**, e está utilizando um computador da rede **sitecnet.com.br**, utilizando o servidor DNS com autoridade apenas para a zona **sitecnet.com.br**, e não para **exemplo.microsoft.com**. Assim, o servidor **sitecnet.com.br** irá consultar outros servidores de nomes em busca da resolução da solicitação a ele enviada, da seguinte maneira:

1. O servidor de nomes retira apenas a parte do nome do domínio (**microsoft.com**), e utilizando a lista de servidores raiz, localiza um servidor de nomes com autoridade para o domínio root da Internet, representado pelo ponto (.);
2. Localizado o servidor responsável pelo domínio root, o servidor DNS da empresa Sitecnet envia uma consulta iterativa para o servidor DNS do domínio root solicitando informações sobre o servidor responsável pelo domínio .com;
3. O servidor DNS do domínio Sitecnet recebe a resposta informando qual o servidor DNS responsável pelo domínio .com.
4. O servidor DNS do domínio Sitecnet envia uma consulta para o servidor DNS responsável pelo .com (informado no passo 3), solicitando informações sobre o servidor responsável pelo domínio **microsoft.com**;
5. O servidor DNS responsável pelo .com não é a autoridade por **microsoft.com**, mas sabe informar quem é a autoridade deste domínio. O servidor DNS responsável pelo .com retorna para o servidor DNS do domínio Sitecnet, o número IP do servidor DNS responsável pelo domínio **microsoft.com**;
6. O servidor DNS do domínio Sitecnet recebe a resposta informando o número IP do servidor responsável pelo domínio **microsoft.com**;
7. O servidor DNS do domínio Sitecnet envia uma consulta para o servidor DNS responsável pelo **microsoft.com** (informado no passo 6) solicitando informações sobre o servidor responsável pelo domínio **exemplo.microsoft.com**;
8. O servidor DNS responsável pelo Sitecnet não é a autoridade para **exemplo.microsoft.com**, mas sabe informar quem é a autoridade deste domínio. O servidor DNS responsável pelo **microsoft.com** retorna para o servidor DNS do domínio Sitecnet, o número IP do servidor DNS responsável pelo domínio **exemplo.microsoft.com**;

9. O servidor DNS do domínio Sitecnet recebe a resposta informando o número IP do servidor responsável pelo domínio **exemplo.microsoft.com**.
10. O servidor DNS do domínio Sitecnet recebe a resposta da consulta, faz uma cópia desta resposta no *cache* do servidor DNS e retorna o resultado para o cliente que originou a consulta.
11. . No cliente, o *resolver* recebe o resultado da consulta, repassa este resultado para o programa que gerou a consulta e grava uma cópia dos dados no *cache* local do DNS.

O processo descrito anteriormente termina com o servidor DNS (após ter consultado vários outros servidores) retornando uma resposta positiva para o cliente, isto é, conseguindo resolver o nome e retornando a informação associada (normalmente o número IP associado ao nome) para o cliente. Mas nem sempre a resposta é positiva; muitos outros tipos de resultados podem ocorrer em resposta a uma consulta, tais como:

***An authoritative answer* (resposta com autoridade):** Este tipo de resposta é obtido quando o nome é resolvido diretamente pelo servidor DNS que é a autoridade para o domínio pesquisado. Por exemplo, um usuário da intranet da empresa (**sitecnet.com.br**), tenta acessar uma página da intranet da empresa, por exemplo: **exemplo.microsoft.com**. Neste caso a consulta será enviada para o servidor DNS da empresa, o qual é a autoridade para a zona **sitecnet.com.br** e responde diretamente à consulta, informando o número IP do servidor **exemplo.microsoft.com**. É também uma resposta positiva só que com autoridade, ou seja, respondida diretamente pelo servidor DNS que é a autoridade para o domínio pesquisado.

A *positive answer* (resposta positiva): É uma resposta com o resultado para o nome pesquisado, isto é, o nome pode ser resolvido e uma ou mais informações associadas ao nome são retornadas para o cliente.

A *referral answer* (uma referência): Este tipo de resposta não contém a resolução do nome pesquisado, mas sim informações e referência a recursos ou outros servidores DNS que podem ser utilizados para a resolução do nome. Este tipo de resposta será retornado para o cliente, se o servidor DNS não suportar o método de recursão, descrito anteriormente. As informações retornadas por uma resposta deste tipo são utilizadas pelo cliente para continuar a pesquisa, usando um processo conhecido como iteração (o qual será descrito mais adiante). O cliente faz a pesquisa em um servidor DNS e recebe, como resposta, uma referência a outro recurso ou

servidor DNS. Agora o cliente irá interagir com o novo recurso ou servidor, tentando resolver o nome. Este processo pode continuar até que o nome seja resolvido ou até que uma resposta negativa seja retornada, indicando que o nome não pode ser resolvido.

A *negative answer* (uma resposta negativa): Esta resposta pode indicar que um dos seguintes resultados foi obtido em resposta à consulta: um servidor DNS que é autoridade para o domínio pesquisado informou que o nome pesquisado não existe neste domínio, ou um servidor DNS que é autoridade para o domínio pesquisado informou que o nome pesquisado existe, mas o tipo de registro não confere.

Em uma pesquisa no modo *iterativo*, o servidor de nomes retorna a melhor resposta que já conhece, podendo ser uma resposta à solicitação ou uma indicação de um outro servidor de nomes. O servidor de nomes consultado pesquisa a resposta em seus dados locais (incluindo o *cache*), e caso não encontre a resposta, retorna como referência os nomes e endereços dos servidores de nomes mais próximos do nome de domínio consultado.

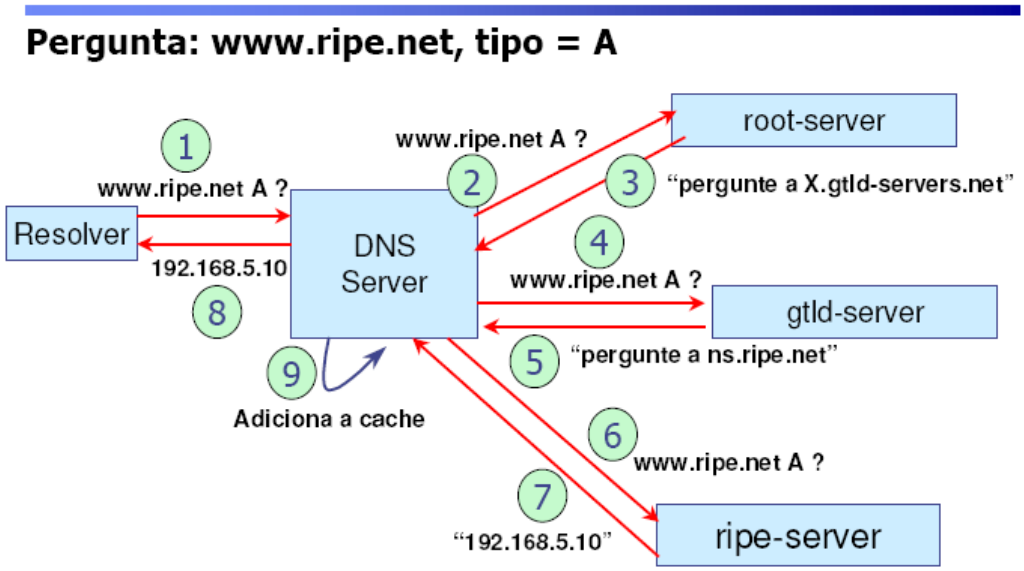

Figura 3: resolução interativa

2.2.4. Pesquisa reversa

Considerando-se a facilidade de encontrar dados tendo-se o nome servindo de índice, criou-se uma parte do espaço de nomes de domínios que utiliza os endereços como nomes. Esta porção do espaço de nomes é o domínio IN-ADDR.ARPA. Por causa da ordem de significância do sistema de nomes ser mais elevado na direita, a notação para endereços é reversa. Por

exemplo, se o endereço IP de **sitecnet.com.br** é 201.30.147.2, a entrada correspondente no domínio IN-ADDR.ARPA é 147.30.201.IN-ADDR.ARPA, o qual mapeia de volta para o nome de domínio **sitecnet.com.br**.

"A consulta inversa é uma pesquisa de nome de domínio que indexa um determinado dado. É processada somente pelo servidor de nomes que recebeu a consulta. Esse servidor de nomes pesquisa todos os dados locais em busca do item solicitado e, se possível, retorna o nome do domínio que o indexa. Como qualquer servidor de nomes conhece apenas parte do espaço geral de nomes de domínio, nunca é garantido que a consulta inversa devolva uma resposta. [3]".

3. Os problemas do DNS

O sistema de nomes de domínio é responsável por traduzir diversos nomes de domínios existentes para seus respectivos endereços IP, atendendo a requisições de clientes DNS (*resolvers*) na porta 53, utilizando, para estas comunicações, tanto o protocolo UDP (*User Datagram Protocol*) quanto o TCP (*Transmission Control Protocol*). As pesquisas DNS são manipuladas sobre o protocolo UDP, enquanto as transferências de zonas utilizam o protocolo TCP, ambos na porta 53[6].

Por seu uso ser amplamente utilizado e necessário na Internet, os administradores de rede não podem bloquear, em seus mecanismos de defesa, como *Firewalls*, a passagem de pacotes com destino a porta 53.

A classificação dos problemas do DNS não é uniforme, pois é divergente entre vários autores quanto à classificação e terminologia. A seguir será feita uma compilação da literatura, abordando alguns dos principais problemas do DNS.

3.1. O problema do protocolo UDP

Uma das principais razões que causam a insegurança do DNS é o uso do protocolo UDP como protocolo de transporte[38]. Ao realizar uma consulta, utilizando o protocolo UDP, o cliente escolhe um número de porta aleatório e envia um pacote destinado à porta 53 do servidor DNS. O servidor, ao receber a requisição, responde com um ou mais pacotes para a porta de origem da máquina do cliente. São características do protocolo UDP:

- não orientado à conexão, ou seja, não garante a entrega do pacote ao destino e não necessita estabelecer, gerenciar e fechar conexões, tendo apenas que transmitir os dados;
- não utiliza mecanismos de reconhecimento para assegurar que as mensagens transmitidas cheguem ao seu destino;
- não ordena as mensagens que chegam;
- não provê meios para controlar a taxa com que as informações fluem entre as máquinas.

Todas estas características do protocolo UDP na pesquisa DNS significam que o cliente não pode garantir que a resposta enviada pelo servidor à sua consulta seja necessariamente respondida pelo servidor ao qual foi enviada a consulta.

3.2. Interceptação de pacotes

Por todos os métodos existentes de captura de pacotes, este tipo de ataque é um dos mais comuns contra o DNS. Capturando um pacote, o atacante pode simplesmente modificar os dados da resposta de uma consulta a um servidor DNS e redirecionar o tráfego para um local à sua escolha.

Este tipo de ataque torna-se relativamente simples pelo próprio comportamento do DNS e seu protocolo UDP: o comportamento do DNS ao ser consultado é enviar respostas às consultas ou encaminhá-las, utilizando um pacote UDP não assinado e não criptografado. Se um atacante conseguir capturar uma consulta a um servidor DNS e injetar uma resposta do tipo *referral*[2], pode fazer com que o cliente que originou a consulta acredite que o atacante possui o servidor que tem autoridade para responder à sua consulta, já que o protocolo UDP não realiza nenhum tipo de verificação na resposta enviada pelo servidor.

Um exemplo deste tipo de ataque é o chamado ***Man-in-the-Middle***, onde um atacante tenta interceptar uma comunicação entre duas partes. Um usuário tenta acessar um *site* de *e-commerce* para realizar uma compra. Ao clicar em um *link* onde seja necessário inserir dados financeiros, o usuário é redirecionado para um outro *site* onde o atacante captura as informações fornecidas pelo usuário. Após isso, o usuário é redirecionado novamente para o *site* original e o atacante consegue os dados desejados.[10]

Outro tipo de ataque que pode ser aplicado utilizando a interceptação de pacotes é quando o atacante utiliza um servidor DNS e faz com que ele se passe por um servidor confiável (*Trusted Server*), no qual os clientes enviarão consultas e receberão respostas falsas.

Este tipo de vulnerabilidade aplica-se à **comunicação entre cliente-servidor**, levando a ataques do tipo *Cache Poisoning, DoS* e *DNS Spoofing*.

3.3. Transferência de zonas e atualizações dinâmicas

O DNS armazena suas informações em uma tabela, contendo dados que relacionam nomes de máquinas e endereços (registros do tipo A), apelidos para essas máquinas (registros do

[2] Ver seção 2.2.3.

tipo CNAME), máquinas responsáveis por receber e-mails do domínio (registros do tipo MX), entre outros dados que podem revelar informações sobre a rede ou domínio para qual o servidor é autoridade. Esta tabela é chamada de **arquivo de zona**.

A transferência de zonas entre servidores ocorre quando um servidor consultado não tem autoridade para responder a esta consulta e a encaminha para outro servidor DNS, e tem por objetivo a replicação das informações de um servidor DNS primário para um ou mais servidores DNS secundários.

Em uma transferência de zonas, um invasor pode ter acesso não autorizado às informações contidas na troca dos dados entre os servidores, e até seqüestrar uma zona inteira, redirecionando todo o tráfego enviado ao servidor do domínio seqüestrado e respondendo ao cliente com respostas falsas. Este ataque é conhecido como *DNS Hijacking*, e a vulnerabilidade acontece na **atualização entre servidores**.

Uma atualização dinâmica permite que clientes DNS registrem e atualizem dinamicamente seus registros de recursos em um servidor DNS sempre que ocorrerem alterações. Isso reduz a necessidade de administração manual de registros de zona, especialmente para clientes que se movem ou alteram suas localizações com freqüência e usam DHCP para obter um endereço IP.[13]

O protocolo de atualização dinâmica do DNS provê mecanismos para controlar quais sistemas são permitidos para atualizar dinamicamente o servidor primário. Segundo a RFC 2136, um ataque do tipo DoS pode ser enviado por um remetente de uma atualização com o envio de seções TCP simultâneas contendo atualizações que o servidor master primário irá recusar devido a problemas de permissões. Os mecanismos de gerenciamento de conexões podem prevenir danos maiores devido ao ataque, mas não previnem que algumas consultas sejam enviadas sem resposta durante o ataque.

3.4. Cache Poisoning

Sempre que o servidor DNS não tem a resposta para a pesquisa em cache, passa a pesquisa para outro servidor DNS em nome do cliente. Se o servidor passa a pesquisa para outro servidor DNS que contém uma informação errada, então o envenenamento[3] do *cache* (*Cache Poisoning*) pode acontecer.

[3] Significa fazer com que um servidor armazene em *cache* uma informação falsa sobre a resolução de um determinado nome de domínio.

O envenenamento do cache envolve o envio de uma informação de mapeamento incorreto por um servidor DNS com um TTL alto. Quando o servidor DNS for solicitado novamente, ele irá responder com a informação errada. A informação errada é armazenada em *cache* por outro servidor DNS, que torna a informação errada disponível para programas que requisitam a informação do servidor DNS através de uma interface de um cliente.

O servidor DNS com o *cache* poluído responde à pesquisa, mas não necessariamente com a resposta correta; ele preenche a seção "registros adicionais" da mensagem de resposta do DNS com uma informação que não necessariamente se relaciona com a resposta. O servidor que realizou a requisição aceita a resposta e não realiza nenhum tipo de checagem para assegurar-se de que a resposta está correta ou se relaciona com a pesquisa. O servidor aceita a informação e a adiciona em seu *cache*.

Com as informações armazenadas em *cache* alteradas de acordo com sua vontade, um atacante pode redirecionar o tráfego intencionalmente de um *site* verdadeiro para um que esteja sob seu controle. Dois métodos são mais conhecidos para este tipo de ataque:

O objetivo do **redirecionamento do tráfego** é encaminhar a vítima para um local falso sem que a mesma desconfie da falsidade do destino para onde foi indicado. Este ataque é geralmente utilizado para redirecionar *sites* de bancos para ganhar acesso às contas dos usuários ou redirecionar *sites* de notícias para manipular as informações e publicar falsas histórias.

O envenenamento do cache também pode ter como objetivo lançar um ataque do tipo *DoS – **Denial of Service***. Um dos mais perigosos ataques consiste em negar a disponibilidade do serviço de DNS congestionando um ou mais servidores de DNS com consultas recursivas. À medida que o servidor é sobrecarregado com consultas, a CPU atingirá o nível máximo de processamento e o servidor DNS ficará indisponível. Assim, quando um usuário solicitar uma pesquisa verdadeira ao servidor, o serviço de resolução de nomes estará indisponível.

Uma maneira de realizar o DoS em um servidor é inserir um registro do tipo CNAME no cache do servidor referindo-se a ele mesmo no nome canônico.

```
exemplo.sitecnet.com.br     IN     CNAME     exemplo.sitecnet.com.br
```

Após inserir este registro de recurso no *cache* do servidor, o atacante pode causar uma falha no DNS por uma simples transferência de zona.

3.5. Comprometimento do Servidor

Um atacante que conseguir acesso não autorizado ao servidor DNS com privilégios administrativos pode comprometer totalmente o serviço DNS, modificando informações que são servidas aos clientes. Uma vez com o controle do servidor, o invasor tem o controle de todas as informações referentes à pesquisa de nomes da rede Este tipo de ataque é utilizado principalmente para realizar outros tipos de ataques, como DoS e *cache poisoning*.

Outro objetivo desse tipo de ataque é a obtenção de informações referente ao domínio à qual o servidor é autoridade. De posse dos dados da zona para o qual o servidor é autoridade, o atacante tem acesso aos nomes dos *hosts*, domínios e endereços IP de recursos da rede. Muitas vezes, nomes de máquinas e de domínios representam serviços vitais da rede, como servidores de *e-mail*, *firewall*, arquivos e FTP, por exemplo, revelando informações que o atacante pode tirar proveito até para realizar outros tipos de ataque.

3.6. DNS Spoofing

Basicamente, a técnica do DNS *spoofing* (falsificação de respostas DNS) consiste em responder com informações falsas consultas recursivas feitas a servidores DNS, resultando no envenenamento do cache do servidor que originou a requisição, entre outros tipos de conseqüências citadas mais adiante.

Quando um servidor DNS envia uma consulta em busca de o endereço IP de um determinado nome de domínio, envia na consulta informações como o IP da estação de origem, porta UDP de origem e o *transaction ID*, que é um número de identificação utilizado em todas as requisições de resolução de nomes. Se o servidor consultado não tiver autoridade sobre o domínio enviado na consulta, o mesmo fará consultas recursivas até chegar ao servidor do domínio consultado. Este servidor irá responder à consulta em uma das informações verificadas pelo servidor que originou a consulta o campo ID do cabeçalho do pacote DNS, além das seguintes verificações:

- Se o campo ID e o campo “nome a ser resolvido” do pacote DNS são iguais aos enviados na consulta;
- Se a porta de origem é igual à porta de destino do pacote UDP enviado a consulta;
- Se o endereço de IP de origem é igual ao endereço de IP de destino enviado na consulta.

Se um atacante conseguir interceptar o pacote UDP enviado na resposta e injetar informações falsas nele, o servidor que gerou a consulta irá armazenar esta resposta e assim terá seu *cache*

poluído. Quando o cliente consultar este servidor, terá como resposta uma informação falsa e poderá ser direcionado para outro local que não o verdadeiro destino desejado pelo cliente em sua consulta original.

Este ataque visa enviar uma resposta falsa ao servidor DNS vítima antes que a resposta verdadeira enviada pelo servidor DNS questionado chegue à vítima. Esta resposta verdadeira irá chegar até o servidor vítima, mas será descartada, pois este já recebeu uma resposta, mesmo que falsa, com as informações desejadas.

O *dnsspoofing* pode ser realizado através de interceptação de pacotes e predição de *transactions ID*, por exemplo. Quando bem sucedido, o *cache* do servidor vítima permanece poluído por um determinado tempo, armazenando informações falsas sobre um determinado nome de domínio, podendo levar a sérias conseqüências, tais como:

- Redirecionar de tráfego;
- Driblar mecanismos de defesa, baseados em softwares de filtragem de pacotes que utilizam nomes para verificar a autenticidade entre eles;
- Ataques a todas as aplicações que necessitam do serviço de DNS para resolução de nomes;
- Ataques *DoS*;
- Ataques de *buffer overflow* em resoluções reversas.

A figura abaixo ilustra um ataque do tipo DNS *Spoofing*:

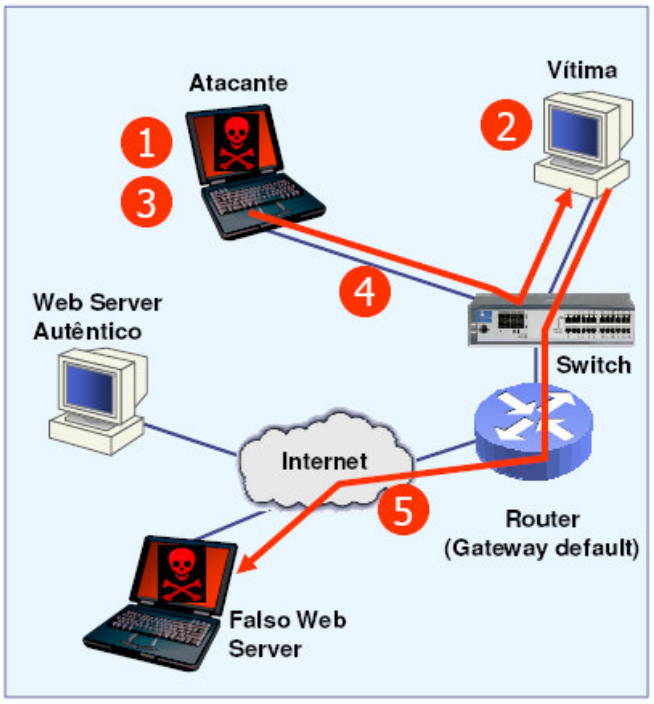

Figura 4: Ataque do tipo DNS *Spoofing*.

O ataque *dnsspoof* funciona da seguinte maneira[14]:

1. o atacante ativa o *dnsspoof*, uma ferramenta que forja consultas DNS;
2. a vítima inicia o acesso a um *web Server* e tenta resolver um nome usando o DNS;
3. o atacante captura a consulta DNS;
4. o atacante responde à consulta DNS com o número IP falso do *web Server*.
5. a vítima acessa o falso *web Server* ao invés do *web Server* verdadeiro.

4. DNSSec: Extensões de segurança para o DNS

Pelo seu funcionamento, explicado anteriormente, o DNS oferece pontos de vulnerabilidade que, se atacadas, podem levar a serias conseqüências, que afetam a integridade dos servidores, da informação oferecida pelo DNS, das comunicações envolvidas no protocolo e a disponibilidade do serviço de DNS.

Por estes motivos, surgiu a necessidade de transformar o DNS em um protocolo seguro e confiável, que oferecesse uma maneira de mitigar ataques e ao mesmo tempo disponibilizasse mecanismos de verificação da autenticidade dos dados, já que a maior parte dos ataques ao DNS visam oferecer dados falsos a clientes e servidores.

Assim, alterações às RFC´s 1034 e 1035, que definem o protocolo DNS, foram propostas, adicionando ao DNS extensões que suprem as necessidades de segurança ausentes no protocolo. A RFC 2065 detalha estas alterações, onde a característica principal das extensões de segurança do DNS, ou DNSSec, é prover a autenticação da origem dos dados das zonas e verificar a integridade destes dados utilizando criptografia de chave pública e assinatura digital.

Resumidamente, o DNSSec oferece os seguintes serviços:

- Distribuição de chaves, por meio da adição de um registro de recurso chamado KEY, armazenado no arquivo de zonas;
- Autenticação da origem dos dados, através das assinaturas criptográficas associadas aos registros de recursos;
- Certificação da transação e requisição, resolvendo o problema das respostas negativas, ou seja, a resposta dada por um servidor quando um nome não existe.

Com estes serviços, será possível para o protocolo checar se os dados publicados por uma entidade são realmente de sua autoridade (autenticidade), se os dados recebidos são da mesma forma como foram publicados (integridade) e a garantia de que a resposta esperada de uma consulta está na resposta da consulta original e que esta resposta vem do servidor para o qual a consulta foi enviada.

4.1. Criptografia de Chave Pública e Assinaturas Digitais

O conceito de criptografia de chaves públicas baseia-se na utilização de dois parâmetros de controles, por meio do qual a mensagem pode ser cifrada ou decifrada, chamado de **chave criptográfica**. Este par de chaves é gerado por meio de uma fórmula matemática, sendo assimétricas entre si, ou seja, uma decodifica o que a outra codificou.

Uma das chaves é de domínio público, sendo disponibilizada em um "canal público" (por exemplo, um repositório de acesso público). Esta chave é chamada de chave pública (*public key*). O outro par da chave é mantido em segredo, e é representada como sendo a identidade do seu proprietário. Esta chave é chamada de chave privada (*private key*). As duas chaves pertencem ao mesmo proprietário. Este processo é chamado de criptografia assimétrica.

Qualquer uma das chaves é utilizada para cifrar uma mensagem e a outra para decifrá-la. As mensagens cifradas com uma das chaves do par só podem ser decifradas com a outra chave correspondente. Por exemplo, se alguém que enviar uma mensagem criptografada a outra pessoa, deve utilizar a chave pública do destinatário para codificar a mensagem, e quando o destinatário receber a mensagem, utilizará sua chave privada para decodificar a mensagem. De forma contrária, um remetente também pode utilizar sua chave privada para codificar uma mensagem e enviar para o destinatário, junto com a mensagem, sua chave pública, que será usada para descriptografar a mensagem.

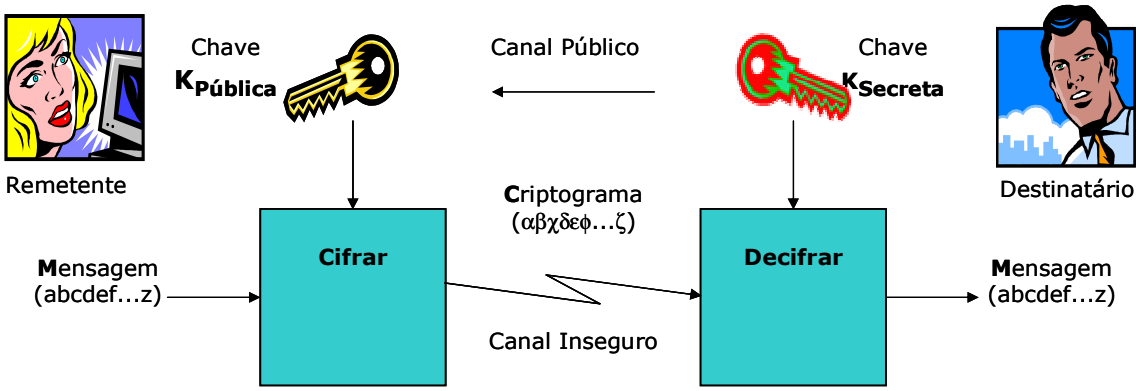

Figura 5: Algoritmo de criptografia assimétrica (chave pública)

Os algoritmos mais utilizados na criptografia assimétrica são o RSA, ElGamal, Diffie-Hellman e algoritmo de Curvas Elípticas.

Além de criptografar as mensagens, o DNSSec também precisa certificar-se de que quem responde às consultas é realmente quem foi consultado. Esta verificação é realizada utilizando o conceito de Assinaturas Digitais.

A assinatura digital é um item que acompanha uma informação, confirmando a origem desta informação, certificando que esta informação não foi modificada e impedindo a negação da origem desta informação. Com isto, o receptor pode verificar a identidade alegada pelo transmissor da informação, não pode repudiar o conteúdo da mensagem nem tentar ele mesmo forjar a mensagem. A assinatura digital apenas não garante o sigilo da mensagem, pois o remetente da mensagem a codifica com sua chave privada a envia junto com a mensagem para o destinatário. Este, por sua vez, decodifica a mensagem com a chave pública do remetente, garantindo assim a autenticidade do envio da mensagem pelo remetente. Estas assinaturas digitais são geradas a partir de algoritmos criptográficos de chave pública.

Porém, a assinatura digital, como dito anteriormente, não garante a confidencialidade da mensagem. Para isto, deve-se combinar os dois métodos de criptografia, ou seja, criptografar a mensagem duas vezes, uma vez com a chave privada e outra com a chave pública do destinatário.

Ao enviar uma mensagem, por exemplo, o remetente a codifica com sua chave privada, e depois a codifica novamente, desta vez incluindo sua assinatura, com a chave pública do destinatário. Este, por sua vez, ao receber a mensagem, deve decodificá-la com sua chave privada, e depois decodificá-la novamente, desta vez com a chave pública do remetente, verificando sua assinatura e garantindo sua autenticidade.

Este método torna-se inviável, quando se trata de grandes mensagens, pois podem levar horas para que estas sejam totalmente cifradas. Um mecanismo foi adicionado a este método para oferecer agilidade nas assinaturas digitais, além de integridade confiável.

O uso de uma função *hashing* unidirecional, que extrai um trecho qualquer do texto simples, e a partir deste, gera um valor pequeno, de tamanho fixo, chamado *hash* (geralmente de 128 a 256 *bits*), independentemente do tamanho da entrada, para autenticar apenas uma parte da mensagem, em vez da mensagem inteira. Assim, torna-se difícil encontrar duas mensagens produzindo o mesmo resultado *hash*.

Seguem abaixo duas figuras ilustrando o mecanismo de geração e verificação do *hash*.

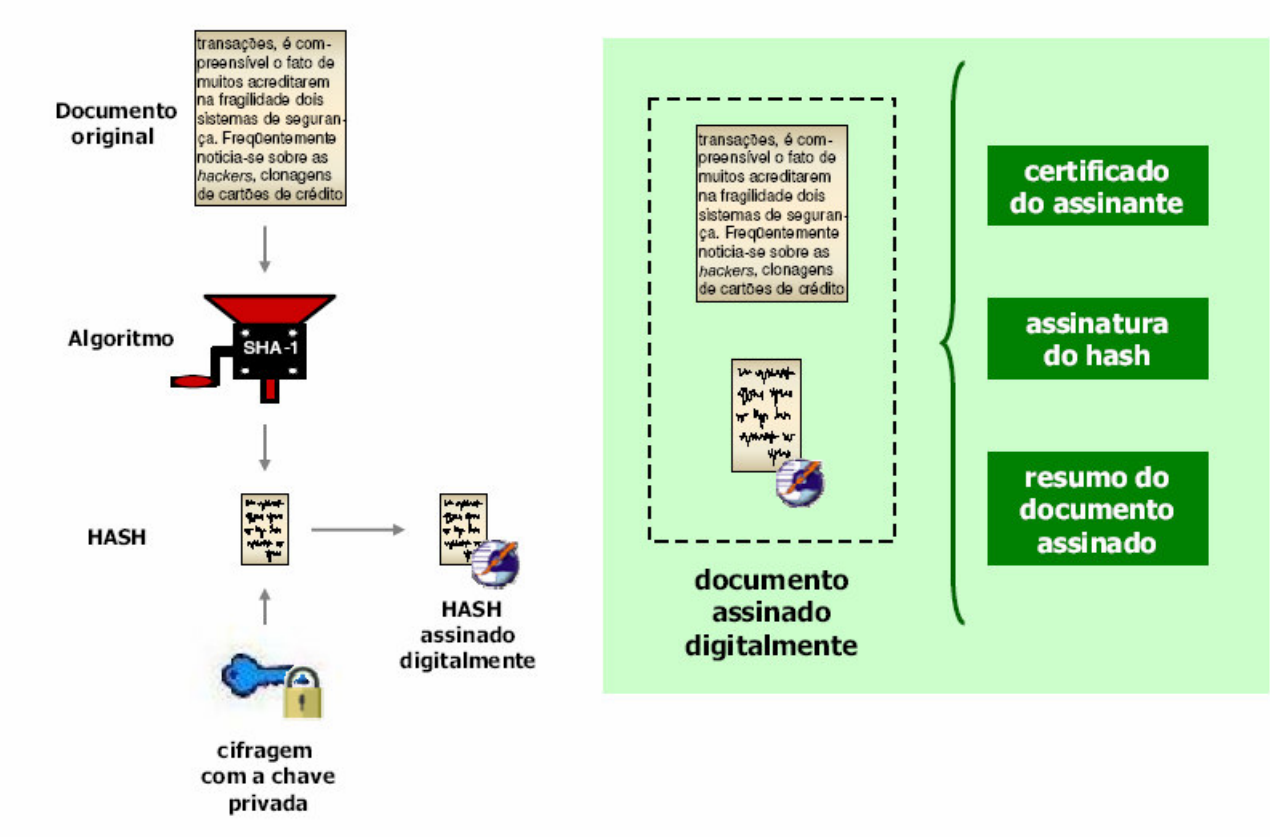

Figura 6: Assinatura Digital – Geração do hash.

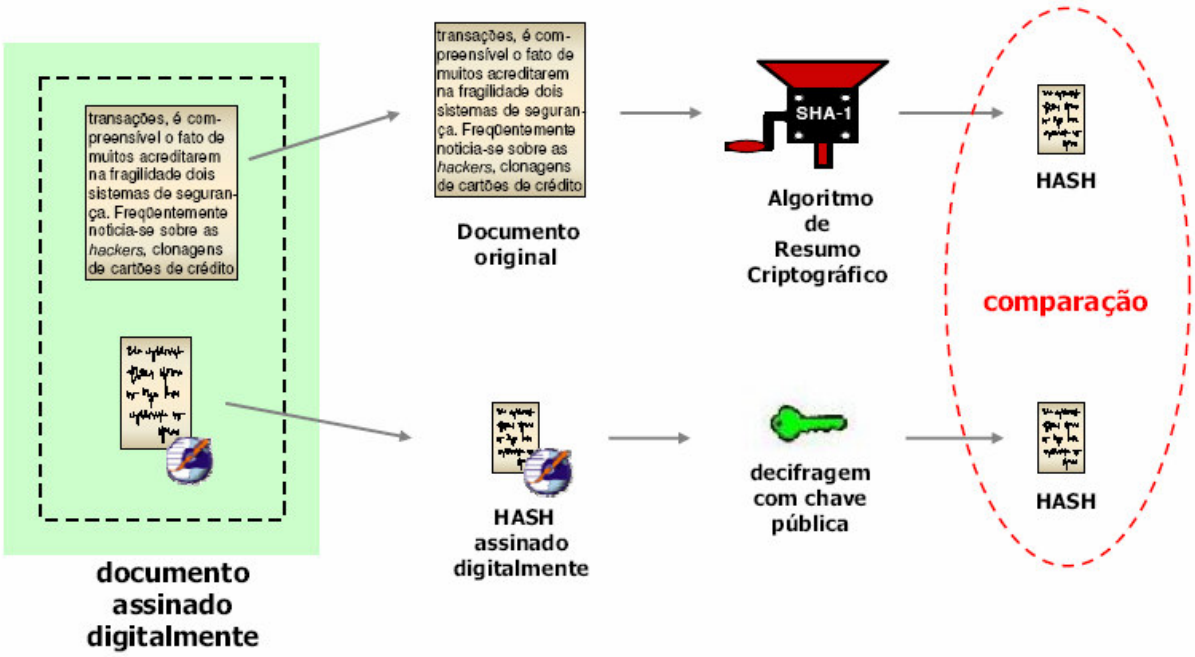

Figura 7: Assinatura Digital – verificação do *hash*.

4.2. Registro DNSKEY

Cada zona segura tem um par de chaves associadas a ela. Os nomes a serem associados podem representar uma zona, uma máquina, um usuário ou uma outra entidade final. A chave pública do par de chaves da zona é publicada como um novo tipo de registro anexado ao nome de domínio da zona. O registro DNSKEY é usado para armazenar chaves públicas que são usados no processo de autenticação de zonas do DNSSEC, pelo uso da chave privada. O *resolver* pode autenticar a zona usando a chave pública para validar a assinatura contida no RR*set*. Este registro é uma atualização do registro KEY descrito na RFC 2535.

A seção RDATA do registro DNSKEY contém informações de segurança como *flags* (sinalizadores), o tipo do protocolo, o tipo do algoritmo e a chave pública, e apresenta-se conforme a figura 8:

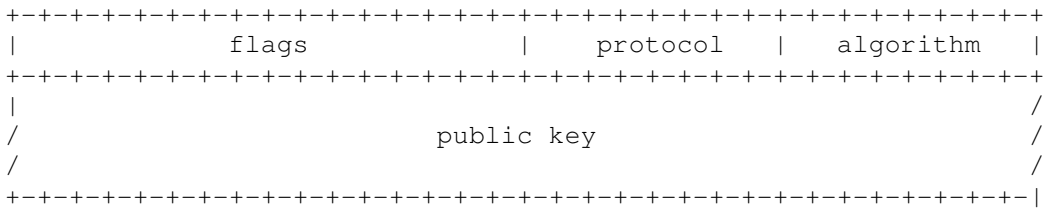

Figura 8: Formato da seção RDATA do registro DNSKEY

O registro DNSKEY também pode ser visto da seguinte forma:

```
sitecnet.com.br      259200 IN DNSKEY 256 3 1 (
      AQP7IGnN/Qc/7jm0s2j139e2QIABpAVRKLuP
      Iip5uH9r1kjkUvva+0XGEU0UNoByew==
) ; key id = 32882
```

onde 256 é o campo *flag*, o número 3 especifica o protocolo e o número 1 especifica o algoritmo. Entre parênteses encontra-se a chave pública propriamente dita, e ao final é mostrado o ID do registro.

O campo *flag* do registro DNSKEY é mostrado na figura 9:

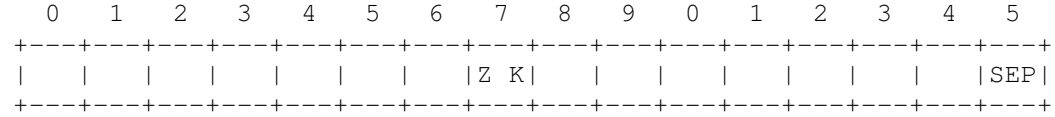

Figura 9: Formato do campo *flags* do registro DNSKEY

O *bit* 7 deste campo é o sinalizador da chave da zona (*Zone Key*), e comporta-se de maneiras diferentes, dependendo do valor que assumir:

- 1:o registro DNSKEY guarda a chave da zona DNS e seu próprio nome deve ser o nome da zona;
- 0: o registro DNSKEY guarda algum outro tipo de chave pública e não deve ser usado para verificar RRSIGs que cobre este RR*set*;
- O *bit* 15 é o mecanismo usado pelo DNSKEY para comunicar-se com outras chaves em outros RR*sets*, chamado SEP (*Securty Entry Point*). Se este *bit* tiver valor 1, ele será usado como chave SEP. Se tiver valor 0, não deve ser atribuído a este *bit* nenhum significado.

Os *bits* de 0 a 6 e de 8 a 14 são reservados e deves ser iguais a 0. Estes *bits* devem ter valor 0 na criação do registro e deve ser ignorado pelo destinatário.

O próximo campo da seção RDATA do registro DNSKEY é o chamado octeto de protocolos, e especifica qual a finalidade de uma determinada chave. Na RFC 2535, poderia assumir os seguintes valores:

- 0: reservada;
- 1: reservada para uso em conexões com a Segurança da Camada de Transporte (TLS – *Transport Layer Security*).
- 2: reservada para uso em conexões com mensagens de *e-mail*;
- 3: é usada com o DNSSec. Todas as chaves DNSSec terão um octeto de protocolo 3;
- 4: reservada para referir-se ao protocolo IPSEC e indica que esta chave é válida para uso em conjunto com este padrão de segurança.
- 5-254: disponível para atribuições futuras pelo IANA – *Internet Assigned Numbers Authority*.
- 255: é utilizada com qualquer protocolo que possa utilizar o registro DNSKEY.

Com a atualização da RFC 2535, este campo apenas pode assumir o valor 3. Se este valor for diferente de 3, este campo deve ser tratado como inválido durante a verificação da assinatura.

Após o octeto de protocolos, o campo a seguir define o número do algoritmo é utilizado pela zona. Os valores definidos são:

- 0: Reservado

- 1: RSA/MD5 [RFC 2537]. A RFC recomenda, mas não exige o uso do RSA/MD5;
- 2: Diffie-Hellman. A RFC 2535 torna o uso deste algoritmo opcional.
- 3: DAS/SHA-1 [RFC 2536]. A RFC 2535 torna obrigatório o suporte ao DSA, mas a RFC 2536 a torna opcional;
- 4: reservado para algoritmos baseados e curvas elípticas;
- 5: RSA/SHA-1. A RFC 3110 torna obrigatório o suporte a este algoritmo;
- 251: Disponível para atribuição do IANA;
- 252: Indireto.
- 253 e 254: Privados;
- 255: Reservado

O último campo do registro DNSKEY é a própria chave pública, codificada em base 64. O DNSSEC suporta chaves de diversos tamanhos, porém quanto maior chave, mais difícil torna-se descobrir a chave privada correspondente e mais tempo será necessário para assinar os dados da zona com a chave privada e verificá-los com a chave pública.[3]

4.3. Registro RRSIG

Como o registro DNSKEY armazena a chave pública da zona, a assinatura da chave privada deve ser armazenada em outro local. O registro de recurso RRSIG (*signature*) armazena essa chave em um RR*set*, que é um grupo de registro de recurso com o mesmo proprietário, classe e tipo[3]. O verificador pode usar este registro para autenticar RR*sets* de uma zona. Este registro é uma atualização do registro SIG descrito na RFC 2535.

A parte de dados específica do registro RRSIG (RDATA) é mostrada na figura 10:

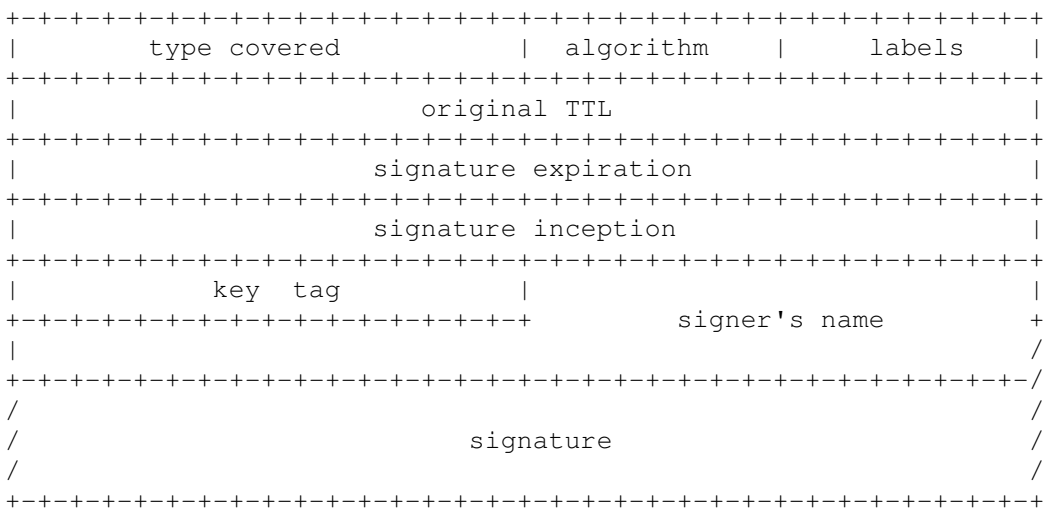

Figura 10: Formato da seção RDATA do registro RRSIG

O registro RRSIG também pode ser visto da seguinte forma:

```
sitecnet.com.br      259200      RRSIG  DNSKEY 1 3 259200 20030503125727
2003040325727 32882 net.
Q3F0w43P23vzUSvEoI3Q65L4Tw307JWz
cADghwT8y32DZu92rk/DYIubc7A= ).
```

O campo *type covered*, ou tipo contido, representado no exemplo acima pelo campo DNSKEY, indica o tipo do registro a ser assinado. Deve haver um registro RRSIG para tipo de registro contido na zona. O segundo campo, *algorithm*, que possui valor 1, indica o número do algoritmo utilizado, já descrito na seção 4.1.1. O terceiro campo, *labels*, indica quantos rótulos tem no nome do proprietário dos registros assinados. Por exemplo, para o nome **metron.sitecnet.com.br**, o número do rótulo seria 4. No exemplo acima, o campo tem valor 1. Atualmente, este campo permite até 127 rótulos.

O quarto campo, *TTL original*, indica o tempo de vida original dos registros do RR*set* que foi assinado. Isto serve para evitar problemas de autenticação em servidores que armazenam em *cache* o RR*set* contido nesse registro SIG, por decrementarem tos TTLs nos registros armazenados em *cache*, além de também evitar problemas de segurança causados pela manipulação de TTLs. No exemplo acima, é representado pelo valor 259200.

Os campos seguintes representam, respectivamente, o término e o início da validade da assinatura. São representados no formato YYYYMMDDHHMMSS, ou seja, no exemplo acima, a hora do término da validade da assinatura é após às 12:57:27 do dia 03/05/2003 (20030503125727), e tem início às 12:57:27 do dia 03/04/2003 (2003040325727). Geralmente, a hora do início e de término da validade da assinatura é a hora em que o programa foi executado

para assinar a zona. Após o término da validade, é preciso reassinar todas as assinaturas para mantê-las válidas.

O sétimo campo, *key tag*, ou identificador da chave, é um identificador de 16 *bits* gerado a partir da chave pública que corresponde à chave privada que assina a zona. No nosso exemplo tem o valor 32882.

O oitavo campo representa o nome do signatário (*signer´s name*), ou seja, o nome de domínio da chave pública que o destinatário deve utilizar para verificar a assinatura. No exemplo, contém o valor *net*. O transmissor não deve usar padrões de compressão de nomes do DNS quando são transmitidos pela rede.

O último campo, *signature*, é a assinatura digital. Este campo é composto pela seção RDATA, (formada por todos os campos RDATA, no próprio registro RRSIG, exceto o campo assinatura) e pelo conjunto de registro de recursos (RR*set*) dos registros (RRs) do campo *type covered*.

4.4. Registro NSEC

Ao enviar uma consulta a um servidor DNS, um cliente pode receber vários tipos de respostas[4]. Um dos tipos de resposta é a resposta negativa (*negative answer*), isto é, quando o servidor informa na resposta que o nome pesquisado não existe neste domínio, ou o nome pesquisado existe, mas o tipo de registro não confere. A não-existência de um nome em uma zona é indicada pelo registro NSEC (*Next SECure*) para um intervalo de nomes contendo o nome não-existente [15], resolvendo o problema de respostas negativas. Este registro é uma atualização do registro NXT descrito na RFC 2535.

O registro NXT definido na RFC 2535 era limitado a carregar no campo *Type Bit Map* dados sobre a existência dos primeiros 127 tipos indicados para o próximo nome do domínio, e reservava um *bit* para especificar um mecanismo de extensão desses tipos.

Este registro é usado para definir intervalos entre dois nomes de domínios consecutivos em uma zona e indica qual o próximo registro da zona, que deve ser ordenada de forma canônica (em ordem alfabética) durante o processo de assinatura. Segue um exemplo de uma zona ordenada de forma canônica:

[4] Ver os tipos de respostas na seção 2.2.3.

sitecnet.com.br
ns.sitecnet.com.br
metron.sitecnet.com.br
control.sitecnet.com.br
central.sitecnet.com.br

Com a zona ordenada de forma canônica, o formato do registro NSEC é mostrado na figura 11:

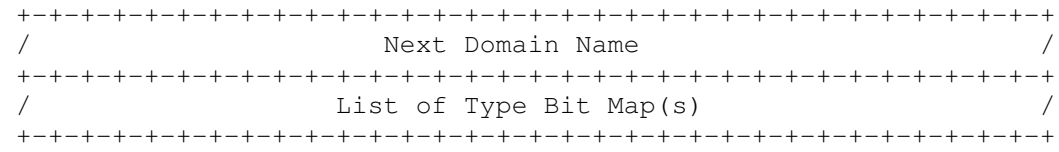

Figura 11: Formato da seção RDATA do registro NSEC

Este registro também pode ser visto da seguinte forma:

sitecnet.com.br IN NSEC ns.sitecnet.com.br NS SOA MX RRSIG DNSKEY

A seção RDATA do registro NSEC contém apenas o nome do próximo nome de domínio (*Next Domain Name*). Nomes próprios de RR*sets* que não são autoritativos para uma dada zona não deve ser listado no registro NSEC a menos que exista no mínimo 1 (um) RR*set* autoritativo em algum nome. O transmissor não deve usar compressão de nomes do DNS neste campo quando transmitir o registro NSEC.

O campo *List of type bit map(s)* indica a existência dos tipos indicados para o próximo registro informado. No exemplo acima, o nome **ns.sitecnet.com.br** é o próximo nome de domínio após o nome **sitecnet.com.br**, e este nome existe para os tipos NS, SOA, MX, RRSIG e DNSKEY.

O último registro NSEC de uma zona deve apontar para o primeiro registro da zona, já que não existe nenhum nome de domínio depois do último nome, assim:

central.sitecnet.com.br IN NSEC sitecnet.com.br NS SOA MX RRSIG DNSKEY

Assim, se uma consulta buscar por um nome de domínio que não existe na zona, um registro RRSIG que abrange todo o registro NSEC acompanha a resposta, autenticando a não-existência do nome de domínio ou do tipo de dados solicitado[3].

Uma conseqüência do uso deste registro é que um atacante pode pesquisar pelos registros NXT anexado a um nome de domínio da zona para localizar o em ordem alfabética o nome do próximo nome de domínio da zona e assim mapear todos os nomes de domínios da zona.

4.5. Registro DS

O registro DS (*Delegation Signer*) é envolvido na autenticação de delegações de zonas entre zonas pai e filhas. Ele refere-se a um RR DNSKEY pelo armazenamento dos caracteres da chave, do número do algoritmo e do "*digest*" do RR DNSKEY, e é usado no processo de autenticação do RR DNSKEY, para indicar que a delegação da zona é assinada digitalmente. Este registro aparece apenas no lado superior da delegação (zona pai).

A parte de dados específica do registro DS (RDATA) é mostrada na figura 12:

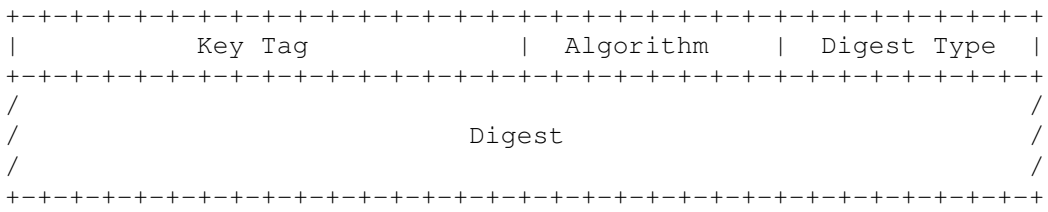

Figura 12: Formato da seção RDATA do registro DS

Este registro também pode ser visto da seguinte forma:

```
sitecnet.com.br      172800 IN DS 32882 1 1 (33CC2CCEC70C72C153FFF495D563
```

O primeiro campo (*Key Tag*) lista os sinalizadores chave do registro DNSKEY referenciado pelo registro DS. Este campo é idêntico ao mesmo campo do registro RRSIG. O campo *Algorithm* refere-se ao número do algoritmo do RR DNSKET referenciado pelo registro DS, e pode conter os mesmos valores que o campo *algorithm* dos registros DNSKEY e RRSIG podem conter.

O RR DS consulta a um registro DNSKEY incluindo um sumário daquele registro DNSKEY RR. e identifica o algoritmo usado para construir o *digest*. O valor 1 representa o algoritmo SHA-1 e a RFC 4034 torna seu uso obrigatório.

O campo *digest* é calculado pela concatenação da forma canônica do FQDN do registro DNSKEY com o RDATA deste registro. O tamanho do *digest* depende do algoritmo e do tamanho do registro DNSKEY.

4.6. Cadeia de confiança

Cada zona segura possui seu par de chaves, e sua chave pública é armazenada no registro DNSKEY, para permitir que outras zonas verifiquem a sua assinatura digital (RRSIG). Porém,

como uma zona pode certificar-se que esta chave pública realmente pertence à zona da qual ela se diz ser, ou seja, sendo alterada por um atacante?

Esta chave pública deve ser certificada por alguma entidade superior, a zona pai, que confirme que esta chave realmente pertence ao seu domínio, e não a outro. Quando o par de chaves é gerado, a zona envia sua chave pública à zona pai para que este a certifique. A zona pai assina o registro DNSKEY da zona filha com sua chave privada e o envia de volta para que seja incluído na zona.

Assim, cada zona realiza o mesmo processo, até que atinja os servidores raiz, ou seja, a chave pública da zona raiz será configurada em todos os servidores de nomes, e cada servidor poderá certificar-se que as assinaturas a eles enviadas são realmente de quem as enviou. Este processo é chamado de cadeia de confiança, é estabelecida pela verificação do RR*set* DNSKEY, do RR*set* DS da zona pai e do RR*set* DNSKEY no filho. Isto é criptograficamente equivalente ao uso de apenas registros DNSKEY [34].

4.7. Como o DNSSEC tenta resolver os problemas do DNS

O DNSEC utiliza a criptografia de chave pública e assinatura digital para minimizar os problemas do DNS. Alguns deles foram explicados no capítulo 3, e aqui será explicado como o DNSSEC resolve alguns deles. Abaixo segue uma figura que ilustra em que ponto do fluxo das informações do DNS as vulnerabilidades se encontram:

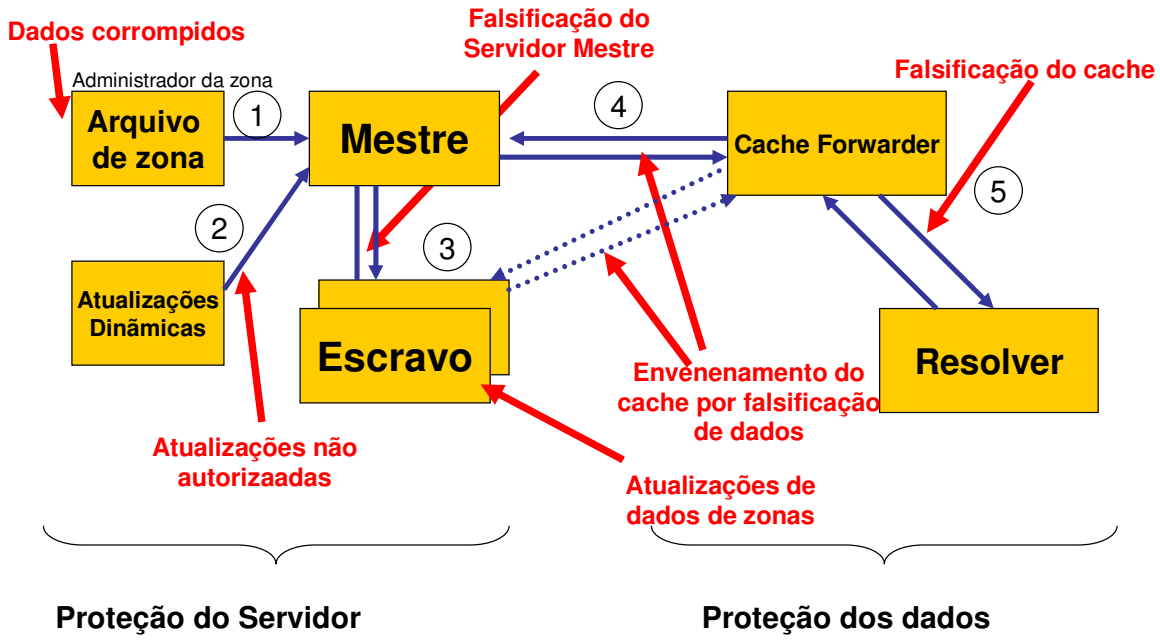

Figura 13: Vulnerabilidades do DNS

O protocolo UDP é substituído pelo TCP, já que todos os pacotes do DNS precisarão de recursos que o UDP não dispõe. O TCP é orientado a conexão, ou seja, trata de perda de pacotes, retransmissões e reordenação dos pacotes, além do uso de mecanismos próprios para controle de fluxo. Assim, os ataques que visam às vulnerabilidades inerentes ao protocolo UDP não serão mais realizados.

O DNSSEC dispõe da checagem da integridade dos dados fim-a-fim, ou seja, desde sua origem até seu destino, sendo a melhor solução para problemas de interceptação de pacotes durante uma operação de pesquisa DNS, pois executa todas as checagens de assinaturas previstas no DNSSEC, pode re-assinar os pacotes em caso de ataques de interceptação de pacotes e ainda pode utilizar o TSIG para assegurar-se da integridade da comunicação entre um cliente e o servidor.

As transferências de zonas e atualizações dinâmicas, onde ocorrem as transações das mensagens DNS, são protegidas por um mecanismo de autenticação destas transações, chamado TSIG (*Transaction Signature*), que é utilizado para a verificação da autenticidade das mensagens DNS, especialmente as respostas e as atualizações.

O TSIG utiliza criptografia simétrica, ou seja, a mesma chave que é usada para criptografar é a mesma usada para descriptografar. Esta chave é chamada de chave secreta compartilhada. As assinaturas TSIG são verificadas no momento do recebimento dos pacotes e depois são descartados, servindo apenas para verificar que o remetente da mensagem tem uma chave compartilhada e que esta mensagem não foi modificada após seu envio. O processo de autenticação é realizado através do *hash-based message authentication codes* (HMAC), que utiliza como parâmetros a mensagem de entrada e a chave secreta. Por ser um meta-registro que nunca aparece nos dados da zona e nunca é armazenado, sendo incluído apenas na mensagem do DNS, ele não é utilizado pelo DNSSEC, e por esta razão é mais utilizado nas transferências de zona e atualizações dinâmicas.

O DNSSEC em si não oferece proteção contra o ataque *server compromising*, pois ele não tem como proteger um servidor de nomes de ser atacado. Porém, se isto acontecer e um atacante conseguir alterar dados da zona, gerar seu próprio par de chaves e re-assinar a zona, o DNSSEC oferece a proteção da chamada cadeia de confiança, onde uma zona superior àquela contida no servidor invadido certifica a chave pública de sua zona filha. Através do registro DS, contido no

DNSSEC e ausente na definição padrão do DNS, a divulgação destas alterações podem ser inibidas.

O DNS é vulnerável ao *spoofing* e ao *cache poisoning* porque não utiliza autenticação dos dados. Com a utilização do DNSSEC, autenticando a origem dos dados informados a um cliente, este poderá certificar-se de que o nome do domínio referente a um determinado endereço IP realmente pertence àquele endereço IP, através da assinatura digital e da chave pública daquela zona.

4.8. Os problemas do DNSSEC

A necessidade de proteger as informações contidas em mensagens DNS e as conseqüências trazidas pela falta de segurança neste protocolo levaram ao surgimento das extensões de segurança do DNS, que vieram para oferecer principalmente segurança aos dados utilizando criptografia e assinaturas digitais.

Porém, o DNSSEC também tem suas limitações e suas conseqüências. Por exemplo, não protege contra ataques ao protocolo, apenas os detecta, mas impede que as informações sejam modificadas pelo atacante.

Em um ataque do tipo DoS, o DNSSEC não pode fazer absolutamente nada; pelo contrário, pode transformar-se em um "amplificador" deste ataque.

Por exemplo, um atacante, utilizando técnicas de ataques DoS baseado em operações criptográficas, pode tentar esgotar os recursos do alvo atacado. O atacante pode consumir os recursos de um *name server* no código de validação da assinatura de um *resolver* com a alteração de RRs SIG em mensagens de resposta, ou construindo seqüências de assinaturas complexas desnecessárias. Em servidores com suporte a atualizações dinâmicas, pelo envio de seqüenciais mensagens de atualizações, um atacante força o servidor a re-assinar alguns RR*sets* com mais freqüência que o necessário.

Em Janeiro de 2006, a Verisign sofreu um ataque do tipo DoS. Geralmente, estes ataques vêm de *computer´s bot*, ou seja, computadores comandados remotamente por *hackers*. Desta vez, os ataques foram originado por outros servidores DNS. Segundo Dan Kaminski, pesquisador de segurança focado em mecanismos de análise e entendimento de grandes redes, "a barreira foi diminuída; pessoas com algum recurso podem lançar ataques potencialmente perigosos", informando que o DNS é o maior vetor para ataques tipo DDoS.[35]

O mecanismo do DNSSEC também não oferece segurança contra as vulnerabilidades referente a transações tipo transferência de zonas e atualizações dinâmicas. Estas transações contam com uma proteção especial, descrita na RFC 2137. A autenticação de mensagens DNS é descrita na RFC 2845.

O conjunto de documentos que formam o núcleo das extensões de segurança do DNS é formado pelas RFC´s 4033, 4034 e 4035. Devido a estas mudanças, o DNSSEC não fornece confidencialidade.

Mais alguns problemas do DNSSEC são listados a seguir [27]:

- Deficiente estrutura de atualização de chaves públicas: a infra estrutura de chaves públicas é necessária para outros protocolos, como o IPSEC, ou *e-commerce*; se uma chave pública for violada, deve-se gerar rapidamente um novo par de chaves (privada/pública), porém, haverá incompatibilidade com a chave antiga que está armazenada no *cache* de outros servidores.
- Assinatura de grandes zonas: as TLD´s, por exemplo, contém um banco de dados com milhões de delegações. Para assinar toda esta zona,são necessários várias horas e dias, dependendo do hardware e software disponível, sem contar que assinar uma zona aumenta em cerca de sete vezes seu tamanho.
- Performance do tráfego: com o uso de autenticação, o protocolo TCP faz-se necessário, pois o cliente reenviará uma requisição para resposta que foi truncada, por ser maior que 512 bytes. O protocolo TCP gera mais tráfego para gerencia a transmissão de dados pela rede. Para assinar todas as requisições, a maior parte do tráfego DNS utilizaria TCP.
- Requisitos de hardware: a validação de assinaturas também consome tempo de CPU e memória, e o tempo de encriptação/decriptação depende do algoritmo utilizado. Assim, o custo de processamento para que todas as requisições e suas respectivas respostas sejam criptografadas aumenta consideravelmente, tornando-se um dos maiores desafios para a implantação do DNSSec.

Um estudo publicado pelo RIPE NCC / NLnet Labs [38], em Outubro de 2005, mede quais os recursos necessários, em termos de máquinas, para a implementação do DNSSec. Foram medidos recursos como uso de CPU, memória e banda consumida. O resultado do estudo mostra que a medição do impacto da implementação do DNSSec depende de muitas variáveis, e que a implementação em larga escala torna-se difícil, pois, segundo os resultados obtidos, muitos

servidores de nomes teriam que atender aos recursos de máquinas demandados pela implementação do DNSSec.

Outro fator que dificulta a implementação do DNSSec são os softwares que o suportam. O mais utilizado é o BIND (*Berkeley Internet Name Domain*). Sua popularidade deve-se principalmente por ser um *software* gratuito, originado de um projeto da Universidade da Califórnia e ser incluído em sistemas operacionais como Solaris, Mac OS X, Linux e BSD, por exemplo, além de possuir versão para Windows. As versões mais novas do BIND suportam a implementação do DNSSec, com a habilidade de assinar zonas, gerar diferentes tipos de chaves e manipular zonas assinadas e RRs. Suporta também transações seguras utilizando o TSIG.

Porém, o BIND possui algumas vulnerabilidades, mostradas a seguir:

- *Cache Poisoning*: quando atacado desta forma, o servidor BIND é forçado a realizar suas consultas de DNS em um servidor de DNS especificado pelo intruso. Assim, o intruso consegue passar resultados falsos para as consultas do servidor atacado, que por sua vez, passará esses resultados para seus clientes. Então, os clientes estarão acessando sites falsos (por conta do redirecionamento do mapeamento IP no servidor remoto) o que pode acarretar em sérios problemas;
- Pesquisa reversa: neste tipo de ataque, o intruso pode conseguir acesso de root no servidor atacado ou até causar a parada total do sistema. Por padrão, os servidores não respondem a pesquisas falsas, porém, se estiverem configurados para responder a essas pesquisas, podem sofrer este tipo de ataque;
- DoS: as versões mais atuais do BIND conseguem trabalhar muito bem com a questão do DoS, porém versões mais antigas podem cair facilmente nesta armadilha e se tornarem inoperantes. Este seria um dos vários motivos para utilizar as últimas versões estáveis do *software*;

Existem vários outros softwares que implementam o DNS som suporte ao DNSSec, como o NSD (*Name Server Daemon*), por exemplo, não é tão complexo quanto o BIND, mas também é *open source*, tem suporte ao DNSSec e pode ser utilizado em cenários simples. O Microsoft Server 2003 apenas suporta armazenamento e carga de zonas assinadas, mas não é capaz de validar e assinar RRSIGs. Pode-se citar também como exemplos de softwares que implementam o DNS, o DNSJava, NET::DNS::SEC, Dents (http://www.dents.org), Djbdns (http://cr.yp.to/djbdns.html), MaraDNS (http://www.maradns.org), CustomDNS (http://customdns.sourceforge.net/), Ibnamed (http://www.stanford.edu/~schemers/docs/lbnamed/lbnamed.html/) e Ibdns [40][41].

5. Conclusão

Segundo dados da ICANN de novembro de 2004, são realizadas cerca de 20 bilhões de resoluções DNS por dia, mais de 700 milhões de usuários utilizam a Internet todo dia e 313 milhões de endereços IPV4 (endereços IP versão 4, atualmente utilizado) foram alocados desde 1999 [36].

Com estes dados, podemos observar que o sistema de nomes de domínio é um dos principais mecanismos do funcionamento da Internet. Porém, suas definições e funcionamento contêm falhas que podem levar a sérias conseqüências, como a falsificação da informação, redirecionamento do tráfego e roubo de dados de usuários.

As extensões de segurança do DNS, chamada de DNSSEC tentam minimizar estas falhas, utilizando-se da criptografia para codificar as informações e garantir sua autenticidade. Mas este novo mecanismo também apresenta falhas e motivos que dificultam sua implantação.

Os custos da adoção do DNSSEC passam por diversos fatores, como o aumento da performance dos atuais servidores DNS, incluindo aumento da memória, CPU e banda destes. Custos adicionais como *software*, treinamento de pessoal especializado, gerenciamento de chaves, interação com entidades certificadoras e mantenedoras dos servidores *root* do DNS devem ser analisados.

Atualmente, existem grupos de parceiros que desenvolvem e divulgam o uso do DNS, como o *DNSSEC Deployment Initiative Coordination Partners*, que incluem empresas e órgãos como o NIST (EUA), Sparta, INC, Shinkuro, INC (EUA), SRI (EUA), NLnetLabs (Holanda). Outros órgãos, como PIR (EUA), RIPE (Europa - Londres), TLD.se (Suécia), o TLD experimental.nl.nl (Holanda) e Verisign já adotaram e ainda desenvolvem projetos para o DNSSEC.

A adoção do DNSSEC em larga escala é um processo lento e poderá levar alguns anos até ser totalmente implantado, pois ainda necessita de muitas melhorias e pesquisa para minimizar seus custos. Sua adoção em larga escala enfrentaria grandes dificuldades, pois o processo de resolução de validação de nomes tornaria-se muito mais lento, mas futuramente será um recurso mais seguro, necessário e confiável de se utilizar.

6. Referências

[1] DAVIDOWICZ, Diane. "*Domain Name System Security*", 1999. Disponível em (http://compsec101.antibozo.net/papers/DNSSec/DNSSec.html).

[2] VERISIGN. "*Soluções de Garantia de DNS*", 2003. Disponível em (http://www.verisign.com).

[3] ALBITZ, Paul, LIU, Cricket. – "*DNS e BIND*", 4ª ed., Rio de Janeiro, Campus 2001.

[4] COMER, Douglas E. "*Interligação em rede com TCP/IP*", Rio de Janeiro, Campus, 1998.

[5] COMER, Douglas E. "*Redes de computadores e Internet*: Abrange Transmissão de Dados, Ligação Inter-redes e Web", 2ª ed, Rio de Janeiro, Bookman, 2001.

[6] SACRAMENTO, Vagner, BATISTA, Thais, LEMOS, Guido e MONTEIRO, Claudio. "*Uma Estratégia de Defesa contra Ataques de DNS Spoofing*" *In* 3o. Simpósio de Segurança em Informática (SSI), São José dos Campos - SP, pp 183-192, Outubro 2001. Disponível em (http://www.dimap.ufrn.br/~thais/Publicacoes/ssi2001.pdf).

[7] PFLEGER, Jurgen. "*DNSSEC Resolver Algorithm*", Master Thesis, Wien, Austria, 2003. Disponível em: (http://mitglied.lycos.de/pflegerjuergen/download/DNSSEC_Resolver.pdf)

[8] ICANN. *Lista de Registros de domínios de primeiro nível*. Disponível em (http://www.icann.org/registries/listing.html)

[9] CHAKRABARTI Anirban and MANIMARAN G., "Internet Infrastructure Security: A Taxonomy", IEEE Networks, vol. 16, no. 6, pp. 13-21, Nov./Dec. 2002. Disponível em (http://vulcan.ee.iastate.edu/~gmani/personal/papers/journals/IEEE-Net.pdf)

[10] STEWART Joe, "*DNS Cache Poisoning – The Next Generation*", LURHQ Threat Intelligence Group. Disponível em (http://www.lurhq.com/cachepoisoning.html).

[11] SCHUBA, Christoph, "*Addressing Weakness in the Domain Name System Protocol*", Thecnical Report, Department of Computer Sciences, Pardue University, 1993. Disponível em (http://ftp.cerias.purdue.edu/pub/papers/christoph-schuba/schuba-DNS-msthesis.pdf).

[12] SAX, Doug., "*DNS Spoofing (Malicious Cache Poisoning)*". 12 Nov. 2000. 15 Dec 2002 Disponível em: (http://rr.sans.org/firewall/DNS_spoof.php).

[13] Microsoft Technet, Microsoft Windows Server 2003 Tech Center, "*Atualização Dinâmica*", 2001. Disponível em (http://www.microsoft.com/technet/prodtechnol/windowsserver2003/pt-br/library/ServerHelp/e760737e-9e55-458d-b5ed-a1ae9e04819e.mspx)

[14] MARIZ, Dênio, "*Ataques & Ferramentas – Algumas ferramentas e técnicas de ataque*". Notas de Aula, 2005

[15] RFC 2535 - Eastlake 3rd, D., "*Domain Name System Security Extensions*", Março 1999.

[16] RFC 2536 - Eastlake 3rd, D., "*DSA Keys and SIGs in the Domain Name System (DNS)*", Março 1999.

[17] RFC 1034 - Mockapetris, P., "*Domain Names - Concepts and Facilities*", Novembro de 1987.

[18] RFC 1035 - Mockapetris, P., "*Domain names - Implementation and Specification*", Novembro de 1987.

[19] RFC 3225 – Conrad, D., "*Indicating Resolver Support of DNSSEC*", Dezembro de 2001.

[20] RFC 3833- Atkins, D., "*Threat Analysis of the Domain Name System*", Agosto de 2004.

[21] RFC 2136 – Vixie, P. “*Dynamic Updates in the Domain Name System (DNS UPDATE*”, Abril de 1997.

[22] RFC 2137 - Eastlake 3rd, D., “*Secure Domain Name System Dynamic Update*”, Abril de 1997

[23] RFC 2671 - Vixie, P., "*Extension Mechanisms for DNS (EDNS0)*", RFC 2671, Agosto de 1999.

[24] JUNIOR, José. H. T., MOURA, José A. B., SAUVÉ, Jácques P., “*Disponibilização de Nomes de Diretórios Distribuídos através do DNS*”, DSC/UFPB. Disponível em (http://www.las.ic.unicamp.br/paulo/wais96/papers-tech/helvecio-dns-rev.pdf).

[25] LIMA, Marcelo B., “*Provisão de Serviços Inseguros Usando Filtros de Pacotes com Estados*”, Dissertação de Mestrado, Instituto de Computação, Universidade Estadual de Campinas, 2000. Disponível em: (http://www.kcmelo.com/Files/Dissertacao_MBL.pdf).

[26] POUW, Kessje D., “*Segurança na Arquitetura TCP/IP: de Firewalls a canais seguros*”, Dissertação de Mestrado, Instituto de Computação, Universidade Estadual de Campinas, 1999. Disponível em: (http://www.las.ic.unicamp.br/paulo/teses/19990208-MSc-Keesje.Duarte.Pouw-Seguranca.na.arquitetura.TCPIP-De.firewalls.a.canais.seguros.pdf).

[27] NEMETH, Evi. “*Securing the DNS*”. ;login: The Magazine of Usenix & Sage, Nov. 2000. Disponível em: (http://www.usenix.org/publications/login/2000-11/pdfs/dns.pdf)

[28] RFC 3110 - Eastlake 3rd, D., "*RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS)*",.Maio de 2001.

[29] RFC 3757 – KOLMAN, O., "*Domain Name System KEY (DNSKEY) Resource Record (RR) Secure Entry Point (SEP) Flag* ", Abril de 2004.

[30] RFC 4033 – ARENDS, A., “*DNS Security Introduction and Requirements*”. Março de 2005.

[31] RFC 4034 – ARENDS, A., “*Resource Records for the DNS Security Extensions*”. Março de 2005.

[32] RFC 4035 – ARENDS, A., “*Protocol Modifications for the DNS Security Extensions*”. Março de 2005.

[33] RFC 3845 – SCHLYTER, J., “*DNS Security (DNSSEC) NextSECure (NSEC) RDATA Format*”, Agosto de 2005.

[34] RFC 3658 – GUDMUNDSSON, O., “*Delegation Signer (DS) Resource Record (RR)*”, Dezembro de 2003.

[35] EVERS, J., ”*DNS Server´s do hacker´s dirty work*”. CNET News.com, Março de 2006. Disponível em: (http://news.com.com/DNS+servers+do+hackers+dirty+work/2100-7349_3-6053468.html).

[36] ICANN – Internet Corporation for Assigned Names and Numbers, “*Plano Estratégico para ICANN 2003-2004 a 2006-2007*”, Novembro de 2004. Disponível em: (http://www.icann.org/strategic-plan/strategic-plan-16nov04.pdf)

[37] FECHINE, Joseana. “*Criptografia, protocolos seguros e suas aplicações*”, Notas de aula, partes I,II,III e IV, 2005.

[38] OLSEN, Camila. “*Security issues relating to the use of UDP*”, Research on Topics in Information Security, Option 1, GSEC Certification Practical Assignment Version 1.4 b, SANS Institute, 2004.

[39] KOLMAN, O. “*Measuring the resource requirements of DNSSEC*”, RIPE NCC / NLnet Labs, RIPE-352, Outubro de 2005.

[40] HOMBLAD, J. "*The Evolving Threats to the Availability and Security of the Domain Name Service*", SANS GIAC/GSEC Practical, Outubro de 2003.

[41] KARLOV, A. "*DNSSEC: DNS Security Extensions*", École Polytechnique Fédérale de Lausanne, Maio de 2004.