UNIVERSIDADE DO ESTADO DE SANTA CATARINA

CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT

BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FELIPE DE SOUZA LONGO

HONEYPOT PARA SERVIDORES DNS RECURSIVOS: ADAPTAÇÃO, COLETA E ANÁLISE DE RESULTADOS

JOINVILLE - SC

2015

FELIPE DE SOUZA LONGO

HONEYPOT PARA SERVIDORES DNS RECURSIVOS:

ADAPTAÇÃO, COLETA E ANÁLISE DE RESULTADOS

Trabalho de Conclusão apresentado ao Curso de Ciência da Computação, do Centro de Ciências Tecnológicas, da Universidade do Estado de Santa Catarina, como requisito parcial para obtenção de grau de Bacharel em Ciência da Computação.

Orientador: Rafael Rodrigues Obelheiro

JOINVILLE - SC

2015

FELIPE DE SOUZA LONGO

HONEYPOT PARA SERVIDORES DNS RECURSIVOS: ADAPTAÇÃO, COLETA E ANÁLISE DE RESULTADOS

Trabalho de Conclusão de Curso, Bacharelado em Ciência da Computação/ Centro de Ciências Tecnológicas/ Universidade do Estado de Santa Catarina, Bacharel em Ciência da Computação.

Banca Examinadora

Orientador: _____

Prof. Dr. Rafael Rodrigues Obelheiro
Universidade do Estado de Santa Catarina

Membro: _____

Prof. Dr. Charles Christian Miers
Universidade do Estado de Santa Catarina

Membro: _____

Prof. Dr. Guilherme Piêgas Koslovski
Universidade do Estado de Santa Catarina

Joinville - SC, 16/11/2015

RESUMO

O sistema DNS (*Domain Name System*) é um dos pilares para o funcionamento das aplicações e serviços na Internet, tendo como principal funcionalidade a tradução de nomes em endereços IP. No entanto, o DNS possui certas vulnerabilidades estruturais de segurança, que permitem que o mesmo seja atacado ou usado como instrumento de ataque a terceiros. A maior parte das ameaças pode ser contida pela configuração adequada de servidores DNS, que torna os ataques ineficazes. Todavia, isso limita a capacidade de se observar o comportamento dos atacantes e assim inferir seus objetivos. O presente trabalho apresenta uma arquitetura de um *honeypot* específico para DNS (denominado DNSpot), que monitora e registra o tráfego enviado a um servidor DNS recursivo aberto, permitindo sua análise posterior. O DNSpot foi implantado na rede da UDESC, tendo recebido mais de 4 milhões de requisições ao longo de 49 dias. Uma análise do tráfego observado revelou a alta incidência de ataques de negação de serviço distribuída (DDoS – *Distributed Denial of Service*), e também mostrou aspectos pouco conhecidos do comportamento de atacantes.

Palavras-chave: *Domain Name System (DNS). Honeypot*. Segurança de redes.

ABSTRACT

The Domain Name System (DNS) is one of the pillars for the operation of applications and services on the Internet, with its main function being the translation of names into IP addresses. However, the DNS has certain structural security vulnerabilities which allow it to be attacked or used as a tool for attacking third parties. Most threats can be contained by the proper configuration of DNS servers, resulting in ineffective attacks. However, this approach limits the ability to observe the behavior of the attackers and thus infer their goals. This work presents an architecture for a DNS-specific honeypot (called DNSpot), which monitors and records traffic sent to an open recursive DNS server for later analysis. Our DNSpot was deployed on the UDESC networking, receiving more than 4 million requests over 49 days. An analysis of the observed traffic revealed a high incidence of distributed denial of service (DDoS) attacks, and also showed some lesser-known aspects of attacker behavior.

Keywords: Domain Name System (DNS). Honeypot. Network security.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

DNS	*Domain Name System*
DoS	*Denial of Service*
DDoS	*Distributed Denial of Service*
C&C	*Command and Control*
FQDN	*Fully Qualified Domain Name*
TLD	*Top Level Domain*
DIG	*Domain Information Dangler*
UDP	*User Datagram Protocol*
TCP	*Transmission Control Protocol*
NAT	*Network Adress Translation*
RR	*Resource Record*

SUMÁRIO

1 INTRODUÇÃO

Um dos pilares da Internet é o DNS (*Domain Name System*), uma base de dados hierárquica e distribuída que mapeia nomes em vários tipos de dados, como endereços IP (CONRAD, 2012; MOCKAPETRIS, 1987b). O DNS permite que usuários e aplicações usem “nomes amigáveis” (como www.udesc.br), ou FQDN (*Fully Qualified Domain Name*), para se referir a *hosts* e outros recursos em redes TCP/IP. Assim como vários outros componentes da Internet – de IP (GONT, 2008), TCP (GONT, 2009) a aplicações web (ENISA, 2014) – o DNS também sofre com ameaças e ataques; ataques ao DNS podem afetar a sua segurança, estabilidade e resiliência, e também têm sido utilizados como ponto de partida para atacar outros usuários da Internet, como em ataques de negação de serviço e DNS *Pharming* (redirecionamento de um domínio para um IP fraudulento) (CONRAD, 2012; ENISA, 2015). Por exemplo, episódios de indisponibilidade de grande impacto causados por ataques ao DNS já afetaram empresas como Microsoft (SECURE64, 2001), Amazon (SLAYER, 2009), além de todos os nomes no domínio nacional da Alemanha (.de) (SECURITY WEEK, 2010).

Para garantir escalabilidade e facilitar a administração do espaço de nomes, o DNS é descentralizado, e requer a cooperação de um grande número de servidores. Cada componente envolvido nas requisições aumenta o potencial de perigo e riscos ao DNS. Problemas de segurança e estabilidade no DNS podem ter impacto sobre um elevado número de usuários, com consequências potencialmente severas, que incluem o redirecionamento de usuários para sites falsos, a violação da privacidade dos usuários, e a incapacidade de usar os recursos da rede por indisponibilidade do serviço (ARENDS, 2005a; CONRAD, 2012). Um problema relacionado é o uso indevido de servidores DNS mal configurados para facilitar ataques de negação de serviço contra terceiros (CERT.br, 2013; CERT.br, 2015).

A maior parte das ameaças pode ser contida pela configuração adequada de servidores DNS, que torna os ataques ineficazes. As configurações se baseiam na ideia de restringir quem pode acessar o servidor, negando o acesso de clientes não autorizados (CHANDRAMOULI; ROSE, 2013; PESCATORE, 2014; ISC, 2015; TEAM CYMRU, 2012). Todavia, isso limita a capacidade de se observar o comportamento dos atacantes e assim inferir seus objetivos e métodos de ataque empregados, pois os ataques são bloqueados logo no seu início. Muitas vezes o que se observa é

apenas uma tentativa de acesso não autorizado a um servidor, sem que se saiba a natureza do acesso pretendido, pois o bloqueio de uma interação não autorizada logo no seu início impede que as interações subsequentes sejam realizadas e observadas, de modo a se concluir os objetivos da interação inicial. Perguntas que ficam ao se confrontar com um ataque frustrado são: E se o ataque/transação inicial tivesse sido bem sucedido, o que teria acontecido? Qual seria a extensão das consequências do ataque?

Essas são perguntas cujas respostas seriam úteis para serem analisadas de modo a aperfeiçoar as práticas de segurança, entender melhor os objetivos e motivações dos atacantes, conhecer pontos críticos, vulnerabilidades ou alvos para se prevenir de futuros ataques. Então, de que maneira observar o comportamento dos atacantes para coleta e análise de dados, sem comprometer a rede e os dados da instituição?

Um dos métodos para observar o comportamento malicioso de potenciais atacantes a um dado sistema é usando um *honeypot*. Um *honeypot* é um recurso computacional de segurança, cujo objetivo principal é ser sondado, atacado ou comprometido. Dessa maneira pode-se observar o comportamento dos atacantes e inferir dados sobre os mesmos (CERT.br, 2007; STEDING-JESSEN *et al*, 2008).

Desse modo, este trabalho de conclusão de curso apresenta a concepção e implementação de um *honeypot* específico para servidores de DNS. Essa ferramenta, chamada de DNSpot, foi usada para observar as interações de usuários não autorizados com um serviço DNS recursivo na rede da UDESC, com o propósito inicial de coletar e analisar os dados para descobrir os objetivos/motivações dos mesmos, bem como os métodos de ataque empregados. O DNSpot preenche uma lacuna: de um lado, não existe nenhum *honeypot* específico para a monitoração abrangente de tráfego DNS, limitando a análise de ataques envolvendo o serviço a sistemas em produção, muitas vezes *post-mortem*; de outro, as funcionalidades oferecidas pelos servidores DNS existentes não são suficientes para, ao mesmo tempo, permitir a observação de interações suspeitas e evitar que essas interações causem dados ao próprio servidor ou a terceiros.

1.1 OBJETIVOS

O objetivo geral do trabalho é construir e implantar um *honeypot*, denominado DNSpot, para observar e analisar as interações de usuários não autorizados com um serviço DNS recursivo.

Para alcançar o objetivo geral foram definidos os seguintes objetivos específicos:

- Fazer uma revisão dos riscos de segurança e vulnerabilidades presentes em um sistema DNS, e suas consequências caso exploradas por usuários mal intencionados.
- Fazer um levantamento sobre os principais conceitos de *honeypot*s e *honeynets*.
- Realizar uma revisão de trabalhos correlatos.
- Estudar uma implementação de código aberto de um servidor DNS recursivo.
- Definir requisitos, projetar e implementar alterações/adaptações no servidor DNS para agir como um *honeypot*.
- Implementar o sistema em produção para coleta de dados.
- Analisar os resultados obtidos para adquirir informações sobre as interações realizadas com o serviço.

1.2 ORGANIZAÇÃO DO TRABALHO

O restante deste documento está organizado da seguinte maneira. No Capítulo 2 é feita uma revisão do DNS, seu funcionamento, formato das mensagens DNS, vulnerabilidades estruturais e as consequentes ameaças. O Capítulo 3 apresenta conceitos sobre *honeypots*, como eles são empregados na coleta de informações sobre usuários e atividades maliciosas e algumas ferramentas que implementam esse recurso, e também define os requisitos necessários a um *honeypot* específico para DNS. O Capítulo 4 descreve a arquitetura e implementação propostas para satisfazer os requisitos identificados. O Capítulo 5 traz uma análise de dados coletados com o DNSpot ao longo de 49 dias de operação. Por último, no Capítulo 6 são apresentadas as conclusões do trabalho, bem como perspectivas de trabalhos futuros.

2 DNS – DOMAIN NAME SYSTEM

O presente capítulo tem como objetivo oferecer uma revisão sobre o DNS, cobrindo os aspectos necessários ao entendimento deste trabalho. Inicialmente são apresentados os conceitos básicos e o funcionamento do sistema DNS. Posteriormente, são discutidos aspectos de segurança, incluindo as vulnerabilidades estruturais, as ameaças que visam a explorá-las, e as extensões de segurança para o DNS (DNSSEC).

2.1 DEFINIÇÃO

O *Domain Name System* (DNS) (MOCKAPETRIS, 1987a; 1987b) é tanto um conjunto de protocolos quanto um conjunto global de servidores distribuídos responsáveis por efetuar resolução de nomes na Internet (CONRAD, 2012; SCHUBA, 1993). O sistema DNS é um dos principais componentes da Internet, sendo utilizado pela maioria dos serviços e aplicações. Sua principal função é oferecer um serviço de tradução de nomes alfanuméricos preferidos por usuários em endereços IP usados por máquinas, como, por exemplo, a tradução do domínio www.udesc.br para o endereço 200.19.105.194 (ALBITZ; LIU, 2006; CHANDRAMOULI; ROSE, 2013; CONRAD, 2012; PESCATORE, 2014).

O DNS é um repositório de dados distribuído. Essa estrutura permite que haja controle local de cada segmento do repositório, mas de modo que a informação em cada segmento esteja disponível para toda a Internet num esquema de cliente/servidor. Dessa maneira basta que administradores gerenciem seus segmentos e servidores de nomes para manter as informações atualizadas para o restante da rede. Isso permite também a redistribuição do gerenciamento das informações de domínios em várias organizações especializadas. Robustez e desempenho são alcançadas através de replicação dos dados e armazenamento local por *cache* (ALBITZ; LIU, 2006; SCHUBA, 1993), respectivamente.

Programas chamados de servidores de nomes constituem parte do esquema cliente/servidor do DNS. Eles contêm informações a respeito de segmentos do repositório e garantem que essas informações estejam acessíveis e disponíveis para clientes, chamados de *resolvers* ou resolvedores. Já os *resolvers* são bibliotecas

com rotinas que criam requisições/consultas e as enviam pela Internet para um servidor de nomes (ALBITZ; LIU, 2006; PESCATORE, 2014; SCHUBA, 1993).

Como visto, o sistema DNS oferece o importante serviço de traduzir domínios em endereços IP. A estrutura de repositório de dados distribuído permite que seja criada uma hierarquia entre os domínios, de modo que cada entidade administrativa responsável por um dado segmento da hierarquia gerencie e garanta a disponibilidade das informações atualizadas em seus servidores de nomes para os clientes requisitantes que necessitarem. A estrutura dessa hierarquia e seus componentes são definidos na Seção 2.2.

2.2 HIERARQUIA

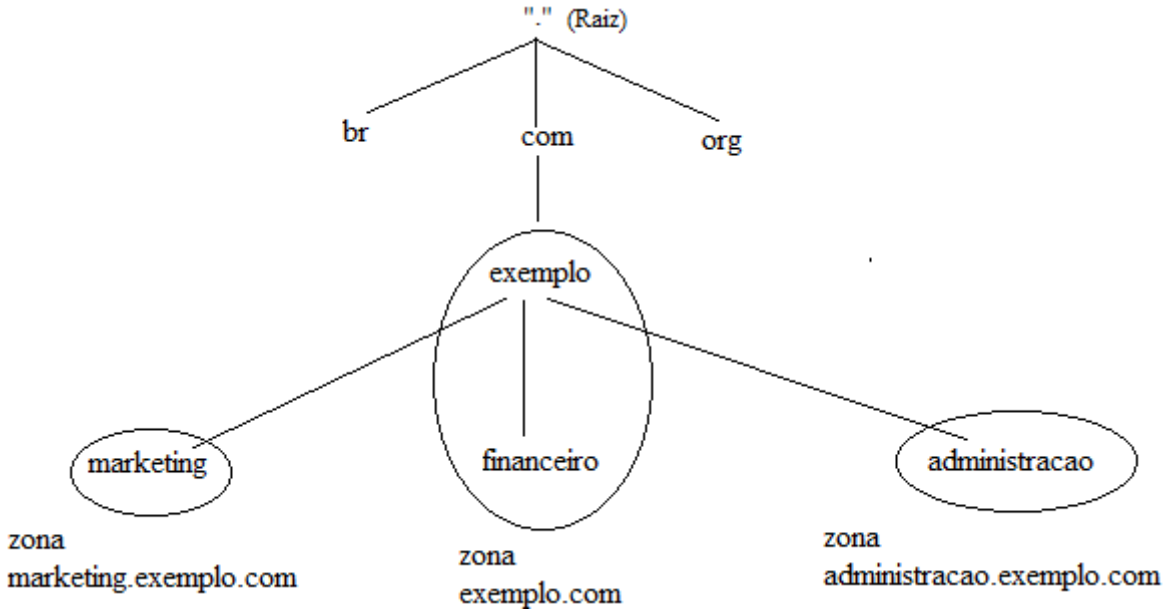
O espaço de nomes do DNS é hierárquico, com estrutura em formato de árvore invertida. A raiz da árvore é o domínio raiz; seus filhos são domínios de primeiro nível (TLD ou *Top-Level Domain*), e cada um pode conter vários níveis de subdomínios.[1] Um domínio é uma sub-árvore ou pedaço distinto do espaço de nomes, sendo gerenciado por uma entidade administrativa. O domínio representa toda a estrutura hierárquica de um nó na árvore de domínios, com o próprio nó incluso. Os dados referentes a um domínio são armazenados em um servidor com autoridade sobre esse domínio, também chamado de servidor autoritativo. Através dessa estrutura, se um servidor possuir a necessidade de pesquisar dados de outros, eles compartilham e cooperam entre si para obter esses dados (ALBITZ; LIU, 2006; MEDEIROS, 2011; SCHUBA, 1993; CHANDRAMOULI; ROSE, 2013).

O conceito de zona é semelhante ao de domínio. Enquanto um domínio é uma subárvore do espaço de nomes, uma zona é a porção do espaço de nomes administrada por um servidor de nomes; a distinção é que uma zona contém todos os nomes de subdomínios e dados de um domínio, à exceção dos subdomínios delegados para outra autoridade (os quais constituem assim uma outra zona administrativa). A zona e o domínio são idênticos quando o servidor de nomes controla todos os nomes de um domínio, ou seja, quando não ocorrer delegação de

[1] No total, um nome de domínio pode ter até 128 componentes, cada um com até 63 caracteres, sendo que o nome completo é limitado a 255 caracteres (MOCKAPETRIS, 1987a; CONRAD, 2012).

autoridade. O mais comum, no entanto, é que algumas porções de um domínio sejam delegadas para outros servidores (ALBITZ; LIU, 2006; MEDEIROS, 2011; SCHUBA, 1993; CHANDRAMOULI; ROSE, 2013). A Figura 1 exemplifica essa divisão de domínios e zonas administrativas.

Figura 1 - Exemplo de domínios e zonas administrativas. Sub-árvores representam domínios, e elipses representam zonas.



Fonte: Imagem elaborada pelo autor.

No exemplo da Figura 1, o domínio `exemplo.com` possui três subdomínios: `marketing.exemplo.com`, `administracao.exemplo.com` e `financeiro.exemplo.com`. Os subdomínios `marketing` e `administracao` são administrados por servidores de nomes distintos do servidor de `exemplo.com`, dessa maneira eles representam zonas diferentes. Já o subdomínio `financeiro` não é administrado de maneira independente, sendo responsabilidade do próprio servidor de nomes do domínio `exemplo.com`. Dessa maneira, este subdomínio está na zona `exemplo.com`.

Cada entrada da árvore do DNS contém um conjunto de informações de recursos, que estão associadas ao nome de domínio que identifica o nó. Esse conjunto é formado por uma coleção de registros de recurso (RR, *resource records*). Os RRs são divididos em classes, nas quais cada uma pertence a uma rede ou

software diferente. A classe mais utilizada é a IN, que representa a Internet (ALBITZ; LIU, 2006; MOCKAPETRIS, 1987a). Outras classes incluem a CH (antiga rede Chaosnet) (MOON, 1981) e HS (*software* Hesiod) (DYER, 1988). No restante deste texto, assume-se a classe IN, salvo indicação em contrário.

Dentro de uma classe, cada RR possui um determinado tipo. Os tipos representam a variedade de informações que podem ser armazenadas para um dado domínio. Cada classe suporta diferentes tipos de RR, sendo que alguns tipos podem ser comuns a mais de uma classe (ALBITZ; LIU, 2006). Os tipos mais comuns estão resumidos na Tabela 1 (MAZIERO, 2010**;** MOCKAPETRIS, 1987b; SCHUBA, 1993).

Tabela 1 - RRs mais comuns no DNS.

Tipo	Descrição
A	Representa um endereço IPv4 para um domínio
AAAA	Representa um endereço IPv6 para um domínio
CNAME	Representa o nome canônico (ou primário) para um apelido/sinônimo (*alias*) de um domínio/servidor.
TXT	Representa um texto de strings
NS	Representa um servidor de nomes autoritativo para o domínio
MX	Representa um servidor de emails para o domínio.
PTR	Representa um ponteiro para um outro nome de um domínio no espaço de domínios.
SOA	Representa o início de uma zona administrativa. Especifica o servidor (a autoridade) que será a fonte de informação autoritativa para o domínio.

Fonte: Tabela elaborada pelo autor.

Um RR padrão é composto pelos seguintes campos (MOCKAPETRIS, 1987a; CHANDRAMOULI; ROSE, 2013):

- *Owner*: É o nome do domínio ao qual o RR está associado.
- Tipo: Valor numérico que identifica qual é o tipo de recurso presente no RR.
- Classe: Valor numérico que identifica a classe.

- TTL (*Time to Live*): É o tempo de vida/validade do RR representado em segundos. É usado principalmente para determinar quanto tempo um RR pode permanecer em *cache* nos servidores de resolução de nomes. Esse valor é determinado pelo administrador da zona na qual se originou esse RR.
- RDATA: É a informação que descreve o recurso, sendo dependente do tipo e, às vezes, da classe ao qual o RR pertence.

2.3 RESOLUÇÃO DE NOMES

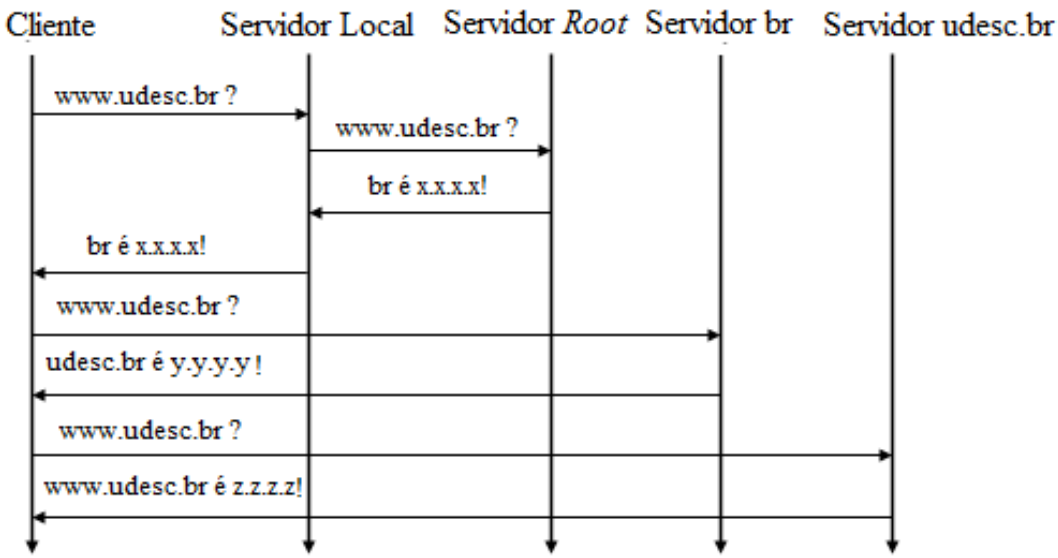
O processo de recuperação de dados armazenados no DNS é chamado de resolução de nomes. Para isso, aplicações recorrem a um cliente chamado de resolvedor ou *resolver*, que tem a responsabilidade de (ALBITZ; LIU, 2006; PESCATORE, 2014; CHANDRAMOULI; ROSE, 2013; SCHUBA, 1993):

- Enviar consultas para os servidores DNS;
- Interpretar as respostas (*Resource Records* ou erro); e
- Retornar a informação para os programas que a requisitaram.

Por conta da divisão da administração do espaço de nomes em zonas sob responsabilidade de servidores distintos, pode ser necessário contactar diversos servidores para resolver um nome. Voltando ao exemplo da Figura 1, o servidor de `exemplo.com` não é responsável pelo nome `abc.marketing.exemplo.com`, pois o subdomínio `marketing` está em uma zona separada; caso receba uma consulta por esse nome, o servidor de `exemplo.com` retorna apenas os dados do servidor responsável por `marketing.exemplo.com`, que é quem tem autoridade sobre o nome. Se a mesma consulta for enviada ao servidor do domínio `com`, este retornará os dados do servidor de `exemplo.com`, que é o próximo na cadeia de delegação (`exemplo.com` está em uma zona separada). Assim, existem dois tipos de consultas DNS que um servidor pode receber: iterativas e recursivas. No modo de consulta iterativa (também chamada de não recursiva), o servidor DNS pode devolver ao cliente uma resposta parcial, indicando um servidor de nomes mais próximo do nome desejado na hierarquia (MOCKAPETRIS, 1987a), permitindo que o cliente dê sequência ao processo de resolução. Ou seja, o servidor requerido retorna simplesmente a melhor resposta que é capaz de fornecer consultando seu

repositório de dados local; se a resposta final não for encontrada, então é realizado o melhor esforço para retornar uma resposta parcial que auxilie na continuação do processo de resolução (ALBITZ; LIU, 2006; SCHUBA, 1993). Esse é o tipo de consulta realizada com servidores DNS autoritativos. A Figura 2 mostra um exemplo de consulta iterativa.

Figura 2 – Exemplo de consulta DNS iterativa

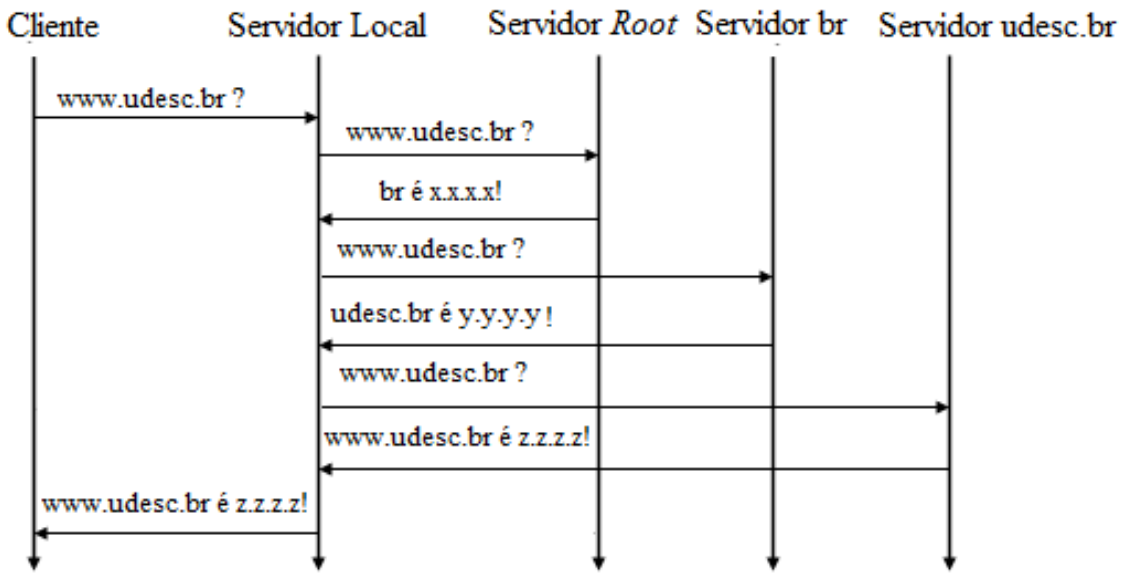


Fonte: Imagem adaptada de Maziero (2010)

No exemplo da Figura 2, um cliente realiza consultas iterativas para descobrir o endereço IP do domínio `www.udesc.br`. Primeiramente o mesmo envia a consulta DNS ao servidor local; supondo que o *cache* desde servidor esteja vazio, ele consulta o servidor autoritativo *root*/raiz pelo domínio `www.udesc.br`. O servidor raiz desconhece o servidor autoritativo do domínio `www.udesc.br`, mas devolve uma resposta parcial informando o endereço do servidor autoritativo do domínio `br` ao servidor local, e essa resposta é repassada ao cliente. O cliente então realiza o mesmo passo anterior para cada resposta parcial recebida (que aponta para um servidor autoritativo), de maneira iterativa, até descobrir o endereço do domínio `www.udesc.br`. Ou seja, o cliente consulta o servidor autoritativo `br` e recebe o endereço do servidor autoritativo `udesc.br`, e, por último, consulta o servidor autoritativo `udesc.br`. Como o domínio `www.udesc.br` está na zona administrativa do servidor `udesc.br`, este devolve ao cliente o endereço IP correto.

No modo de consulta recursiva, um servidor DNS – chamado de servidor recursivo – encaminha a consulta do cliente requisitante a todos os servidores DNS necessários até que a resposta seja encontrada, devolvendo assim ao cliente somente a resposta final (MAZIERO, 2010). Esse tipo de consulta coloca a maior parte da carga de trabalho no servidor recursivo, que é obrigado a retornar uma resposta válida (ou um erro) quando recebe esse tipo de consulta. O servidor repete o mesmo processo básico: envia uma consulta iterativa a um servidor autoritativo e, ao receber de volta referências a outros servidores, escolhe um deles, realizando o mesmo processo sucessivamente até encontrar a resposta final esperada ou obter um erro (por exemplo, o domínio buscado não existe no espaço de nomes). O resultado obtido é então repassado para o cliente requisitante (ALBITZ; LIU, 2006; SCHUBA, 1993). A Figura 3 mostra um exemplo de consulta recursiva.

Figura 3 – Exemplo de consulta DNS recursiva.



Fonte: Imagem adaptada de Maziero (2010)

No exemplo da Figura 3, um cliente envia uma consulta recursiva ao servidor recursivo local para descobrir o endereço IP do domínio `www.udesc.br`. Como o servidor local recebeu uma consulta do tipo recursiva, ele passa a consultar servidores autoritativos de maneira iterativa até obter a resposta desejada pelo cliente. Neste exemplo, o cliente realiza uma única consulta e recebe a resposta

final, enquanto o servidor local realiza várias consultas iterativas aos servidores autoritativos para obter as respostas parciais e, por último, a resposta final.

Para CERT.br (2013), um problema de configuração bastante comum é permitir que qualquer máquina da Internet possa fazer consultas ao servidor DNS recursivo de uma determinada rede. Servidores com esse problema são chamados servidores DNS recursivos abertos, isso porque apenas o servidor autoritativo deve responder consultas de máquinas externas.

O DNS pode usar como protocolo de transporte para consultas e respostas tanto UDP quanto TCP, sendo o primeiro preferido por razões de desempenho; em ambos os casos, o número de porta padrão usado pelo servidor é 53. Originalmente, mensagens DNS sobre UDP tinham um limite de 512 bytes de tamanho máximo (sem contar os cabeçalhos IP e UDP); mensagens maiores que o limite deveriam usar TCP (MOCKAPETRIS, 1987b). Todavia, com a evolução do DNS, esse limite se mostrou demasiadamente restritivo. Por exemplo, a inclusão de RRs usados pelo DNSSEC (Seção 2.7) frequentemente resulta na necessidade de respostas muito maiores que 512 bytes. Por isso, em 1999 foi introduzida uma extensão, conhecida como EDNS(0) (VIXIE, 1999), que possibilita o uso de mensagens maiores que 512 bytes, desde que tanto o cliente como o servidor implementem essa extensão.

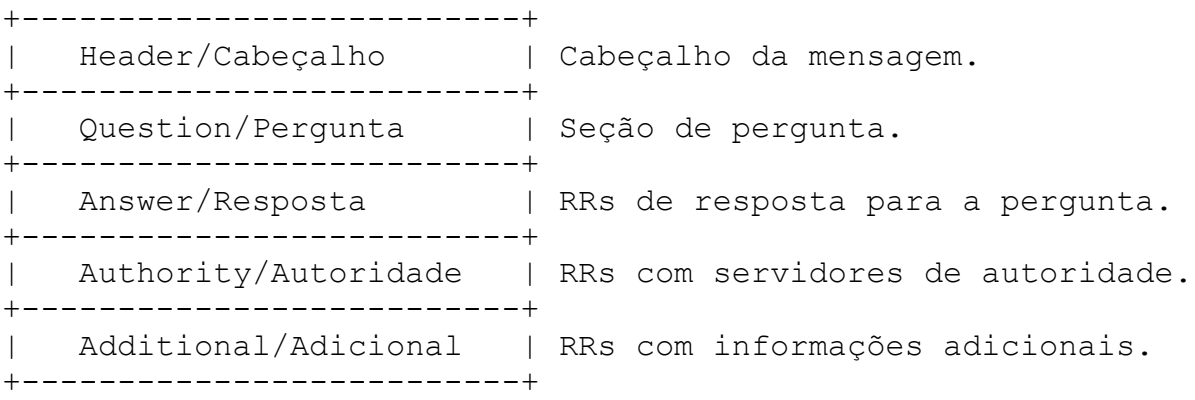
A resolução de nomes, seja iterativa ou recursiva, envolve diversas trocas de mensagens DNS, que precisam seguir uma formatação previamente conhecida por todos os sistemas envolvidos no processo. A estrutura e formato das mensagens DNS são definidos na Seção 2.4.

2.4 FORMATO DAS MENSAGENS DNS

A troca de informação entre clientes e servidores usa mensagens de requisição (*request*) e resposta (*answer*). Ambas possuem a mesma estrutura/formatação, contendo até cinco seções individuais (Figura 4). Dessas cinco seções, duas são encontradas tanto em requisições quanto em respostas, a seção de Cabeçalho (*Header*) e a seção de Pergunta (*Question*). As três últimas seções da mensagem DNS são as seções de Resposta (*Answer*), Autoridade (*Authority*) e Adicional (*Additional*), e são usadas nas mensagens de resposta DNS

(MOCKAPETRIS, 1987b; THE TCP/IP GUIDE, 2005). O conteúdo das diferentes seções é detalhado nas Seções 2.4.1, 2.4.2 e 2.4.3.

Figura 4 – Formato de uma Mensagem DNS de Requisição/Resposta.

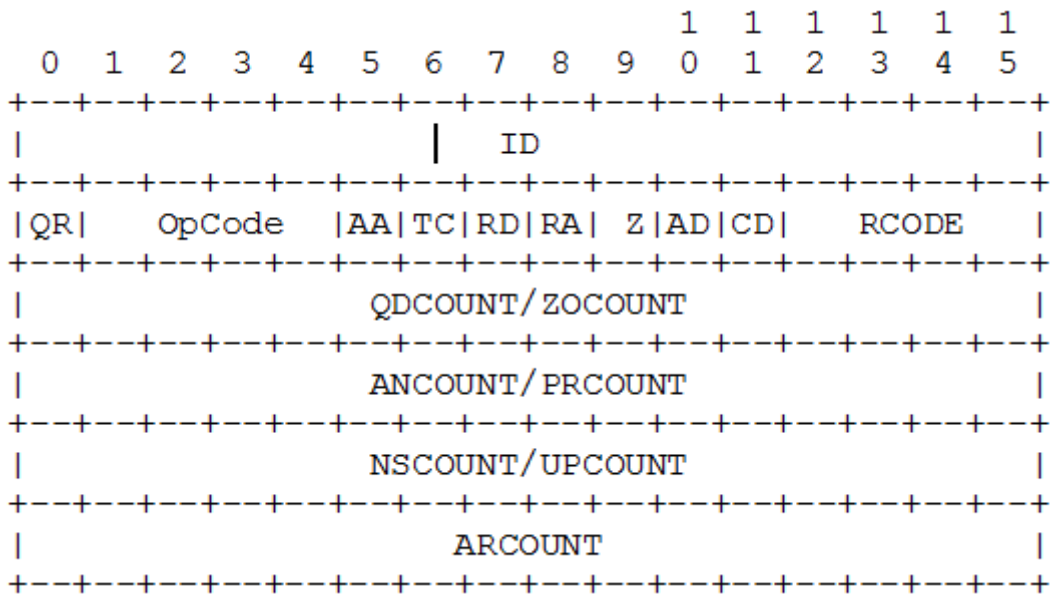


Fonte: Adaptado de Mockapetris (1987b)

2.4.1 Cabeçalho

Toda mensagem DNS inicia por um cabeçalho de 12 bytes, que tem o formato mostrado na Figura 5. Os campos do cabeçalho são os seguintes (IANA , 2015; MOCKAPETRIS, 1987b):

Figura 5 – Formato do Cabeçalho de Mensagem DNS.



Fonte: (EASTLAKE, 2013).

- ID: Um campo de identificação com 16 bits, cujo valor é atribuído pelo programa que gerou a consulta. O servidor copia o valor na mensagem de resposta. O requisitante usa este valor para relacionar as respostas recebidas com as requisições realizadas.
- QR: *Flag* de um bit que especifica se a mensagem é uma requisição/consulta (0), ou uma resposta (1).
- OPCODE: Um campo com 4 bits que determina qual o tipo de consulta presente na mensagem. O valor é atribuído pelo requisitante da consulta e copiado na resposta. Os possíveis valores e seus significados são (EASTLAKE, 2013; MOCKAPETRIS, 1987b):
    - 0: uma consulta padrão (*QUERY*).
    - 1: uma consulta inversa (*IQUERY*) – não implementada por muitos servidores, e finalmente relegada a obsolescência em 2002 (RAISANEN *et. al.*, 2002).
    - 2: uma consulta de *status* do servidor (*STATUS*) – a RFC 1035 definiu esse valor e deixou para especificar sua semântica no futuro, mas isso nunca foi padronizado.
    - 3: valor sem atribuição reservado para uso futuro.
    - 4: mensagem de notificação (*NOTIFY*).
    - 5: mensagem de atualização (*UPDATE*).
    - 6-15: Valores sem atribuição reservados para uso futuro.
- AA: Resposta Autoritativa (*Authoritative Answer*). *Flag* de um bit válida somente em respostas. Determina se o servidor que originou a resposta é autoritativo para o domínio especificado na seção de pergunta da mensagem.
- TC: Truncamento (*Truncation*). *Flag* de um bit que determina se a mensagem foi truncada devido ao tamanho ser superior ao permitido pelo canal de comunicação.
- RD: Recursão Desejada (*Recursion Desired*). *Flag* de um bit que, caso ligada, solicita que o servidor de nomes realize a consulta recursivamente. Valor atribuído pelo requisitante e copiado na resposta.

- RA: Recursão Disponível (*Recursion Available*). *Flag* de um bit que determina se consultas recursivas são oferecidas pelo servidor de nomes. Valor atribuído pelo servidor na resposta.
- Z: Campo reservado para uso futuro. Deve ter valor zero em todas as mensagens DNS.
- AD: Dados Autênticos (*Authentic Data*). *Flag* relevante para respostas de servidores de nomes que implementam recursos de segurança/autenticação (*Security-Aware Recursive Name Server*) (DNSSEC). Se ligada indica que o servidor se responsabiliza e considera todos os RRs nas seções de Resposta e de Autoridade como autênticos. Requisições com essa *flag* ligada indicam que o requisitante entende o seu significado e está interessado no valor (ARENDS et al, 2005c; BLACKA; WEILER, 2013; EASTLAKE, 1998; INACON; 2010).
- CD: Checagem Desabilitada (*Checking Disable*). *Flag* relevante para requisições para servidores de nomes que implementam recursos de segurança e autenticação (*Security-Aware Recursive Name Server*) (DNSSEC). Se ligada indica para o servidor que recebeu a requisição que não é necessário autenticar (verificar assinaturas) na resposta que enviar de volta e nem filtrar RRs que normalmente rejeitaria. Em outras palavras, indica que o requisitante se responsabiliza pela autenticação da resposta, e que dessa maneira o servidor requisitado não deve interferir. A *flag* deve ser copiada na resposta (ARENDS et al, 2005c; BLACKA; WEILER, 2013; EASTLAKE, 1998; INACON; 2010).
- RCODE: campo de 4 bits especificando o código de resposta, que indica se uma resolução foi bem sucedida ou se houve algum erro. A RFC 1035 (MOCKAPETRIS, 1987b) especifica seis valores para RCODE:
    - 0 (*NoError*, sem erro): em uma resposta, indica que a resolução foi bem sucedida. Requisições DNS devem ter RCODE=0.
    - 1 (*FormErr*, erro de formatação): indica que o servidor foi incapaz de interpretar a requisição recebida, que possuía formato incorreto.

- 2 (*ServFail*, falha do servidor): significa que o servidor foi incapaz de processar a requisição recebida, devido a algum erro ou falha interna ocorridos no próprio servidor durante o processamento.
- 3 (*NXDomain*, domínio inexistente): indica que o nome de domínio consultado não existe.
- 4 (*NotImp*, não implementado): significa que o servidor não suporta o tipo de Requisição recebida.
- 5 (*Refused*, recusada): indica que o servidor se negou a realizar a operação especificada porque esta violava alguma política.

Além desses valores, códigos RCODE de 6 a 10 são usados com atualizações dinâmicas no DNS (*DNS UPDATE)* (VIXIE *et. al.*, 1997), e códigos de 16 a 23 são usados para processamento de mensagens DNS do tipo TKEY, TSIG e EDNS (DAMAS *et al*, 2013; EASTLAKE, 2000; THE TCP/IP GUIDE, 2005; VIXIE *et al*, 2000). Como esses códigos não fazem parte do escopo deste trabalho eles não são usados diretamente pela ferramenta de *honeypot* DNS (a ferramenta é capaz de registrá-los para análise mas não os gera ou interpreta), e por brevidade não são detalhados aqui.

- QDCOUNT: Inteiro de 16 bits especificando o número de entradas na seção de pergunta.
- ANCOUNT: Inteiro de 16 bits especificando o número de RRs na seção de resposta.
- NSCOUNT: Inteiro de 16 bits especificando o número de RRs na seção de autoridade.
- ARCOUNT: Inteiro de 16 bits especificando o número de RRs na seção adicional.

2.4.2 Seção de Pergunta

A seção de pergunta contém campos que descrevem a consulta realizada ao servidor DNS, ou seja, define o que o usuário requisitante quer saber do servidor (MOCKAPETRIS, 1987b). Cada requisição carrega apenas um nome, embora a

especificação original permita especificar múltiplas consultas em uma mesma requisição (QDCOUNT > 1), isso não é usado na prática, não sendo sequer suportado pelos servidores DNS mais populares. Os campos da pergunta são:

- QNAME: Campo de tamanho variável indicando o nome do domínio consultado (a codificação é irrelevante para este documento, e não será detalhada).
- QTYPE: Campo de 16 bits que especifica o tipo consultado, como A, SOA ou NS.
- QCLASS: Campo de 16 bits que especifica a classe consultada (tipicamente IN).

2.4.3 Seções de Resposta, Autoridade e Adicional

As seções de resposta, autoridade e adicional possuem o mesmo formato: uma lista concatenada contendo zero ou mais Registros de Recursos (MOCKAPETRIS, 1987b). A seção de resposta contém RRs que respondem à requisição feita pelo cliente na seção de pergunta da mensagem de requisição. A seção de autoridade contém RRs que apontam para os servidores de nomes autoritativos para o domínio consultado. A seção adicional contem RRs que estão relacionados à requisição recebida, mas que não são respostas diretas à ela, como os endereços IP (registros A) correspondentes aos servidores de nomes contidos na seção de autoridade (registros NS). O preenchimento da seção adicional é facultativo, e um servidor de nomes só deve fazê-lo se já tiver os RRs em *cache* — o objetivo é evitar que o resolvedor tenha de fazer uma nova requisição que pode ser antecipada com alta probabilidade, acelerando assim o processo de resolução de nomes e reduzindo a carga sobre os servidores DNS. O número de RRs que cada seção possui é definido nos respectivos campos contadores presentes no cabeçalho.

Cada RR possui os seguintes campos (MOCKAPETRIS, 1987a, 1987b; EASTLAKE, 2013; IANA, 2015):

- NAME: Com tamanho variável, é o nome do domínio ao qual o RR está associado.

- TYPE: Campo de 16 bits que especifica o tipo do RR. Determina o tipo de dado encontrado no RDATA (valor do RR). É um subconjunto dos valores possíveis do campo QTYPE.
- CLASS: Campo de 16 bits que especifica a classe do RR. É um subconjunto dos valores possíveis do campo QCLASS.
- TTL (*Time to Live*): Campo de 32 bits que determina o tempo (em segundos) que um RR pode ser armazenado em *cache* até que se deva descartá-lo. O valor zero determina que o RR só deve ser usado para a transação corrente, e assim não deve ser gravado em *cache*.
- RDLENGTH: Valor inteiro de 16 bits que especifica o tamanho em octetos do campo RDATA.
- RDATA: Campo de tamanho variável, contendo o valor do RR. O formato desse campo varia dependendo do tipo e da classe.

A seção adicional pode ainda conter também um pseudo-RR (*TYPE* 41, chamado também de meta-RR) tanto em requisições quanto em respostas. A presença desse RR em requisições é usada para sinalizar que se trata de uma consulta utilizando EDNS(0) (DAMAS *et al*, 2013).

O conhecimento do formato e utilização das mensagens DNS permite ter um entendimento melhor das vulnerabilidades e ameaças ao sistema. Esses aspectos de segurança são apresentados nas Seções 2.5 e 2.6, respectivamente.

2.5 ASPECTOS DE SEGURANÇA

O DNS usa na maior parte do tempo o protocolo UDP, por questões de desempenho. Problemas de segurança se originam dessa situação, pois o protocolo UDP não oferece nenhum meio de validar as informações trafegadas, e esse problema é repassado ao DNS. No DNS a autenticidade de uma entidade é verificada através de nomes e endereços, dados estes que um atacante pode forjar. Além disso, a ausência de criptografia na comunicação entre entidades facilita a injeção de informações em uma mensagem DNS. Diante desses problemas relacionados a autenticidade, integridade e confidencialidade, Gallois (2010) sumariza as principais vulnerabilidades estruturais do DNS em quatro:

- Inexistência de garantia de autenticidade de origem. Não é possível garantir a origem das mensagens DNS obtidas, uma vez que essa informação é extraída do endereço de origem no cabeçalho IP, que pode ser forjado (GONT, 2008).
- Inexistência de garantia de integridade dos dados. Não é possível verificar se uma mensagem DNS foi alterada ou forjada, posto que não são usados mecanismos criptográficos para proteger a comunicação.
- Ausência de segurança do canal de comunicação. Não é possível impedir que uma mensagem DNS seja capturada, observada ou alterada em trânsito enquanto trafega na Internet. Novamente, a ausência de criptografia permite essa vulnerabilidade.
- Impossibilidade de efetuar uma negação de origem autenticada, isto é, não há maneira segura de verificar a inexistência de um nome. Um atacante pode fornecer respostas fraudulentas com RCODE=3 (*NXDomain*) para consultas por nomes existentes, provocando indisponibilidade ao impedir a comunicação entre os *hosts* afetados (os clientes que recebem as respostas falsas e os *hosts* cujos nomes são indevidamente caracterizados como inexistentes).

Cabe ressaltar que essas vulnerabilidades estão presentes no DNS desde sua concepção e especificação nas RFC 1034 e 1035, de 1987 (MOCKAPETRIS, 1987a; 1987b). Usuários maliciosos podem explorar essas vulnerabilidades estruturais, resultando em algumas ameaças conhecidas do DNS, definidas na Seção 2.6.

2.6 AMEAÇAS AO SISTEMA DNS

Usuários maliciosos podem explorar as vulnerabilidades estruturais do sistema DNS para atacá-lo ou usá-lo como fonte para outros ataques (CONRAD, 2012). Tais ataques, que abrangem corrupção de dados, exposição de informação, negação de serviço, amplificação, canais cobertos e corrupção de aplicação, são descritos nesta Seção.

2.6.1 Ameaças de Corrupção de Dados

A corrupção de dados ocorre quando alguém intencionalmente ou acidentalmente altera dados do DNS de maneira não autorizada, seja na fonte (servidores) ou em trânsito, violando assim a integridade dos dados. Por exemplo, um servidor DNS recursivo pode ter respostas incorretas inseridas em seu *cache* (*cache poisoning*), resultando em seu repasse em consultas futuras (CONRAD, 2012; SANTCROOS; KOLKMAN, 2007). Ameaças de corrupção de dados são facilitadas pela inexistência de garantia de autenticidade de origem, integridade dos dados e segurança no canal de comunicação.

Um outro tipo de ataque de corrupção de dados é o *pharming*, que envolve o redirecionamento transparente e oculto de tráfego para endereços falsos, por meio da alteração de RRs de um domínio alvo para apontar para endereços IP controlados pelo atacante. Dessa maneira, usuários que navegam manualmente para um *site web* acreditarão que estão interagindo com serviços legítimos, enquanto o atacante usurpa o tráfego do domínio alvo. É mais difícil de detectar se comparados a ordinários *sites* fraudulentos. Não é preciso atrair a vítima para o *site* fraudulento, pois a mesma vai acessá-lo por conta própria (CERT.br, 2012; JAKOBSSON et al, 2006).

Essa redireção pode ser feita basicamente de três formas (CERT.br, 2012):

- Pelo comprometimento do servidor DNS de um provedor;
- Pela ação de códigos maliciosos que alteram o comportamento normal do serviço DNS no computador de uma vítima;
- Pela ação direta de um invasor que consiga acesso às configurações de DNS do computador ou equipamento de rede da vítima.

Um exemplo desse ataque é o chamado *drive-by pharming*, onde um atacante corrompe um dispositivo de rede nas instalações de um usuário, tal como um roteador, de modo que o mesmo passe a prover o endereço IP da máquina do atacante ou de um servidor comprometido pelo mesmo, no lugar do IP do servidor DNS legítimo da vítima. Podendo assim o atacante controlar as respostas às requisições DNS da vítima, redirecionando-as para endereços fraudulentos sem que a mesma possa perceber, pois o resolvedor acredita se comunicar com um serviço DNS legítimo. As configurações de um roteador podem ser alteradas, sendo o

endereço do servidor DNS uma delas (CONRAD, 2012; JAKOBSSON et al, 2006). Jakobsson et al (2006) demonstram como um *site web* malicioso pode ser usado para atacar roteadores domésticos, de modo a montar ataques sofisticados de *pharming*, que podem resultar em negação de serviço, infecção por *malware* e roubo de identidade.

Devido à falta de confidencialidade no canal de comunicação, um usuário malicioso pode interceptar mensagens DNS em trânsito, descobrir ou deduzir o identificador (ID) da transação e forjar uma resposta fraudulenta ao cliente com um mapeamento errôneo entre um domínio e um endereço IP, resultando em um redirecionamento do cliente para um servidor fraudulento ou no comprometimento do *cache* de servidores DNS com a injeção de dados falsos. O comprometimento e controle de servidores de nomes e a alteração de seus RRs de maneira não autorizada pelos atacantes é outra abordagem de ataque de corrupção da informação.

2.6.2 Ameaças de Exposição de Informação

Ameaças de exposição de informação são facilitadas pela inexistência de mecanismos de cifragem no canal de comunicação. Nesse tipo de ameaça, os dados fornecidos pelo DNS são visualizados e expostos pelos atacantes, podendo ocorrer a divulgação de informações dos usuários (como sites visitados). De acordo com Arends *et al*. (2005a), o sistema DNS foi originalmente concebido com o pressuposto de que confidencialidade e privacidade não são requisitos, e as informações DNS são consequentemente visíveis. Devido a isso, tradicionalmente não se considera que a observação e inspeção de tráfego DNS afetem o funcionamento correto do sistema. Porém, para Conrad (2012), a exposição da informação pode ser danosa para alguns indivíduos e organizações, dessa maneira podendo afetar a confiança que indivíduos têm no sistema DNS.

2.6.3 Ameaças de Negação de Serviço contra Servidores DNS

Ataques de negação de serviço (*denial of service,* DoS) contra servidores DNS impedem que os usuários usem os serviços de DNS. Esses ataques, que não

são específicos do DNS, são facilitados pela inexistência de segurança no canal de comunicação e são o tipo de ameaça mais significativa para o DNS, e o mais dificil de se defender (CONRAD, 2012; SANTCROOS; KOLKMAN, 2007).

O ataque pode ocorrer pela exaustão ou pela interrupção da disponibilidade de recursos. Na exaustão, uma quantidade insuficiente de recursos fica disponível, resultando em lentidão ou falhas no oferecimento de serviços. Por exemplo, um atacante pode realizar uma grande quantidade de requisições a um servidor DNS, sobrecarregando-o e dificultando ou impedindo que consultas legítimas sejam processadas. Já na interrupção, um evento torna os recursos totalmente indisponíveis até que um evento externo os restaure. Por exemplo, uma falha no suprimento de energia resultaria no desligamento dos servidores (CONRAD, 2012).

A ameaça de DoS se aplica a todos os componentes do DNS, incluindo a infraestrutura física das redes e dos prédios com os servidores; a infraestrutura dos servidores que processam requisições; a infraestrutura de gerenciamento dos RRs que contém as informações de DNS; e a infraestrutura administrativa de negócio (CONRAD, 2012). Ou seja, qualquer tipo de comprometimento que cause uma interrupção ou indisponibilidade não autorizada ou prevista dos serviços de DNS pode ser caracterizada como um ataque de DoS, independente se o *software* ou *hardware* foi comprometido.

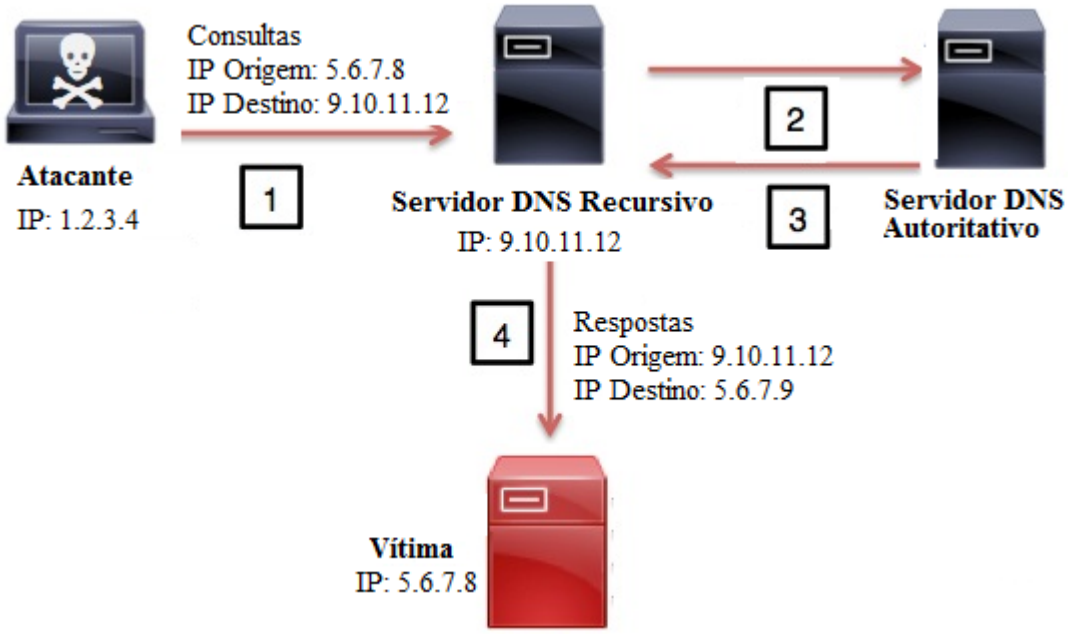
2.6.4 Outros Riscos (Ataques de Amplificação, Canais Cobertos e Ataques de Corrupção de Aplicação)

Devido ao fato do tráfego DNS receber menos filtragem em *firewalls* e outros mecanismos de controle de acesso em redes, uma vez que consultas DNS são tidas como inofensivas e o foco da prevenção está na configuração adequada dos servidores, o DNS pode ser usado como uma fonte ou meio para possíveis ataques a recursos e alvos específicos na Internet (CONRAD, 2012; PESCATORE, 2014). Outros riscos de segurança que são originados pelo DNS, ou seja, exploram vulnerabilidades do DNS para usá-lo como fonte para outros ataques, incluem amplificação, canais cobertos e corrupção de aplicações:

- A amplificação é uma característica típica do DNS, pois respostas geralmente são maiores que consultas (CONRAD, 2012; US-CERT, 2013). Em condições

normais, isso não é um problema (por exemplo, a assimetria de tamanho entre requisições e respostas HTTP é muito maior, sem impacto no funcionamento da Web). O ataque de amplificação se apoia em dois pilares, que são a assimetria e o uso de UDP. A Figura 6 apresenta um exemplo de ataque. No passo 1, o atacante envia um grande número de consultas DNS com um endereço IP de origem forjado (IP da vítima), com o intuito de saturar essa vítima com tráfego DNS (CONRAD, 2012; ARENDS et al., 2005a; SANTCROOS; KOLKMAN, 2007). O servidor contactado (9.10.11.12, no exemplo), chamado de refletor, é tipicamente um servidor recursivo aberto, pertencente a uma rede não relacionada ao atacante ou à vítima; os passos 2 e 3 da figura mostram esse servidor interagindo com um servidor autoritativo para resolver os nomes consultados. Supondo que o fator de amplificação seja 10, ou seja, que as respostas sejam 10 vezes maiores que as consultas, 50 MB de tráfego de consulta gerariam 500 MB de tráfego de resposta, que vai ser enviado para a vítima, conforme mostrado no passo 4.

Figura 6 – Exemplo de um ataque de amplificação usando servidor DNS recursivo.



Fonte: Imagem adaptada de Cisco (2013).

- Canais cobertos (*Covert Channels*) de comunicação. Neste tipo de ameaça, o atacante esconde dados dentro de mensagens de outros protocolos (no caso,

mensagens DNS), para se comunicar remotamente sem ser detectado por sistemas de segurança. Requisições e respostas DNS são usadas para comunicação dissimulada, o que pode resultar no atacante desviar de mecanismos de segurança e comprometer sistemas internos (CONRAD, 2012). Um exemplo deste tipo de ataque é citado por Dietrich et al. (2011), que estudaram um *botnet* que usava tráfego DNS como canal para Comando e Controle (*Command and Control* – C&C). Uma *botnet* é um conjunto de computadores (chamados *bots*) infectados por um *software* malicioso que permite que sejam controlados remotamente. O controlador de uma *botnet*, ou *bot master*, se comunica com os *bots* por intermédio de um canal de C&C, pelo qual o *master* envia comandos e recebe dados dos *bots* (COOKE *et al*, 2005; HOLZ, 2005).

- Ataques de corrupção de aplicação. Em certos casos, respostas DNS podem resultar em comportamento não previsto das aplicações. Devido a vulnerabilidades nas bibliotecas usadas para resolução de nomes, as respostas podem ser forjadas maliciosamente para comprometer sistemas, explorando essas vulnerabilidades. Filtrar os dados das respostas antes de passar para a aplicação pode prevenir o problema (CERT–SEI, 2002a; 2002b; 2003; CONRAD, 2012; CVE, 2002).

  Por exemplo, houve no passado casos de servidores DNS com vulnerabilidades de transbordamento de *buffer* (*buffer overflow*) em código de biblioteca que realizava resolução de nomes (CERT–SEI, 2002a ;2002b; 2003; CVE, 2002). Em um dos casos, a função que lidava com respostas e requisições, quando lia porções das mensagens de resposta DNS, copiava dados recebidos da rede em *buffers* de tamanho inadequado. Dessa maneira, uma resposta DNS especialmente montada poderia gerar um transbordamento de *buffer*, consequentemente injetando código arbitrário na pilha de execução. A exploração dessa vulnerabilidade poderia causar o encerramento da aplicação que solicitou a resolução do nome por acesso ilegal a memória ou ainda a execução de código arbitrário na máquina alvo, com os privilégios da aplicação que fez a chamada à rotina vulnerável. Alguns dos servidores citados com esses problemas incluíam o ISC BIND 4.9.2 até a

versão 4.9.10 (a versão mais atual é a 9.10.3) e outras bibliotecas derivadas do BIND 4 como BSD libc, GNU glibc e System V UNIX.

Muitas das ameaças citadas estão relacionadas às deficiências de autenticidade e integridade das vulnerabilidades estruturais do DNS. As extensões de segurança do DNS (DNSSEC), descritas na Seção 2.7, visam prevenir ataques relacionados a essas deficiências.

2.7 DNSSEC

O DNSSEC (ARENDS *et al*, 2005a; 2005b; 2005c; 2008; BLACKA; WEILER, 2013) é um conjunto de extensões ao DNS que fornece meios de verificar a autenticação de origem e a integridade dos dados DNS. Autenticação significa que administradores de zonas podem prover informações de autoridade para um domínio, enquanto a verificação de integridade garante que a informação não foi modificada em trânsito ou no armazenamento. Assinaturas digitais são usadas para garantir que os dados DNS recebidos em resposta a uma consulta sejam legítimos (CONRAD, 2012; GALLOIS, 2010; SANTCROOS; KOLKMAN, 2007).

O uso do DNSSEC resolve ameaças relacionadas a vulnerabilidades de autenticidade de origem, integridade dos dados e negação autenticada de existência, tais como corrupção de dados e *cache poisoning*. No entanto, não resolve nenhum problema relacionado ao canal de comunicação/transporte, como exposição de informação (CONRAD, 2012; GALLOIS, 2010; SANTCROOS; KOLKMAN, 2007). Além disso, o DNSSEC aumenta o risco de ataques de amplificação e DoS, pois contém respostas ainda maiores que o DNS e exige processamento extra de autenticação criptográfica e validação, consumindo mais recursos dos servidores (CONRAD, 2012; GALLOIS, 2010; ARENDS, 2005a; SANTCROOS; KOLKMAN, 2007; US-CERT, 2013).

2.8 CONSIDERAÇÕES PARCIAIS

Este Capítulo apresenta uma explicação dos principais aspectos do *Domain Name System*, descrevendo sua função, organização, funcionamento, formato das mensagens, aspectos relacionados à segurança e ameaças ao sistema. O DNS apresenta certas vulnerabilidades estruturais advindas da falta de confidencialidade, integridade e autenticidade das mensagens trocadas. Isso permite que usuários maliciosos explorem tais vulnerabilidades para comprometer o sistema e/ou usá-lo para atacar terceiros. A extensão DNSSEC oferece uma solução para ameaças que exploram a falta de integridade e autenticidade, com o custo de processamento e tamanho extra das mensagens, mas não resolve ameaças relacionadas à falta de confidencialidade dos canais de comunicação.

Neste trabalho são consideradas todas as vulnerabilidades citadas na Seção 2.5, ou seja, qualquer tipo de interação suspeita e mal intencionada que venha a explorar essas vulnerabilidades para um ataque ao DNSpot é observada e suas informações colhidas para análise posterior. O DNSpot estará susceptível a receber a maioria dos ataques citados na Seção 2.6, sendo assim necessário realizar o monitoramento das interações recebidas pelo mesmo para detectar os tipos de ataques ou tentativas de ataques sofridos, para análise. Mensagens DNS que usam a extensão de segurança DNSSEC ou os mecanismos de extensão EDNS(0) também são capturadas pelo DNSpot e analisadas posteriormente.

Uma das maneiras de se defender das ameaças é conhecendo melhor a motivação e *modus operandi* dos atacantes. O Capítulo 3 discute o conceito de *honeypot* e como ele é usado para este fim.

3 *HONEYPOTS*

Neste capítulo é realizada uma revisão sobre *honeypots* e *honeynets*. São apresentados os principais conceitos e classificações, discutidos benefícios e limitações, e aplicações dessas ferramentas na segurança de redes. Além disso são definidos os requisitos para uma ferramenta *honeypot* para DNS, e é realizada uma revisão, à luz desses requisitos, das principais ferramentas existentes que implementam o conceito de *honeypot*.

3.1 DEFINIÇÃO

Um dos métodos para observar o comportamento malicioso de potenciais atacantes a sistemas e redes consiste no uso de *honeypots* e *honeynets*. CERT.br (2007) e Steding-Jessen *et al* (2008) definem um *honeypot* como sendo um recurso computacional de segurança cujo objetivo é ser sondado, atacado ou comprometido. Tipicamente, um *honeypot* é um *host* Internet que possui um endereço IP público mas cuja existência não é anunciada, que não hospeda serviços oficiais, e que não origina tráfego para redes externas. Portanto, com base nessas características, um *honeypot* só pode ser encontrado por um usuário externo mediante uma varredura da rede ou adivinhação de seu endereço IP, o que faz com que toda interação com o *honeypot* seja considerada suspeita e potencialmente maliciosa. Para atingir seus propósitos, um *honeypot* precisa ser amplamente monitorado; esse monitoramento deve ser feito de forma bastante discreta, para que os atacantes não suspeitem que estão sendo vigiados e não deixem de interagir com o *honeypot*, indo em busca de outros alvos.

De acordo com Spitzner (2003), os principais benefícios de se usar *honeypots* são:

- Conjunto reduzido de dados: *honeypots* coletam dados somente quando recebem interação. Isso facilita separar dados legítimos da organização de dados maliciosos pois, por definição, dados legítimos não devem chegar ao *honeypot*. Logo, os dados são considerados maliciosos ou, no mínimo, suspeitos. Isso torna os dados dos *honeypots* valiosos, fáceis de gerenciar e simples de analisar se comparados a dados obtidos de um

servidor de produção, cuja análise exigiria uma separação entre tráfego legítimo e ilegítimo.

- Número reduzido de falsos positivos e negativos: *honeypots* reduzem drasticamente o número de eventos legítimos classificados como ataques (falsos positivos), pois por definição toda interação com um *honeypot* é maliciosa e não autorizada. Da mesma forma, o número de ataques genuínos que deixam de ser detectados (falsos negativos) também é reduzido na medida em que os *honeypots* são adequadamente monitorados.
- Criptografia: mesmo que as atividades maliciosas trafeguem criptografadas na rede, o *honeypot* alvo dos ataques conseguirá observá-las e registrá-las, uma vez que o tráfego é decifrado no próprio *honeypot*.
- Flexibilidade: *honeypots* são extremamente adaptáveis e usáveis em um variado número de ambientes, propósitos e contextos, permitindo que analistas de segurança encontrem um equilíbrio adequado entre nível de detalhamento das observações, esforço de manutenção e risco oferecido.
- Recursos reduzidos: *honeypots* necessitam de poucos recursos para operar, mesmo em redes grandes.

Spitzner (2003) cita duas principais limitações de *honeypots*:

- Visão limitada: os *honeypots* somente podem observar eventos que os envolvam diretamente, não sendo capazes de monitorar atividades nos outros sistemas da organização.
- Risco de comprometimento: o objetivo básico de um *honeypot* é permitir que usuários maliciosos interajam com sistemas reais ou emulados. Existem, porém, riscos da interação possibilitar que o invasor use o *honeypot* como plataforma para comprometer outros recursos da organização ou atacar redes externas. Tais riscos adicionais podem ser mitigados configurando o *honeypot* adequadamente e limitando a interação de atacantes ao escopo do que se deseja estudar, que no caso deste trabalho seriam as interações relacionadas a DNS (CERT.br, 2007; STEDING-JESSEN *et al*, 2008).

Além de possibilitarem que profissionais de segurança aprendam o *modus operandi* de atacantes, os *honeypots* têm encontrado outros usos, tais como:

- Modelagem estatística da evolução e tendência de ataques usando dados de *honeypots* (KAÂNICHE et al, 2005);
- Detecção, identificação e observação de ameaças internas, como funcionários mal intencionados e máquinas infectadas com *malware* dentro da rede de uma organização (SPITZNER, 2003);
- Automatização da geração de padrões/assinaturas de ataques e interações maliciosas para serem usados em IDSs (KREIBICH; CROWCROFT; 2004);
- Detecção e coleta de dados relacionados ao abuso da infraestrutura da Internet por parte de *spammers* no envio de mensagens *Spam* (CERT.br, 2015b; 2007b);
- Captura de informações sobre fraudes de cartões de crédito automatizadas e roubo de identidade (THE HONEYNET PROJECT AND ALLIANCE, 2003).

Esses exemplos evidenciam que a aplicação prática dos *honeypots* é abrangente e adaptável as necessidades de uma organização. Os *honeypots* possuem algumas classificações e divisões baseados em suas características, essas classificações são apresentadas nas Seções 3.2 e 3.3. A definição de *honeynet*, uma implementação mais complexa e elaborada do conceito de *honeypot*, é feita na Seção 3.4.

3.2 CLASSIFICAÇÃO DE *HONEYPOTS*

Spitzner (2002) explica que um *honeypot* pode ser classificado com base no seu nível de interatividade, que define o quanto de funcionalidades ou atividades um atacante pode ter com o *honeypot*. Quanto mais interação com um atacante o *honeypot* oferecer, maior será a quantidade de informações que podem ser colhidas sobre esse atacante. No entanto, quanto maior a interação, maior o risco de comprometimento do sistema pelo atacante. Spitzner (2002) define três tipos de *honeypots* com base no nível de interação: os de baixa interatividade, os de média interatividade e os de alta interatividade.

Em *honeypots* de baixa interatividade, são instaladas ferramentas para emular sistemas operacionais e serviços com os quais os atacantes podem interagir.

Oferecem baixo risco de comprometimento pois o nível de interação oferecido ao atacante é limitado, sendo indicados para redes em produção com pouco pessoal e *hardware*. *Honeypots* de alta interatividade utilizam sistemas operacionais, aplicações e serviços reais para os atacantes interagirem. Trazem um alto risco de comprometimento e exigem pessoal qualificado e prevenção de comprometimento (CERT.br, 2007; STEDING-JESSEN *et al*, 2008).

Os *honeypots* de média interatividade oferecem aos atacantes maior nível de interação do que os de baixa interatividade, mas menos funcionalidades do que os de alta interatividade. São projetados de modo a esperar certas atividades do atacante e fornecer certas respostas além das quais um *honeypot* de baixa interatividade iria fornecer (SPITZNER, 2002). Ou seja, o nível de interação dos serviços e sistemas emulados é maior do que o oferecido por *honeypots* de baixa interatividade, mas ainda fazem uso de emulação e não usam totalmente sistemas e serviços reais como os de alta interatividade.

3.3 *HONEYPOTS* DE PRODUÇÃO E DE PESQUISA

Spitzner (2002) e Grimes (2005) apresentam outra classificação de *honeypots* de acordo com o seu uso, dividindo-os em *honeypots* de produção e *honeypots* de pesquisa. Um *honeypot* de produção é usado para proteger uma organização, detectando ataques e mitigando riscos dos atacantes. Oferecem poucos riscos à organização, pois oferecem interação limitada aos atacantes. Já um *honeypot* de pesquisa é usado para aprender e colher informações sobre os atacantes e sobre as ameaças a uma organização. Oferecem maior risco de comprometimento, pois oferecem maior nível de interação e sistemas reais. Por vezes, *honeypots* de pesquisa não recebem instalação de *patches* e atualizações de segurança, de modo a deixar propositalmente brechas a serem exploradas pelos atacantes.

3.4 HONEYNETS

Um conceito frequentemente associado a *honeypots* é o de *honeynet*; embora este trabalho não empregue uma *honeynet*, o conceito é aqui introduzido por razões de clareza. Para CERT.br (2007), uma *honeynet* consiste em uma rede

especificamente projetada para ser comprometida por atacantes, sendo tipicamente um segmento de rede isolado que hospeda um conjunto de *honeypots* de alta interatividade. É um sistema de rede complexo real, no sentido de que nada é emulado. As aplicações, sistemas e serviços encontrados na *honeynet* podem ser os mesmos encontrados na organização. Arquivos e informações adicionais podem ser inseridos na *honeynet* para atrair a atenção do atacante e o fazê-lo interagir com os sistemas. Devido ao risco que introduz, uma *honeynet* precisa ter mecanismos de controle para prevenir que ela seja usada como base para ataques a outras redes (SPITZNER, 2003).

Para a tomada de decisão como deve ser a escolha e implementação das funcionalidades de uma ferramenta *honeypot* para DNS , é necessário definir quais são os requisitos que devem ser satisfeitos para atingir os objetivos da proposta deste trabalho. Os requisitos que determinam o escopo da ferramenta são definidos na Seção 3.5.

3.5 REQUISITOS DE UM *HONEYPOT* DNS

Os requisitos de um *honeypot* DNS se enquadram em duas categorias principais, a funcionalidade oferecida a clientes e os recursos destinados a analistas de segurança. Na primeira categoria, o requisito básico é que o *honeypot* atue como um servidor DNS recursivo, efetuando as resoluções de nomes solicitadas pelos clientes. Nesse contexto, dois requisitos adicionais são:

I. Um observador externo não deve ser capaz de distinguir o *honeypot* de um servidor DNS recursivo real. Por exemplo, um atacante pode iniciar a interação com o *honeypot* fazendo uma consulta por um nome de domínio cujo servidor autoritativo esteja sob seu controle; se algum tráfego diferente do esperado chegar nesse servidor autoritativo, o atacante pode suspeitar de uma armadilha, e assim abandonar o *honeypot*. Em particular, o processo de resolução iterativa de um nome deve aparentar ser realizado pelo próprio *honeypot*, e não por outro servidor (com endereço IP diferente), e não devem ser feitas consultas adicionais para tentar levantar mais informações sobre o domínio usado.

II. O *honeypot* não deve poder ser usado como plataforma para atacar outras máquinas na Internet, especialmente ser usado como refletor em ataques de negação de serviço (Seção 2.6.3 e Seção 2.6.4).

Do ponto de vista do analista de segurança, o *honeypot* deve ser capaz de registrar todas as consultas recebidas e as respostas correspondentes, e eventualmente modificá-las. Além disso, alguns aspectos do seu funcionamento, incluindo como processar determinados tipos de consulta (por exemplo, uma tentativa de descobrir o número de versão do servidor DNS), devem ser configuráveis.

Portanto, os requisitos de um *honeypot* DNS podem ser resumidos da seguinte forma:

R1. Efetuar resolução recursiva de nomes;
R2. Ter comportamento idêntico ao de um servidor recursivo legítimo (perante observadores externos);
R3. Possuir mecanismos que evitem o uso do *honeypot* como plataforma de ataques, especialmente como refletor em ataques de DoS;
R4. Registrar todas as consultas e respostas DNS;
R5. Ser capaz de manipular todas as consultas e respostas DNS;
R6. Permitir a configuração de alguns aspectos de seu funcionamento;
R7. Possuir mecanismos que minimizem o risco de comprometimento do próprio *honeypot*.

Os requisitos R1, R4, R5 e R6 podem ser classificados como requisitos funcionais, e os requisitos R2, R3 e R7 como não funcionais. Na Seção 3.6 são discutidas diversas ferramentas empregadas na implementação de *honeypots* e *honeynets*, e são avaliadas suas funcionalidades à luz dos requisitos definidos.

3.6 FERRAMENTAS PARA *HONEYPOTS*

Existem diversas ferramentas que auxiliam na construção e gerenciamento de *honeypots* e *honeynets*. Dentre as ferramentas mais relacionadas ao propósito deste trabalho, podem ser citadas:

- Honeyd: é um *honeypot* de baixa interatividade que permite criar serviços virtuais emulados na rede (como FTP, HTTP e SMTP) para se comportar como diferentes tipos de servidores (HONEYD, 2008). Pode ser usado para criar múltiplos *honeypots* virtuais em uma única máquina ao mesmo tempo, possibilitando assim a simulação de uma rede inteira. Além disso, uma “personalidade” pode ser configurada de modo que simule um sistema operacional específico. O Honeyd permite que um único *host* use até 65536 endereços IP (HONEYD, 2007). Os serviços são emulados pela utilização de um simples arquivo de configuração ou *script* (várias linguagens são suportadas), que determina como a ferramenta deve processar o serviço e interagir como atacante; quanto melhor é o *script* desenvolvido, mais realista será a simulação do serviço para o atacante. Em vez de simular/emular, é também possível utilizar o recurso de *proxy*, para se conectar e repassar o processamento para uma outra máquina que implemente o serviço real e devolva uma resposta realista ao Honeyd. No caso de serviços DNS, o Honeyd não implementa a rotina de resolução de nomes recursivamente, e criar um *script* de simulação que resolvesse cada requisição do atacante de maneira convincente seria impraticável, devido à complexidade de implementar um serviço de servidor DNS recursivo. Dessa maneira, se faria necessário utilizar o recurso de conexão *proxy* para repassar as requisições do atacante para uma máquina que execute um servidor DNS recursivo real (GRIMES, 2005; HONEYD, 2004).
- Honeywall*:* é uma distribuição *live* de Linux que implementa o *gateway* de uma *honeynet* (THE HONEYNET PROJECT, 2015a), que é um ponto de controle por onde passa todo o tráfego que a *honeynet* troca com o mundo externo. Inclui funcionalidades de controle, registro e análise de tráfego de rede de/para os *honeypots* na *honeynet*.
- InetSim*:* é um emulador de serviços Internet usado principalmente para observar e analisar o comportamento de rede de amostras de *malware* (INETSIM, 2014). A emulação de serviço DNS envia respostas configuradas estaticamente.

- Tracker: é um cliente DNS que monitora periodicamente uma lista de nomes suspeitos (THE HONEYNET PROJECT, 2015b), tentando identificar ataques de *Fast Flux* (THE HONEYNET PROJECT, 2008a), em que um dado nome de domínio possui um grande número de registros A, que são adicionados e removidos em rápida sucessão (esse tipo de ataque é característico de *malware*).
- Kippo: é um *honeypot* SSH que monitora ataques de tentativas de descoberta de senhas por força bruta, e permite registrar as interações de um atacante com um interpretador de comandos emulado no falso servidor SSH (GITHUB,2015).
- Sebek: é um módulo de *kernel* para plataformas Windows e Linux, instalado em *honeypots* de alta interatividade para monitorar atividades como os comandos digitados no sistema, mesmo com conexões cifradas (THE HONEYNET PROJECT, 2008b). Os *logs* de atividades podem ser enviados através da rede para um *host* de monitoramento.

Além dessas, outras ferramentas de *honeypot* podem ser encontradas na página do *Honeynet Project* (THE HONEYNET PROJECT, 2015b).

Nenhuma das ferramentas encontradas atende aos requisitos estabelecidos na Seção 3.5. Aquelas que tratam de DNS, como o InetSim, retornam respostas estáticas, carecendo portanto de flexibilidade. A ferramenta Honeyd permite criar *scripts* para emular a interação com um serviço DNS, mas não realiza o processo de resolução de nomes para as requisições recebidas nativamente, necessitando que as requisições recebidas sejam enviadas pela ferramenta para um servidor DNS externo, que faça a resolução e devolva a resposta. Isso atenderia aos requisitos R1 e R2, mas não aos demais. As ferramentas Honeywall, Tracker, Kippo e Sebek possuem propósito específico, que não contribui suficientemente para os requisitos deste trabalho. Diante disso, optou-se por projetar e implementar uma arquitetura que satisfaça as necessidades de um *honeypot* específico para DNS.

3.7 CONSIDERAÇÕES PARCIAIS

Este capítulo apresenta os *honeypots* e como eles podem ser utilizados para monitoramento e coleta de informações a respeito de interações de usuários maliciosos com um dado serviço da Internet. Um diverso número de ferramentas que implementam o conceito de *honeypot* estão disponíveis, porém não atendem na sua totalidade os requisitos para um *honeypot* DNS que foram identificados neste trabalho. Dessa maneira, optou-se pela implementação de uma solução especifica que atenda a tais requisitos, cuja arquitetura é discutida no Capítulo 4.

4 FERRAMENTA DNSPOT

O objetivo principal deste trabalho é desenvolver um *honeypot* que permita observar as interações de usuários potencialmente maliciosos com servidores DNS recursivos, e analisar o comportamento desses usuários. Este Capítulo apresenta o DNSpot, um *honeypot* projetado especificamente para o monitoramento e coleta de dados para análise de tráfego DNS, descrevendo a arquitetura proposta e como a mesma atende aos requisitos definidos na Seção 3.5, e discutindo os principais aspectos de sua implementação. De acordo com as definições das Seções 3.2 e 3.3, o DNSpot constitui um *honeypot* de pesquisa, uma vez que tem o objetivo de permitir que interações maliciosas ocorram para poder observá-las, e de alta interatividade, pois oferece acesso controlado a um servidor DNS real

4.1 ARQUITETURA

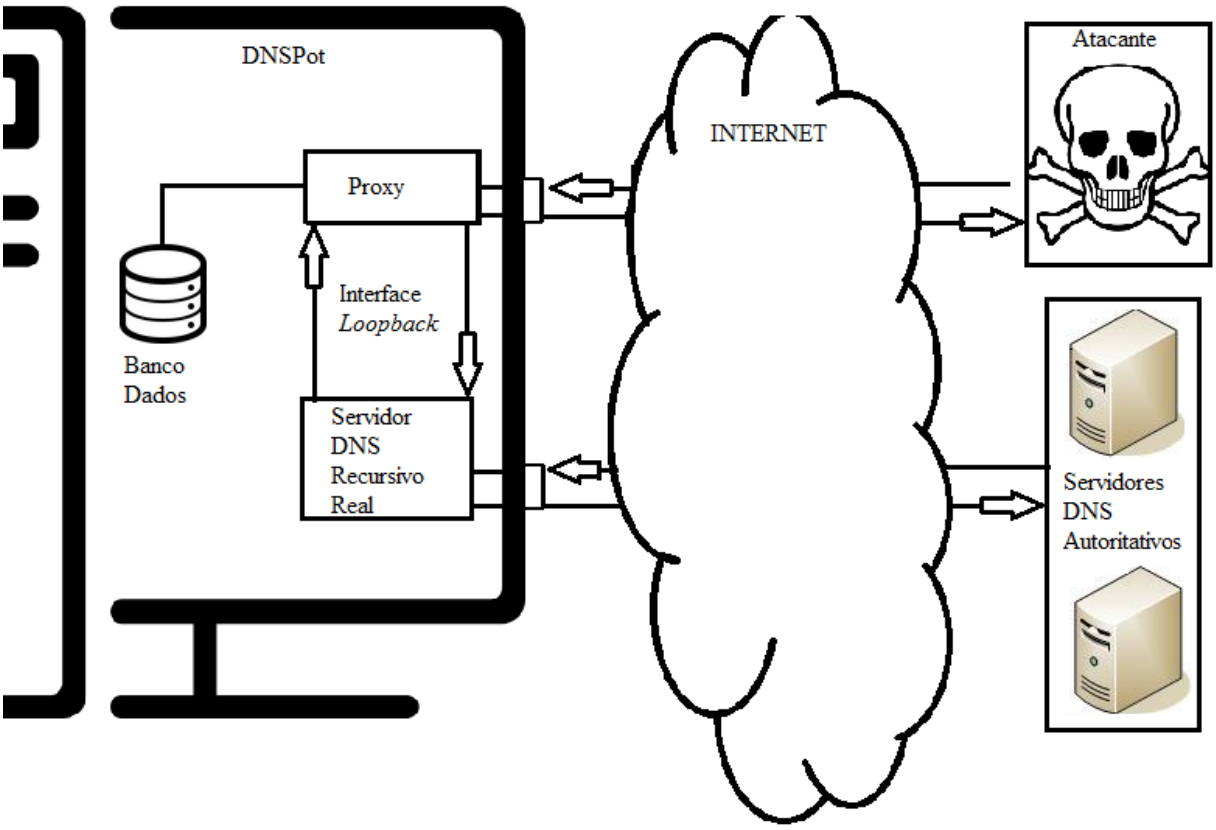
Esta seção introduz a arquitetura do DNSpot, um *honeypot* DNS projetado para atender aos requisitos identificados na Seção 3.5. A arquitetura do DNSpot pode ser observada na Figura 7. Em linhas gerais, existe um componente (chamado de *proxy*) responsável por todas as interações com os clientes (atacantes). Ao receber uma consulta, o *proxy* a armazena em um banco de dados e repassa a um servidor DNS recursivo real, que interage com servidores autoritativos na Internet para efetuar a resolução do nome consultado. A resposta final retornada para o *proxy* pelo servidor real é repassada para o cliente, e igualmente armazenada no banco de dados.

Pode-se observar que essa dinâmica de funcionamento atende ao requisito R1 (resolução recursiva de nomes). Em vez de implementar uma funcionalidade de resolução recursiva ou modificar um servidor DNS já existente, optou-se por manter as funcionalidades de monitoramento e resolução separadas; para a funcionalidade de resolução, qualquer servidor de nomes existente pode ser usado. Para ajudar a garantir o requisito R2 (indistinguibilidade), o servidor DNS real reside no próprio DNSpot (isto é, no mesmo *host*), aceitando apenas requisições locais (via interface de *loopback*), o que impede que ele receba consultas através da rede. Isso significa que a interação entre servidor real e *proxy* é invisível a observadores externos, e

que todas as consultas a servidores autoritativos externos virão do DNSpot. Outro aspecto referente ao requisito R2 é que o *proxy* interfere o mínimo possível nas requisições e respostas DNS, repassando-as diretamente ao servidor recursivo real ou ao cliente/atacante.

.

Figura 7 – Arquitetura da Ferramenta DNSpot



Fonte: Elaborada pelo autor.

Para satisfazer o requisito R3 (não servir de base para ataques), o DNSpot deve, após receber uma consulta, decidir se a repassa ao servidor real para resolver o nome e devolve a resposta ao cliente, se envia uma mensagem de erro, ou se ignora a consulta recebida. O critério de decisão escolhido abrange duas abordagens:

- Randomização: o analista de segurança define uma porcentagem de consultas que receberão respostas reais, e o *proxy* decide aleatoriamente (segundo a probabilidade definida) se envia uma resposta ou um erro;
- Limitação de consultas: o analista de segurança define um limite diário para a taxa ou para o número de consultas atendidas para cada IP de cliente, e,

caso o limite seja excedido, o *proxy* simplesmente passa a ignorar novas consultas até o dia seguinte, quando volta a aceitar consultas do IP ignorado.

Um mensagem de erro é uma resposta DNS com RCODE=2 (*ServFail*), que, conforme discutido na Seção 2.4.1, sinaliza uma falha inespecífica do servidor DNS. A ideia da mensagem de erro é que o *honeypot* se comporte como um servidor DNS recursivo funcional, ainda que às vezes pouco confiável, e com isso os atacantes se sintam confortáveis em usá-lo (o que contribui também para o requisito R2).

O requisito R4 (monitoramento abrangente) é atendido na medida em que o *proxy* observa todas as requisições e respostas DNS e as armazena no banco de dados. Da mesma forma, o *proxy* é capaz de manipular tanto requisições quanto respostas DNS, o que satisfaz o requisito R5 (possibilidade de modificação). Aspectos relacionados ao requisito R6 (configurabilidade) serão detalhados na Seção 4.4. O requisito R7 está relacionado com a segurança operacional do DNSpot, e as escolhas de implementação usadas para atendê-lo são discutidas na Seção 4.2.

4.2 IMPLEMENTAÇÃO

Uma implementação da ferramenta DNSpot foi desenvolvida. O *proxy* foi implementado em linguagem Python[2] (versão 3.4.3), sendo baseado em uma implementação já existente de um servidor DNS simplificado, denominado *MINI-DNS-SERVER*[3], cujo código é de domínio público. A linguagem Python foi escolhida por ser de fácil entendimento e implementação, por conter uma boa comunidade para suporte na Internet, e por contar com um número de bibliotecas e módulos com classes úteis para a finalidade deste trabalho. Dentre essas destaca-se a DNSLib[4] (versão 0.9.4), biblioteca que permite interpretar e manipular com facilidade mensagens DNS dos mais diversos tipos, uma funcionalidade crucial para este trabalho.

[2] https://www.python.org/
[3] https://github.com/alexsilva/MINI-DNS-Server/tree/master
[4] http://pydoc.net/Python/dnslib/0.9.3/

Para o banco de dados foi utilizado o SQLite3[5]. Para utilizar este banco não é necessário instalar nenhum cliente na máquina, todas as definições de tabelas, índices e os registros são armazenados em um único e simples arquivo facilmente portável. Além disso, já existe uma biblioteca em Python chamada sqlite3[6] (versão 3.8.3.1), que realiza o interfaceamento entre aplicações Python e o SQLite. É um banco simples e fácil de configurar e usar, atendendo as necessidades deste trabalho.

Foi utilizado o servidor Unbound[7] como servidor DNS recursivo real. Outros servidores recursivos, como BIND[8] ou MaraDNS[9], também poderiam ser usados. O Unbound é uma opção popular para servidores DNS recursivos, se caracterizando por bom desempenho e segurança, além de ter configuração simples e flexível. O mesmo suporta todos os recursos e funcionalidades mais atuais do DNS (como EDNS(0), DNSSEC) (UNBOUND, 2006; 2008).

O sistema foi implantado e testado no sistema operacional OpenBSD[10], sendo compatível com qualquer sistema operacional Unix-*like*. Além do OpenBSD ser reconhecido pela segurança, foram tomadas precauções na configuração do sistema (minimização dos serviços ativos, configuração segura do SO e dos serviços, uso de redirecionamento de portas para que o *proxy* execute sem permissão de administrador/*root*).

Os testes confirmaram que a arquitetura funciona como o esperado, respondendo as consultas recebidas e armazenando as informações no banco de dados, como é detalhado na Seção 4.3. Para a finalização deste trabalho, o código implementado foi refinado para corrigir *bugs* e deixá-lo apto a ser colocado em produção. Na sequência o DNSpot foi colocado em produção e monitorado durante 49 dias para coleta de dados, como é apresentado no Capítulo 5.

[5] https://www.sqlite.org/
[6] https://docs.python.org/3.4/library/sqlite3.html
[7] http://www.unbound.net/
[8] http://www.isc.org/software/bind
[9] http://www.maradns.org/
[10] http://www.openbsd.org/

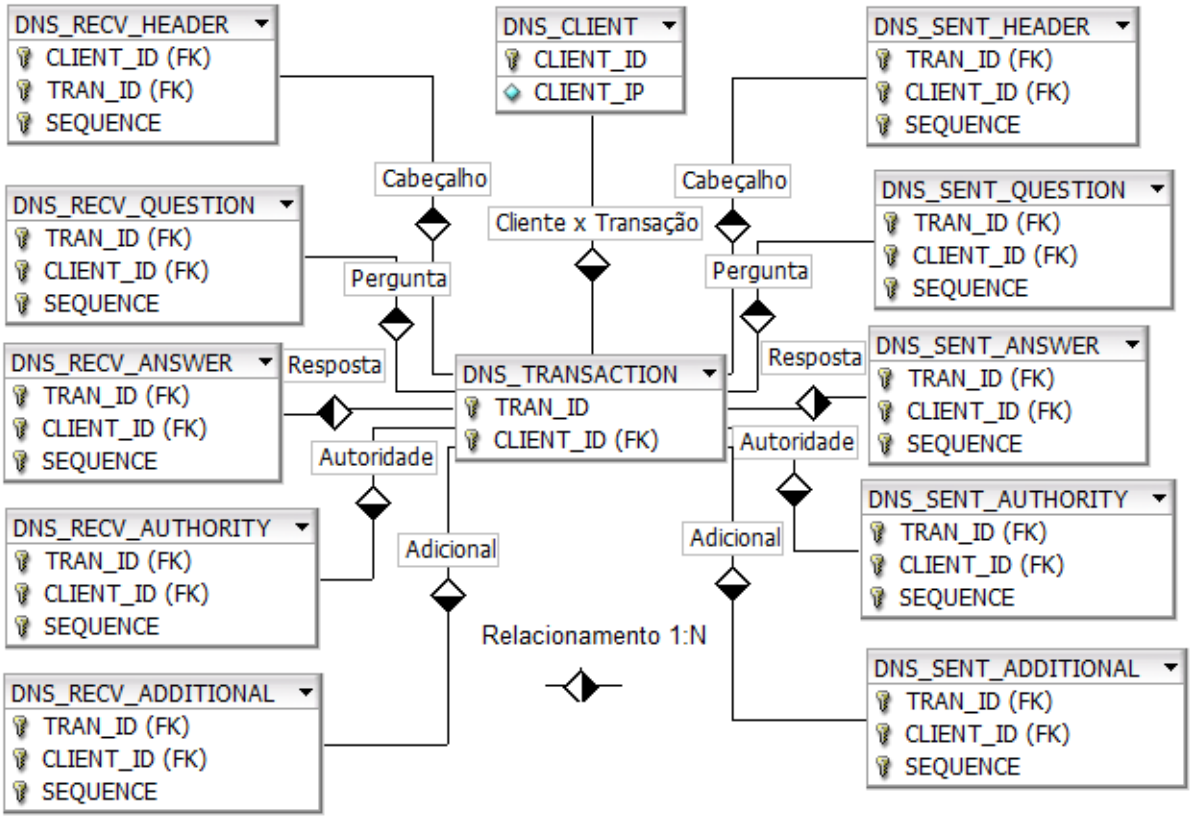
4.3 FLUXO DE FUNCIONAMENTO

O sistema DNSpot é capaz de processar e interpretar todos os campos de uma consulta e resposta DNS padrão. Todas as requisições recebidas pelos clientes, quanto as respostas geradas pelo servidor Unbound devem ser armazenadas em tabelas geradas pelo sistema para que a análise posterior dos dados possa ser realizada.

Todas as transações DNS processadas pelo DNSpot seguem o mesmo fluxo de funcionamento, desde o recebimento de uma requisição DNS até o envio da resposta e persistência dos dados no banco de dados. O componente *proxy* é responsável por interagir diretamente com o cliente no recebimento das requisições e envio das respostas. Ele abre um *socket* UDP na porta configurada para o DNSpot (padrão é a 53) e fica esperando por mensagens nesse *socket*. Para cada mensagem recebida é criada uma nova *thread* no programa para processar a transação DNS até o fim, enquanto a *thread* principal continua escutando a porta por novas mensagens. O uso de *threads* auxiliares para processar cada requisição recebida individualmente permite que o processamento das transações seja feito em paralelo, enviando assim a resposta do cliente o mais rápido possível após finalizar o processo de resolução de nomes em cada *thread*.

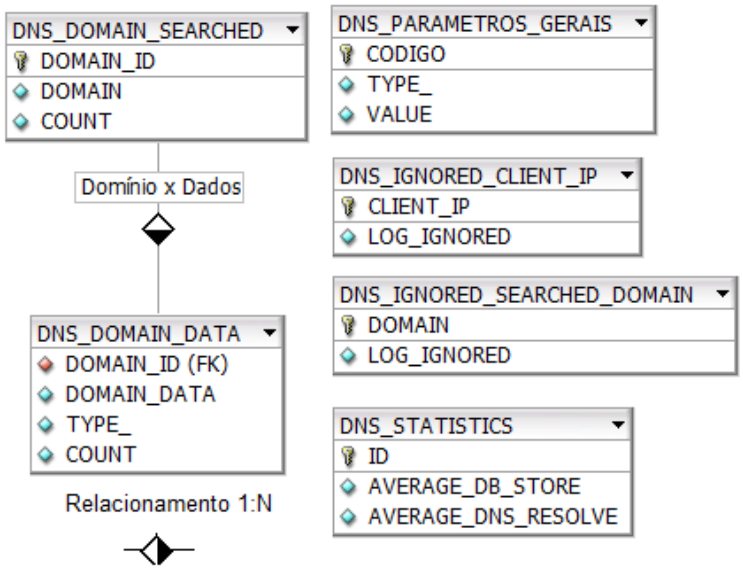
O processamento das mensagens é apoiado por um banco de dados que contém diversas tabelas, cujos relacionamentos estão mostrados na Figura 8, Figura 9 e Figura 10. A Figura 8 mostra os relacionamentos entre as tabelas responsáveis por guardar as informações de todos os campos das mensagens DNS de uma transação. A Figura 9 mostra as tabelas de configuração do funcionamento do sistema, bem como tabelas de estatísticas gerais (não vinculadas a uma transação DNS específica). A Figura 10 mostra as tabelas que guardam informações de erros e exceções ocorridos durante o processamento de uma transação DNS, bem como as tabelas que armazenam as mensagens de requisição e resposta em formato binário e não processadas.

Figura 8 - Diagrama(1) UML das tabelas do DNSpot.



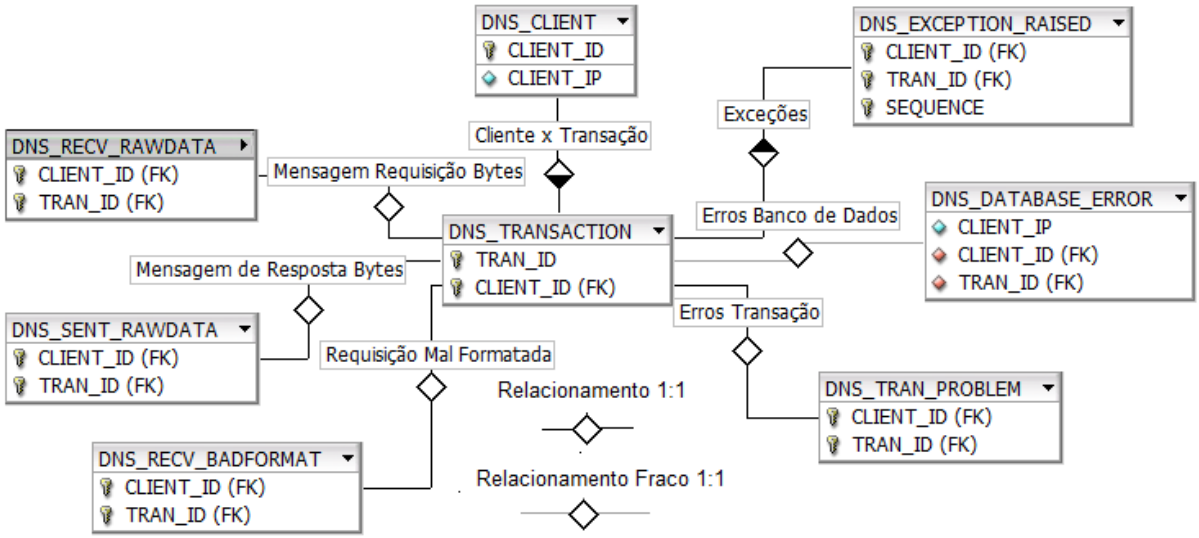
Fonte: Elaborada pelo autor.

Figura 9 - Diagrama(2) UML das tabelas do DNSpot.



Fonte: Elaborada pelo autor.

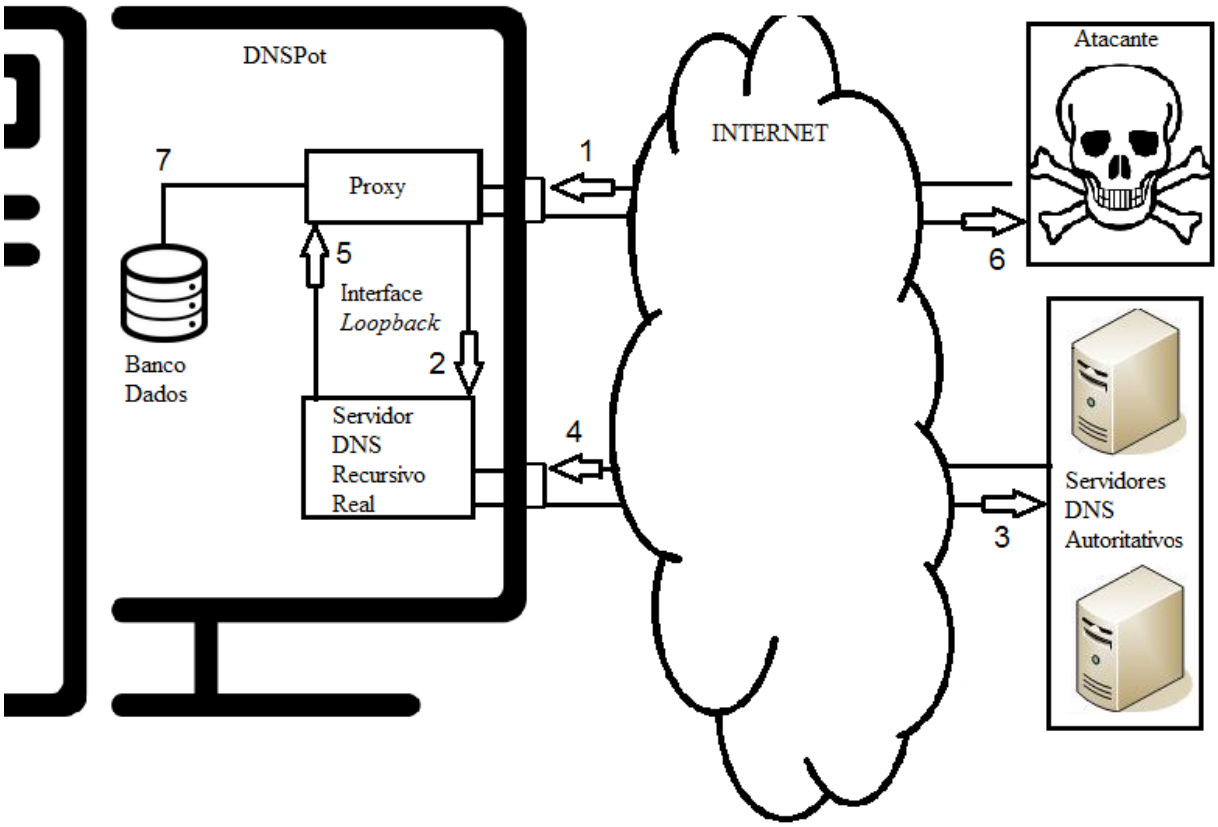
Figura 10 - Diagrama(3) UML das Tabelas do DNSpot.



Fonte: Elaborada pelo autor.

As Seções 4.3.1, 4.3.2 e 4.3.3 descrevem o fluxo de processamento de mensagens, contextualizando o uso das tabelas do banco de dados. Uma descrição mais pormenorizada das tabelas pode ser encontrada no Apêndice A. A Figura 11 mostra as etapas pelas quais passa uma requisição DNS recebida pelo DNSpot, e será usada para auxiliar no entendimento do fluxo de processamento de mensagens.

Figura 11 - Fluxo do processamento de uma requisição pelo DNSpot



Fonte: Elaborada pelo autor.

4.3.1 Recebimento da Requisição DNS

O tratamento de uma transação DNS se inicia quando uma requisição é recebida pela *thread* principal do sistema (passo 1 da Figura 11) que cria uma nova *thread* auxiliar para processar essa requisição. Para cada mensagem recebida é criada uma nova transação na tabela DNS_TRANSACTION; nesse momento é gerado um identificador único para essa transação, que é usado como chave em várias outras tabelas. A mensagem binária recebida é convertida para um objeto Python, usando a DNSLib. Caso a conversão falhe devido à mensagem estar malformada, o erro referente à transação é registrado na tabela DNS_TRAN_PROBLEM. Além disso, o *payload* do datagrama UDP recebido é armazenado na tabela DNS_RECV_BADFORMAT, ficando disponível para análise manual posterior.

Em seguida o sistema realiza uma série de validações para decidir se ignora a transação DNS ou se a repassa para o servidor recursivo, a saber:

- Verificar se o IP do cliente que realizou a requisição está na lista de IPs a serem ignorados (tabela DNS_IGNORED_CLIENT_IP). Isso pode ser usado para desconsiderar requisições de clientes que geram grandes volumes de tráfego ou usados em varreduras automatizadas de servidores DNS (OPEN RESOLVER PROJECT, 2015; SATELLITE, 2015).
- Verificar se o nome do domínio da seção de pergunta da requisição DNS está na lista de domínios a serem ignorados (tabela DNS_IGNORED_ SEARCHED_DOMAIN). Isso pode ser usado, por exemplo, para desconsiderar consultas por nomes que sabidamente geram respostas muito grandes, ou que são usados em varreduras automatizadas de servidores DNS.
- Verificar se o IP do cliente da requisição já atingiu o limite máximo de requisições permitidas por dia (o número de requisições é um atributo do cliente, armazenado na tabela DNS_CLIENT, e o limite máximo é um atributo global do sistema, armazenado na tabela DNS_PARAMETROS_GERAIS). Isso implementa uma limitação da taxa de consultas admitidas, com baixo *overhead*. O objetivo é prevenir que o DNSpot sofra uma negação de serviço por excesso de requisições, e limitar o tráfego gerado quando ele é usado como refletor em um ataque DDoS (Seção 2.6.3 e Seção 2.6.4).

Quando a decisão for de ignorar a requisição, o sistema não realiza o processo de resolução de nomes e grava as informações da transação na tabela DNS_TRANSACTION, passando para a geração do *log* (Seção 4.3.4). Caso contrário é realizado o processo de resolução de nomes, conforme descrito na Seção 4.3.2.

4.3.2 Processo de Resolução de Nomes

O próximo passo no fluxo do sistema é realizar o processo de resolução de nomes da requisição recebida. Para isso, o *proxy* repassa a requisição ao servidor recursivo real (passo 2 da Figura 11) e aguarda a resposta. O servidor real então realiza o processo de resolução de nomes (passos 3 e 4) para a consulta recebida e

repassa ao *proxy* o resultado (passo 5). Após receber a resposta do servidor real, o *proxy* verifica se deve repassar essa resposta ao cliente requisitante ou gerar uma resposta falsa de *ServFail*, com base na porcentagem parametrizada no sistema; a decisão só é tomada nesse momento para permitir que os dados obtidos em resposta à consulta sejam registrados pelo DNSpot. Se houver algum erro ou *timeout* na consulta ao servidor real, o *proxy* responde ao cliente com *ServFail* e registra os dados e o erro no banco de dados (tabela DNS_TRAN_PROBLEM e DNS_EXCEPTION_RAISED).

Uma variação do processo descrito no parágrafo anterior ocorre quando a consulta for pela versão do servidor DNS (QNAME=VERSION.BIND, QCLASS=CHAOS, QTYPE=TXT). Nesse caso, a consulta não é repassada ao servidor real, e o próprio *proxy* sintetiza uma resposta contendo um número de versão definido pelo administrador (armazenado na tabela DNS_PARAMETROS_GERAIS). O administrador pode definir uma versão em branco, ocultando a versão do servidor DNS real, ou uma versão com vulnerabilidades conhecidas, tentando atrair ataques que explorem essas vulnerabilidades para poder estudá-los.

Em todos os casos, a resposta recebida do servidor real ou gerada pelo *proxy* é enviada para o cliente requisitante (passo 6) e o passo de resolução de nomes é finalizado, sendo o próximo passo o processamento, armazenamento e persistência das informações da transação no banco de dados.

4.3.3 Armazenamento no Banco de Dados

Após o cliente ter sido respondido com sucesso, o DNSpot passa à rotina de persistir os dados da transação DNS no banco de dados (passo 7 da Figura 11). As informações a serem persistidas são mantidas em memória até que o processo de resolução de nomes tenha finalizado e o cliente tenha sido respondido, para que a resposta seja enviada o mais rápido possível ao cliente, diminuindo o *overhead* entre recebimento da requisição e envio da resposta.

Nessa fase as mensagens de requisição e resposta são processadas e os dados DNS extraídos e preparados para persistência no banco. Cada transação DNS (Requisição + Resposta) é identificada unicamente por um registro na

DNS_TRANSACTION, que guarda informações gerais e resumidas acerca da transação realizada, e é referenciado pelas outras tabelas para ligar os dados DNS e não DNS à transação.

Além disso, o IP do cliente (atacante ou vítima) interagindo com o DNSpot é armazenado no sistema pela tabela DNS_CLIENT, para guardar estatisticas e controlar a interação do mesmo com o sistema ao longo do tempo. Cada cliente possui um identificador único que é usado como chave em outras tabelas (em conjunto com o identificador da DNS_TRANSACTION). Os dados referentes aos domínios pesquisados e das respostas associadas a esses domínios são também registrados no sistema para contagem das ocorrências (tabelas DNS_DOMAIN_SEARCHED e DNS_DOMAIN_DATA).

Os dados DNS referentes a mensagem de consulta são armazenados nas tabelas DNS_RECV_HEADER, DNS_RECV_QUESTION, DNS_RECV_ANSWER, DNS_RECV_AUTHORITY e DNS_RECV_ADDITIONAL. Por sua vez, os dados das respostas DNS enviadas são armazenados nas tabelas DNS_SENT_HEADER, DNS_SENT_QUESTION, DNS_SENT_ANSWER, DNS_SENT_AUTHORITY e DNS_SENT_ADDITIONAL.

Além das informações das mensagens DNS, dados estatisticos (tabela DNS_STATISTICS) e de controle do sistema (tabela DNS_TRAN_PROBLEM); e erros ocorridos e relacionados ao processamento da transação DNS (tabelas DNS_TRAN_PROBLEM e DNS_EXCEPTION_RAISED) também são enviados ao banco para monitorar o bom funcionamento do sistema. Por último, o *payload* do datagrama UDP recebido é armazenado no sistema (tabelas DNS_RECV_RAWDATA e DNS_SENT_RAWDATA) para permitir que um reprocessamento dos dados DNS possa ser realizado caso se mostre necessário, devido a erro no processamento atual ou necessidade de extrair outras informações.

Se algum erro de banco ocorrer, a transação é marcada como inconsistente (no que tange o banco) e quais tabelas foram corretamente inseridas ou atualizadas são marcadas (tabela DNS_DATABASE_ERROR). Finalizado a persistência dos dados, a *thread* vai para o último passo, que é a escrita da linha no arquivo de *log.*

4.3.4 Geração do Arquivo de *Logs*

O último passo no processamento da transação pela *thread* é o registro de um resumo da transação, em formato ASCII, em um arquivo de *log*. Isso facilita o acompanhamento das requisições processadas pela ferramenta e o monitoramento de sua operação.

Para cada transação, as seguintes informações são registradas no *log*:

- data e hora da requisição;
- endereço IP e porta de origem;
- identificadores únicos das tabelas DNS_TRANSACTION e DNS_CLIENT gerados para a transação (para facilitar consultas subsequentes no BD);
- nome, classe e tipo do domínio consultado;
- para consultas respondidas com RCODE=0 (*NoError*), as quantidades de RRs presentes nas seções de resposta, autoridade e adicional da resposta;
- para respostas com erro ou não respondidas, um código indicando o resultado, que pode ser o próprio RCODE, uma indicação de *ServFail* forjado, ou uma indicação de que a consulta foi ignorada (e o motivo);
- erros e exceções no processamento de uma transação (quando houver).

A Figura 12 mostra alguns exemplos de *logs* gerados pelo DNSpot. As linhas 1, 2 e 3 representam transações respondidas com sucesso. A linha 4 corresponde a uma consulta por nome inexistente (RCODE=*NXDomain*). A linha 5 mostra um *ServFail* falso, gerado pelo próprio DNSpot. As linhas 6, 7 e 8 mostram casos de requisições ignoradas pelo domínio (linha 6) ou o endereço IP de origem (linha 7) estarem em uma *blacklist*, ou pelo IP de origem ter excedido o limite diário de requisições (linha 8). Por fim, a linha 9 ilustra uma exceção no acesso ao banco de dados.

Figura 12 - Exemplo de *log.*

```
1 2015-10-27 01:20:00.779805 186.207.222.4:61501 65 1 quark.das.ufsc.br. IN A 1-0-0
2 2015-10-27 10:27:12.139351 177.67.91.100:52389 54 103 quark.das.ufsc.br. IN AAAA 0-1-0
3 2015-10-28 05:27:35.003536 185.94.111.1:49539 4 112 com. IN ANY 21-0-1
4 2015-10-28 10:27:19.605308 177.67.91.100:56736 55 103 quark.das.ufsc.br.corporate.corp. IN A NXDOMAIN
5 2015-10-29 01:20:00.360488 186.207.222.4:61499 63 1 quark.das.ufsc.br. IN A SERVFAIL(f)
6 2015-10-30 00:20:26.894020 185.207.222.4:61501 1 1314 dnsscan.shadowserver.org. IN A IGNORED(by Domain)
7 2015-10-30 00:20:26.894020 187.207.222.4:61501 1 1314 quark.das.ufsc.br. IN A IGNORED(by IP)
8 2015-10-30 00:20:26.894020 188.207.222.4:61501 1 1314 quark.das.ufsc.br. IN A IGNORED(by IP-Max-Requests)
9 2015-10-31 19:25:20.855558 178.218.223.68:80 20 1080 google.com. IN A (Transaction Not Sucessfully) (DataBase Error) (Exception)
```

Fonte: Imagem elaborada pelo autor.

Após a geração do *log*, a *thread* chega ao final do seu ciclo de vida e é encerrada para liberar os recursos de memória alocados do sistema. Quanto mais tempo uma *thread* fica em execução, mais tempo ela permanece com os recursos alocados.

4.3.5 Erros, *Bugs* e Exceções de Processamento

Durante o fluxo de processamento de uma transação DNS, existe a possibilidade de erros ocorrerem no sistema, na interação com o servidor real ou com o banco de dados. A rotina do sistema ao se deparar com erros e exceções inesperados é gravar todos em tabelas do banco de dados, para que possam ser detectados, analisados e corrigidos em uma nova versão da implementação. Erros e exceções da aplicação ou erros relacionados à comunicação com o banco de dados são armazenads na DNS_EXCEPTION_RAISED. Um resumo dos erros ocorridos na aplicação e durante a persistência no banco de dados é armazenado nas tabelas DNS_TRAN_PROBLEM e DNS_DATABASE_ERROR respectivamente. Se o erro for grave ao ponto de impossibilitar o processamento da transação, o cliente requisitante nestes casos recebe uma mensagem gerada de *ServFail*, indicando erro transiente no servidor DNSpot.

4.4 ASPECTOS CONFIGURÁVEIS (PARAMETRIZAÇÕES)

O DNSpot permite que o analista de segurança possa realizar uma série de configurações e parametrizações. Essas configurações afetam o seu funcionamento e os aspectos de segurança do mesmo. Os aspectos configuráveis são:

- Interface de rede do *proxy*: É possível definir a qual interface da máquina o DNSpot irá ser vinculado para receber e responder as transações DNS.
- Porta de escuta: A porta na qual o DNSpot irá escutar para receber as requisões DNS.

- Lista de IPs a serem ignorados: Quais clientes, baseados em seus IPs, terão suas requisições ignoradas pelo sistema.
- Lista de nomes de domínio a serem ignorados: Quais requisições DNS serão ignoradas, com base no dominio requisitado na seção de pergunta.
- Número máximo de requisições diárias processadas por cliente: O número máximo de requisições diárias que o DNSpot irá processar e responder, para cada cliente.
- *Timeout* dos *Sockets*: Ao repassar a requisição DNS do *proxy* para o servidor recursivo real realizar o processo de resolução de nomes, é possível definir o tempo ou *timeout* que o *proxy* irá esperar pela resposta vinda do servidor antes de emitir um *ServFail* (RCODE=2) ao cliente requisitante.
- Consulta VERSION.BIND: A consulta pelo domínio VERSION.BIND (classe CHAOS e tipo TXT) faz com que os servidores respondam informando a versão de implementação dos mesmos, se estiverem configurados para tal. O DNSpot permite que seja configurado qual resposta deve ser enviada para esse tipo de requisição. Ou seja, é possível utilizar esse recurso para “enganar” o atacante e faze-lo pensar que está interagindo com uma versão de implementação antiga que possuí *bugs* e vulnerabilidades conhecidas que já foram corrigidas em versões posteriores, estimulando um possível interação e ataque direcionado a versão
- Porcentagem de *ServFail* (RCODE=2): É possivel definir a taxa ou porcentagem de transações DNS recebidas que serão respondidas com falha de servidor ao cliente. Dessa maneira é possível simular um servidor DNS de implementação instável e falha, que pode conter vulnerabilidades interessantes ao atacante.

4.5 CONSIDERAÇÕES PARCIAIS

Neste Capítulo é discutida a arquitetura da ferramenta DNSpot e como ela satisfaz os requisitos definidos na Seção 3.5. Conforme discutido na Seção 3.1, um *honeypot* oferece o risco de comprometimento da rede interna da instituição, ou seja, um meio para atacantes conseguirem uma brecha aos sistemas internos. Para

minimizar os riscos, o DNSpot usa um sistema operacional e um servidor DNS reconhecidos pela sua segurança, configurados de forma restritiva. Além disso, a aplicação executa como um usuário não privilegiado e implementa mecanismos para limitar o tráfego gerado em ataques de amplificação. A Tabela 2 sumariza os requisitos e quais os mecanismos usados pelo DNSpot para satisfazê-los.

Tabela 2 – Mecanismos do DNSpot que atendem aos requisitos de um *honeypot* para DNS.

Requisito	Mecanismos
R1: Resolução de nomes	Presença de um servidor recursivo real
R2: Indistinguibilidade	*Proxy* e servidor real residem na mesma máquina; Mínima interferência nas mensagens
R3: Evitar abuso/ataque	*ServFail* falso; Limitação de consultas diárias por IP
R4: Monitoramento e coleta de dados	Consultas e respostas passam pelo *proxy*, que as armazena no banco de dados (DNSLib + Sqlite)
R5:Manipulação de mensagens	Capacidade do *proxy* de interpretar e manipular as mensagens DNS (DNSLib)
R6: Configurabilidade	Apectos de funcionamento alteráveis (Seção 4.4)
R7: Contenção/mitigação	Uso de sistema operacional e servidor DNS reconhecidos pela segurança e configurados de forma restritiva e segura (Ex: mínimo privilégio de usuário)

FONTE: Tabela elaborada pelo autor.

A grande limitação da ferramenta é estar restrita a coletar e processar dados e interações que cheguem a ela. Se o atacante não interagir com a ferramenta, os dados não poderão ser coletados sobre o mesmo. Além disso, nem todas as funcionalidades estendidas do DNS, tal como o DNSSEC, serão suportadas e processadas pela ferramenta (mas os dados de requisições que usam essas funcionalidades serão coletados para análise).

No Capítulo 5, é realizada uma análise de dados coletados com o DNSpot em um período de 49 dias, entre entre 9 de setembro e 28 de outubro de 2015.

5 RESULTADOS E ESTATÍSTICAS

O DNSpot foi implantado e posto em produção na rede da Universidade do Estado de Santa Catarina (UDESC). Este capítulo faz uma análise dos dados coletados pelo DNSpot no período entre 09/09/2015 e 28/10/2015, que corresponde a 49 dias de observação. Inicialmente é descrito o ambiente operacional em que o DNSpot foi implantado; na sequência, são apresentadas estatísticas gerais sobre o tráfego observado, os ataques identificados no período e considerações sobre o desempenho do banco de dados adotado.

5.1 IMPLANTAÇÃO

O sistema foi implantado em uma máquina na rede interna do Departamento de Ciência da Computação (DCC) da UDESC. Como essa rede usa endereços reservados (DE GROOT *et al*, 1996), foi usado NAT (*Network Address Translation*) para redirecionar o tráfego destinado à porta 53/UDP de um endereço IP público ocioso para a mesma porta no DNSpot, sem nenhuma filtragem preliminar. As configurações de sistema operacional, *hardware* e *software* da máquina são as seguintes:

- Sistema Operacional: OpenBSD 5.7 i386;
- Processador: Intel Core2 Duo CPU E6550 @ 2.33GHz ("GenuineIntel" 686-class);
- Memória RAM: 1 GB;
- Rede Ethernet 100 mbps
- Servidor DNS recursivo: Unbound, versão 1.5.2;
- Python, versão 3.4.2;
- DNSLib, versão 0.9.4.
- SQLIte3, versão 3.8.6.

Com relação aos parâmetros de configuração introduzidos na Seção 4.4, a porcentagem de *ServFail* foi definida em 20%, e a quantidade máxima de

requisições por dia foi inicialmente fixada em 100, valor que foi reduzido para 30 após terem sido observados os primeiros ataques de negação de serviço. A quantidade máxima de requisições diárias foi escolhida analisando o montante de tráfego ao DNSpot nos primeiros dias, com o objetivo de achar um valor que não gerasse muito tráfego de ataque DoS pelo DNSpot e que, ao mesmo tempo, não espantasse um usuário malicioso que queira usar o sistema para ataques de amplificação. A porcentagem de *ServFail* foi escolhida tentando evidenciar para um atacante humano (i.e., não uma ferramenta automatizada) que o servidor tem um comportamento moderadamente errático sem contudo levá-lo a desistir de interagir com o sistema.

Observações de tráfego DNS preliminares à implantação do DNSpot revelaram a ocorrência de consultas que fazem parte de varreduras (*scans*) "benignas", isto é, que buscam por servidores vulneráveis para fins de pesquisa ou mesmo para poder alertar os responsáveis da existência do problema. Em comum, essas consultas têm a característica de pesquisarem nomes de domínio fixos (ou com um sufixo bem definido). O DNSpot foi configurado para ignorar consultas por tais nomes (listados na Tabela 3), para não ser identificado nessas varreduras como um servidor recursivo aberto.

Tabela 3 - Sufixos ignorados pelo DNSpot

Sufixo	Referência
dnsresearch.cymru.com	http://dnsresearch.cymru.com
dnsscan.shadowserver.org	http://dnsscan.shadowserver.org
openresolverproject.org	http://openresolverproject.org
openresolvertest.net	http://openresolverproject.org[11]
satellite.cs.washington.edu	http://satellite.cs.washington.edu
syssec.rub.de	http://scanresearch.syssec.rub.de

FONTE: Tabela elaborada pelo autor.

[11] As consultas que usam o sufixo openresolvertest.net provêm do endereço IP 204.42.253.2, cuja resolução reversa é openresolverproject.org.

5.2 ESTATÍSTICAS DE TRÁFEGO

Durante o período de monitoramento, o DNSpot recebeu mais de quatro milhões de requisições. Esta seção faz uma análise geral do tráfego observado, descrevendo o período de monitoramento, o volume e distribuição das transações, e caracterizando os ataques DoS recebidos.

5.2.1 Período de monitoramento

A coleta de dados com o DNSpot começou em 09/09/2015 às 07:57:02, e os resultados descritos neste documento consideram as observações até 28/10/2015 às 22:29:48. Isso corresponde a 49 dias de observação, conforme pode ser visto na Tabela 4. Cabe notar aqui que, durante esse período, ocorreram diversas interrupções no serviço, devido basicamente a três causas:

1. Queda de energia afetando o host onde o DNSpot é executado;
2. Problema na inicialização do DNSpot após retorno de energia;
3. Interrupção do *firewall* que dá acesso da *Internet* para o DNSpot, por queda de energia ou panes decorrentes de problemas de hardware.

Embora não haja dados conclusivos, a estimativa da administração da rede do DCC é que, somadas, as interrupções nesse período devem ter ocasionado entre 24 e 72 horas de indisponibilidade do serviço (*downtime*) oferecido pelo DNSpot.

Tabela 4 - Periodo do DNSpot em produção.

Início	09/09/2015 07:57
Fim	28/10/2015 22:29
Total (segundos)	4.285.966
Total (minutos)	71.432
Total (horas)	1.190
Total (dias)	49

FONTE: Tabela elaborada pelo autor.

5.2.2 Transações

Durante os dias que o DNSpot ficou ativo, foi recebido um total de 4.035.605 transações DNS, sendo a maioria aparentemente de DoS usando o DNSpot como refletor. Isso foi um montante de consultas maior que o esperado inicialmente neste trabalho, principalmente pelo fato de se tratar de um servidor DNS não anunciado, tendo assim que ser descoberto pelos atacantes para ser explorado. Isso leva à hipótese de que usuários maliciosos realizam constantemente varreduras pela rede à procura de máquinas com serviço DNS que possam ser usadas para fins de ataques, que uma vez encontradas passam a ser exploradas. A Tabela 5 sumariza algumas informações referentes ao montante de transações recebidas pelo DNSpot. Como pode ser observado, 99,995% das requisições recebidas foram consultas que sinalizaram o uso do protocolo EDNS(0), que permite mensagens DNS maiores que 512 bytes, o que aumenta o fator de amplificação entre requisição/resposta

Tabela 5 - Resumo das transações DNS. Porcentagens[12] em itálico representam a proporção dentro de uma categoria.

Transações	quantidade	%
Respondidas	488.299	12,1
• Válidas	391.050	*80,1*
• *ServFail* falso	97.249	*19,9*
Não respondidas	3.547.306	87,9
• Ignoradas	3.544.876	*99,9*
• Erros	2.370	*0,1*
Total	4.035.605	100,0
• EDNS(0)	4.035.409	99,995

FONTE: Tabela elaborada pelo autor.

Do total de consultas recebidas, apenas 12,1% foram respondidas de fato ao cliente. Dentre as respostas, 80,1% foram respostas válidas do servidor recursivo (esse número inclui os erros de resolução sinalizados por esse servidor) e 19,9%

[12] Ao longo do capítulo, os resultados são expressos preferencialmente com uma casa decimal. Em algumas tabelas, porém, é usada maior precisão para permitir a diferenciação de números distintos.

foram *ServFails* falsos gerados pelo DNSpot (a porcentagem configurada era de 20%).

Das 3.547.306 requisições que não tiveram resposta (87,9% do total), apenas 2.370 requisições (0,06% do total) não foram respondidas de fato, devido a algum problema transiente no DNSpot. As 99,94% restantes foram ignoradas pelo DNSpot, em função das configurações descritas na Seção 5.1. A Tabela 6 mostra o total de transações ignoradas e quais as regras foram aplicadas. Dentre as transações ignoradas, apenas 51 (0,0014%) foram pela regra de filtragem de domínio e 3.544.825 foram pela regra de número máximo de consultas diárias por IP. Durante o período observado não foi usada a *blacklist* de endereços IP.

Tabela 6 - Transações ignoradas por regras

Transações	quantidade	%
Ignoradas	3.544.876	100,0000
• Blacklist de domínios	51	0,0014
• Blacklist de IPs	0	0,0000
• Máximo diário atingido	3.544.825	99,9986

FONTE: Tabela elaborada pelo autor.

Dentre as respostas válidas enviadas ao cliente (das consultas respondidas), diferentes RCODES foram enviados indicando sucesso ou erro no processamento da transação. Conforme mostrado na Tabela 7, 384.863 consultas (78,8%) foram processadas e respondidas com sucesso ao cliente, ou seja, com *NoError*. Os códigos *FormError* (três respostas), *NXDomain* (3.565 respostas), *NotImp* (três respostas) e *Refused* (27 respostas) somados corresponderam a aproximadamente 0,74% das respostas. Foram enviadas ainda 99.838 respostas com *ServFail*, sendo 2.589 legítimas (*ServFail* gerado pelo Unbound) e 97.249 geradas pelo servidor DNSpot com base na porcentagem estabelecida.

Tabela 7 - RCODEs enviados ao Cliente.

RCODE	Quantidade	%
NoError	384.863	78,817
FormErr	3	0,001
ServFail falso	97.249	19,916
ServFail real	2.589	0,527
NXDomain	3.565	0,726
NotImp	3	0,001
Refused	27	0,006
Unknown	0	0,000
Total	488.299	100,000

FONTE: Tabela elaborada pelo autor.

Considerando o período de observação apresentado na Tabela 4 e as quantidades de transações resumidas na Tabela 5, a taxa média observada foi de 0,94 transações por segundo, ou cerca de 81.353 transações por dia. Essa taxa é, na verdade, uma subestimativa: por ser desconhecido, o tempo de *downtime*, durante o qual não são registradas transações, não pode ser descontado do tempo de observação, o que reduz a taxa calculada. Como pode ser visto na Tabela 8, porém, levar em consideração as estimativas de *downtime* disponíveis pouco altera as taxas calculadas. Logo, na sequência, os resultados que envolverem o tempo de coleta irão desconsiderar o *downtime*.

Tabela 8 - Transações por período.

	Transações/s	Transações/dia
Sem *downtime*	0,94	81.353,02
Downtime de 24h	0,96	83.026,74
Downtime de 72h	1,00	86.589,65

FONTE: Tabela elaborada pelo autor.

5.2.3 Volume de dados em bytes

O volume de dados processado durante o período monitorado totalizou 1.560,1 MB, sendo 165,1 MB (10,6%) de consultas recebidas e 1.395,0 MB (89,4%) de respostas enviadas. O número relacionado às respostas seria ainda maior se não houvesse o mecanismo de limitação de consultas diárias para mitigar os efeitos de

ataques DoS. Usando o tamanho médio das respostas válidas para cada RR (desconsiderando *ServFails* falsos) para extrapolar os bytes que seriam de fato enviados nas respostas às requisições ignoradas, o DNSpot teria enviado cerca de 14.610,4 MB somente em respostas (14.775,6 MB no total). Percebe-se, portanto, que a limitação de consultas diárias processadas reduziu em 90,5% o tráfego gerado pelo *honeypot*, evidenciando assim a sua eficácia. A Tabela 9 resume os dados sobre volume de tráfego.

Tabela 9 - Volume de tráfego processado e esperado

Tipo de tráfego	volume (MB)	%
Tráfego processado	1.560,1	100,0%
• Consultas	165,1	10,6%
• Respostas	1.395,0	89,4%
Tráfego esperado	14.775,6	-
• Respostas	14.610,4	-
Redução de tráfego de resposta	13.215,4	90,5%

FONTE: Tabela elaborada pelo autor.

O tamanho das consultas recebidas variou entre 1 e 413 bytes, com média de 42,9 bytes e desvio padrão de 3,3 bytes. O tamanho tem uma distribuição assimétrica positiva (concentração de valores pequenos); dos mais de 4 milhões de consultas recebidas, apenas 6 eram maiores que 100 bytes. O tamanho das respostas válidas, por sua vez, variou entre 12 e 4.096 bytes (limite do servidor DNS recursivo com EDNS(0)), com média de 3.734 bytes e desvio padrão de 577,4 bytes (ou seja, o tamanho das respostas apresenta maior variação que o tamanho das consultas). A distribuição do tamanho das respostas é assimétrica negativa (concentração de valores grandes), mas com uma maior proporção de mensagens pequenas – das 391.050 respostas válidas, 6.232 (1,59%) tinham até 200 bytes. A Tabela 10 apresenta estatísticas descritivas dos tamanhos de consultas e respostas; o 99º percentil do tamanho das consultas e o 5º percentil do tamanho das respostas evidenciam a assimetria das distribuições. Na Tabela 11 são mostrados os cinco tamanhos mais frequentes de consultas e respostas, juntamente com a sua frequência relativa. Enquanto as consultas mais frequentes são todas pequenas,

entre 36 e 64 bytes, dois dos cinco tamanhos mais frequentes de resposta (1.503 e 2.019 bytes, agregando 2,46% das observações) encontram-se fora da região entre o 5º percentil e o máximo, a qual abrange 95% de todas as observações.

Tabela 10 - Estatísticas de consultas e respostas.

estatísticas	tamanho (bytes)	
	consultas	respostas
Média	42,9	3.734
Desvio padrão	3,3	577,4
Mínimo	1	12
1º percentil	35	96
5º percentil	39	3.850
1º quartil	39	3.850
Mediana	44	3.850
3º quartil	44	3.850
95º percentil	44	3.853
99º percentil	64	3.892
Máximo	413	4.096

FONTE: Tabela elaborada pelo autor.

Tabela 11 - Cinco tamanhos mais frequentes de consultas e respostas.

consultas		respostas	
tamanho (bytes)	freq. relativa (%)	tamanho (bytes)	freq. relativa (%)
44	72,26	3850	90,38
39	23,88	3853	2,97
64	1,16	1503	1,47
47	0,86	3892	0,99
36	0,63	2019	0,99
freq. total	98,79	freq. total	96,80

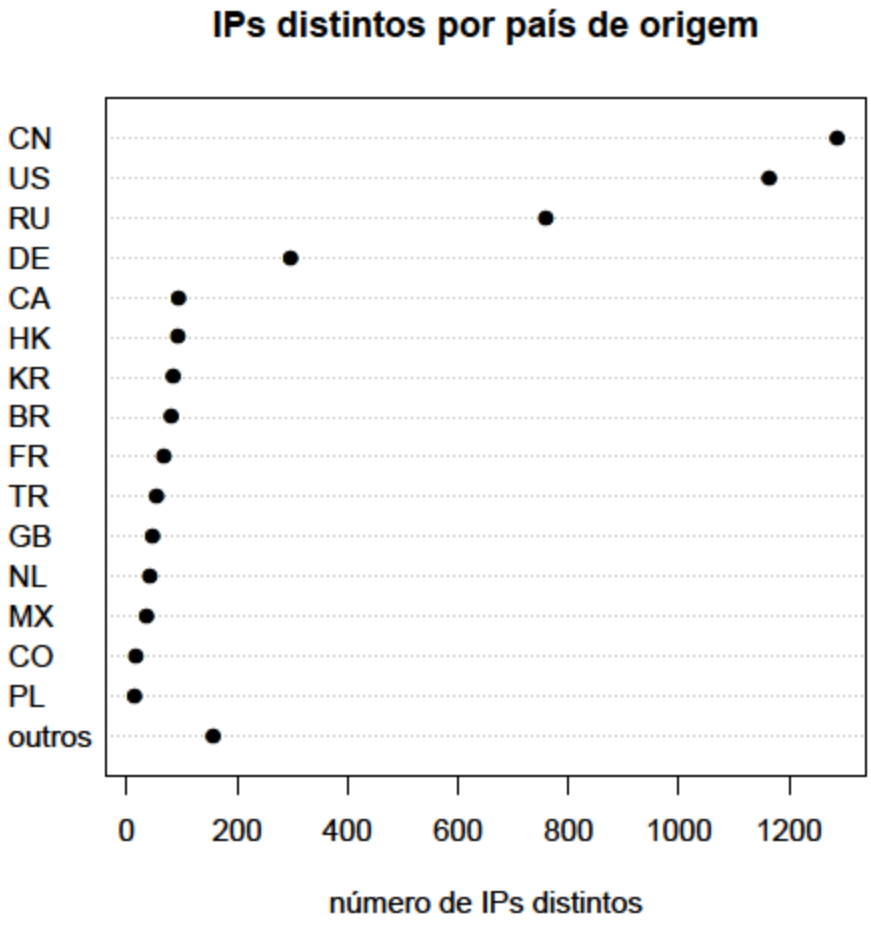
FONTE: Tabela elaborada pelo autor.

Cruzando os dados da Tabela 9 com os dados da Tabela 4, pode-se determinar a taxa de bytes verificada no DNSpot. Considerando o tráfego total, a taxa ficou em 381,69 bytes/s, o que dá 31,45 MB/dia.

5.2.4 Clientes IP, Domínios e RRs

O DNSpot recebeu consultas de 4.287 endereços IP distintos durante o período de observação. A Tabela 12, ilustrada na Figura 13, mostra a distribuição de endereços por país, usando dados de geolocalização disponíveis publicamente na Internet (FREEGEOIP.NET, 2015). Os dois que aparecem com mais frequência, China (1.287) e Estados Unidos (1.164), respondem por 57% dos IPs. Cabe observar aqui que essa análise não distingue entre clientes que realizam consultas e endereços de vítimas de ataques DoS por amplificação. Essa não diferenciação pode ajudar a explicar as posições de destaque de China, Hong Kong e Coréia do Sul, países frequentemente envolvidos em rumores de ataques cibernéticos (CULPAN, 2015; FIFIELD, 2015 ; NETWORKS ASIA, 2015). Muitos dos países mostrados na Figura 13 também aparecem com destaque no mapa de ataques DDoS disponibilizado pela Arbor Networks (ARBOR NETWORKS, 2015).

Figura 13 - IPs distintos por país de origem.



FONTE: Figura elaborada pelo autor.

Tabela 12 - Países de origem das consultas ao DNSpot.

País de origem	Nº de IPs distintos	%
China (CN)	1.287	30,0
Estados Unidos (US)	1.164	27,2
Rússia (RU)	759	17,7
Alemanha (DE)	297	6,9
Canadá (CA)	94	2,2
Hong Kong (HK)	92	2,1
Coreia do Sul (KR)	84	2,0
Brasil (BR)	80	1,9
França (FR)	67	1,6
Turquia (TR)	54	1,3
Grã-Bretanha (GB)	46	1,1
Holanda (NL)	42	1,0
México (MX)	35	0,8
Colômbia (CO)	16	0,4
Polônia (PL)	14	0,3
outros	156	3,6
Total	4.287	100,0

FONTE: Tabela elaborada pelo autor.

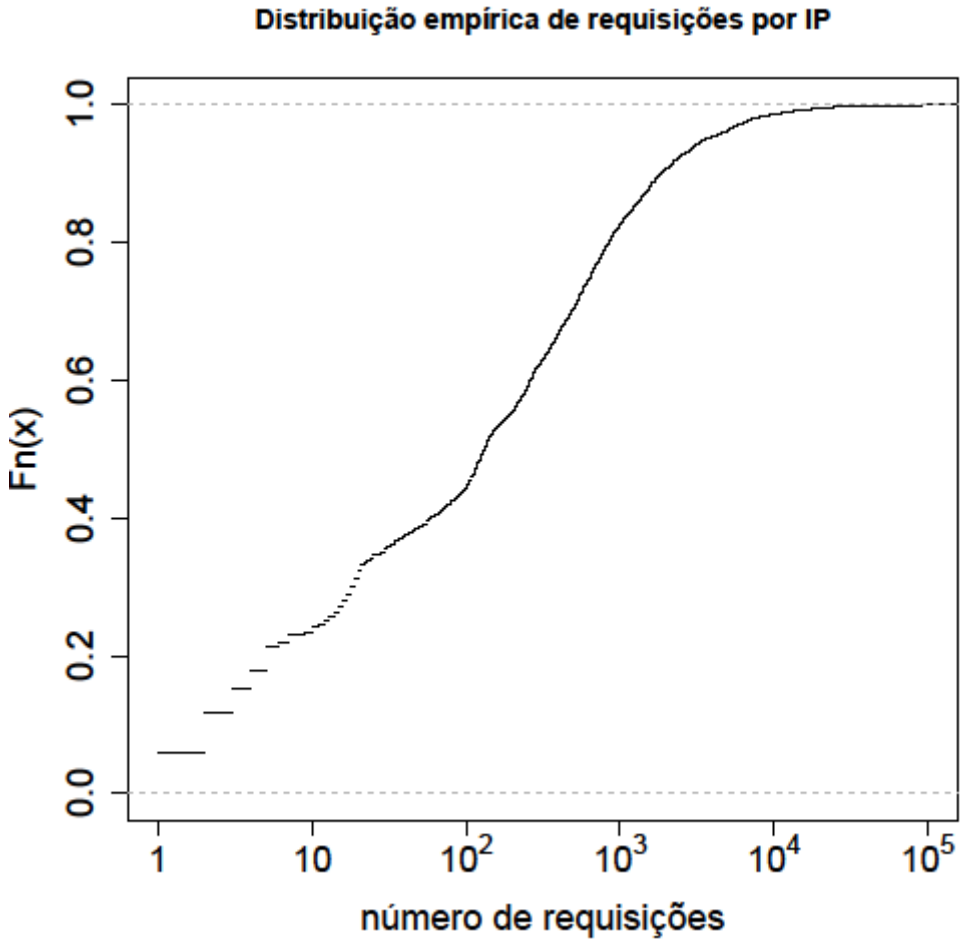
O número de consultas por IP variou de 1 a 98.217, sendo a média 941,4 e o desvio padrão 3.964,2, conforme a Tabela 13. Os dados têm uma distribuição assimétrica positiva (concentração de pequenos valores); isso é evidenciado pela distribuição empírica mostrada na Figura 14 (cuja escala do eixo x é logarítmica). Cerca de 40% dos IPs fizeram até 100 consultas, e cerca de 80% fizeram até 1.000 consultas, ou seja, a maioria dos IPs possui baixo número de consultas e poucos possuem alto índice de consultas. Além disso, 99% dos IPs fizeram aproximadamente 12.275 consultas (12,5% do máximo).

Tabela 13 - Estatísticas de número de transações por IP.

Estatísticas	Número de consultas
Média	941,4
Desvio padrão	3.964,2
Mínimo	1
1º quartil	13
Mediana	132
3º quartil	644
95º percentil	3.623,2
99º percentil	12.276,4
Máximo	98.217

FONTE: Tabela elaborada pelo autor.

Figura 14 - Distribuição empírica de requisições por IP.



FONTE: Figura elaborada pelo autor.

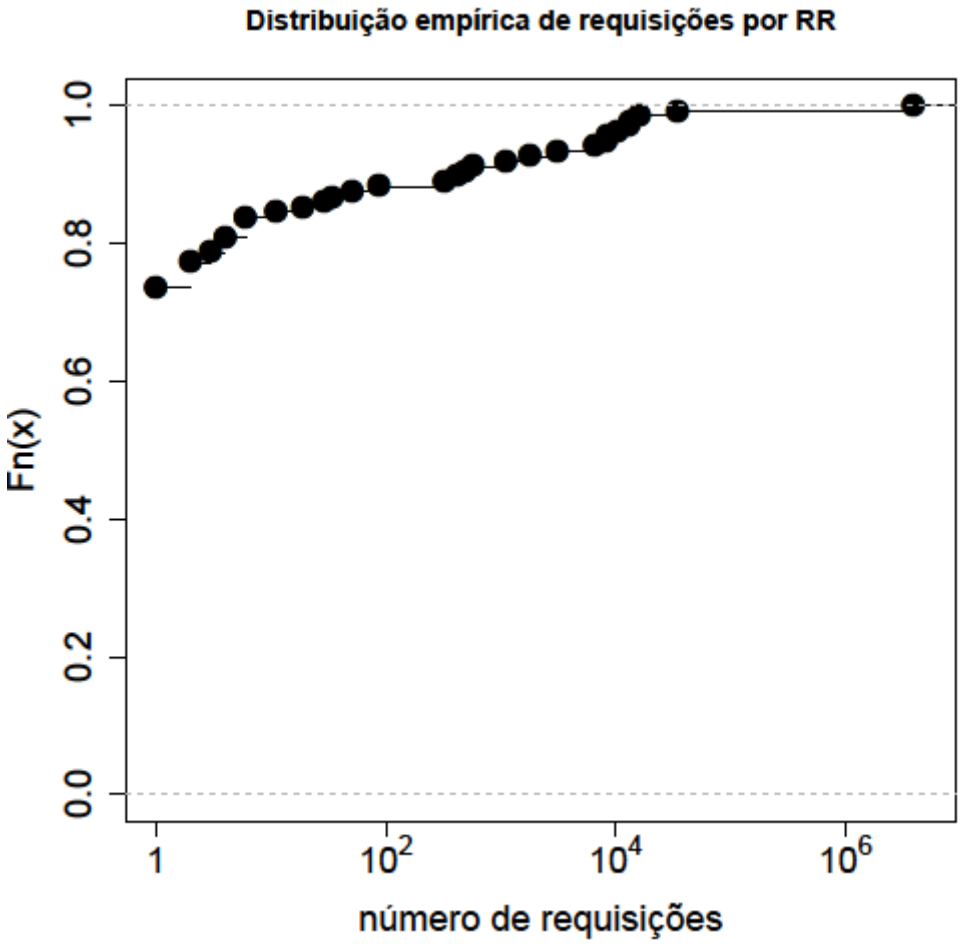
Se considerar um domínio como sendo o QNAME da seção de pergunta de uma requisição DNS, o DNSpot recebeu consultas para 129 domínios distintos. Se for considerada a definição de um RR, ou seja, levando em conta o QNAME, QTYPE e QCLASS, o DNSpot observou 136 RRs diferentes nas consultas recebidas, pois alguns nomes foram consultados usando mais de um QTYPE. A Tabela 14 resume a distribuição das consultas observadas; o número total de consultas é inferior ao encontrado na Tabela 5, pois houve 9 requisições malformadas que não continham nenhum RR na seção de pergunta. Um único RR (hehehey.ru ANY) responde por 97,05% das consultas, e as 10 consultas mais populares abrangem 99,87% do total. Como pode ser visto na Figura 15 (na qual o eixo x tem escala logarítmica), a distribuição empírica do número total de consultas por RR possui assimetria positiva. O número de consultas por RR variou entre 1 e 3.916.398, sendo a média 29.673,5 e o desvio padrão 335.774,9, conforme a Tabela 15. Pouco mais de 70% dos RRs tiveram uma única consulta, 95% dos RRs tiveram menos de 8.318 requisições (0,2% do máximo observado), e apenas dois RRs tiveram número de consultas acima do 99º percentil.

Tabela 14 – Distribuição das consultas observadas pelo DNSpot

RR	Nº de consultas	%
hehehey.ru ANY	3.916.398	97,05
mototrazit.ru ANY	34.714	0,86
vp47.ru ANY	16.141	0,40
l3x.ru ANY	13.455	0,33
. ANY	12.984	0,32
3858 ANY	10.387	0,26
gransy.com ANY	8.466	0,21
vp47.ru A	8.268	0,20
6z2.ru TXT	6.569	0,16
lifemotodrive.ru ANY	3.128	0,08
outros	5.086	0,13
Total	4.035.596	100,0

FONTE: Tabela elaborada pelo autor.

Figura 15 - Distribuição empírica de requisições por RR.



FONTE: Figura elaborada pelo autor.

Tabela 15 - Estatísticas de número de transações por RR.

Estatísticas	Número de consultas
Média	29.673,5
Desvio padrão	335.774,9
Mínimo	1
1º quartil	1
Mediana	1
3º quartil	2
95º percentil	8.317,5
99º percentil	28.213,5
Máximo	3.916.398

FONTE: Tabela elaborada pelo autor.

A Tabela 16 mostra o número de consultas por QTYPE. Observa-se que apenas três tipos distintos (A, TXT e ANY) foram usados, com amplo predomínio de ANY (99,2% do total), como já era possível inferir da Tabela 14.

Tabela 16 – Tipos (QTYPE) usados nas consultas

QTYPE	Nº de consultas	%
ANY	4.006.054	99,2
A	22.953	0,6
TXT	6.589	0,2
Total	4.035.596	100,0

FONTE: Tabela elaborada pelo autor.

A Tabela 17 apresenta o tamanho (em bytes) de uma consulta e resposta padrão para os 10 RRs mais populares consultados no DNSpot, bem como o respectivo fator de amplificação (razão entre os tamanhos da resposta e da consulta). Foram utilizadas as ferramentas dig[13] (*Domain Information Groper*), RawCap[14] e Wireshark[15] para determinar o tamanho do *payload* dos datagramas UDP das consultas e respostas dos domínios, com EDNS(0) habilitado. Em média, cada consulta de 37,5 bytes resultou em uma resposta de 3.637 bytes, com um fator de amplificação de 96,3, o que mostra a eficácia do uso do DNS para ataques de DoS; o máximo fator de amplificação observado foi de 110,7.

Na Tabela 17 não foram considerados os RRs 3858 ANY e 6z2.ru TXT (presentes na Tabela 14) porque eles não produzem amplificação: o RR 3858 ANY é inexistente, enquanto que 6z2.ru TXT retorna uma resposta vazia com *NoError*. Embora o primeiro caso seja um erro indiscutível (pois nunca existiu o domínio de primeiro nível “3858”), o segundo pode ser um engano (embora o QTYPE TXT retorne uma resposta vazia, uma consulta por 6z2.ru ANY gera uma resposta de 4.071 bytes) ou um registro que deixou de existir no DNS mas ainda é referenciado por uma ferramenta de ataque.

[13] ftp://ftp.isc.org/isc/bind9/cur/9.10/doc/arm/man.dig.html
[14] http://www.netresec.com/?page=RawCap
[15] https://www.wireshark.org/

Tabela 17 - Tamanho de consulta e resposta e fator de amplificação para os 10 RRs mais populares.

RR	Consulta (bytes)	Resposta (bytes)	Fator de amplificação
hehehey.ru ANY	39	3850	98,7
mototrazit.ru ANY	42	3853	91,7
vp47.ru ANY	36	3979	110,5
l3x.ru A	35	3875	110,7
. ANY	28	1503	53,7
gransy.com ANY	39	3591	92,1
vp47.ru A	36	3892	108,1
lifemotodrive.ru ANY	45	3969	88,2
nhl.msk.su ANY	39	3965	101,7
oi69.ru A	36	3892	108,1
Média	37,5	3637	96,3

FONTE: Tabela elaborada pelo autor.

Cruzando a quantidade de consultas recebidas para cada RR (Tabela 14) com os tamanhos das consultas e respostas da Tabela 17, pode-se inferir o tráfego de resposta esperado por parte dos atacantes, conforme apresentado na Tabela 18. Constata-se que apenas esses 10 RRs mais comuns poderiam, na ausência de um mecanismo contra ataques DoS, ter gerado um volume de tráfego superior a 15 GB.

Tabela 18 - Tráfego esperado de resposta para os 10 RRs mais consultados.

RR	Tráfego de consulta (MB)	Tráfego esperado de resposta (MB)
hehehey.ru. ANY	152,74	15.078,13
mototrazit.ru. ANY	1,46	133,75
vp47.ru. ANY	0,58	64,23
l3x.ru. A	0,47	52,14
. ANY	0,36	19,51
gransy.com. ANY	0,33	30,40
vp47.ru. A	0,30	32,18
lifemotodrive.ru. ANY	0,14	12,41
nhl.msk.su. ANY	0,07	7,11
oi69.ru. A	0,04	4,31
Total	156,49	15.434,17

FONTE: Tabela elaborada pelo autor.

5.2.5 Ataques DoS

Durante o período que esteve em produção o DNSpot sofreu uma série de requisições cujo padrão permite inferir que elas fazem parte de ataques de negação de serviço por amplificação. Para poder analisar esses ataques, é necessário definir como agrupar as várias requisições que compõem um mesmo ataque DoS a partir das características observadas. Como não foi encontrada na literatura nenhuma definição que pudesse ser usada, decidiu-se, com base em uma análise preliminar do tráfego do DNSpot, adotar a seguinte definição:

> Um ataque DoS é formado por um conjunto com no mínimo 5 consultas com o mesmo IP de origem e com espaçamento máximo de 60 segundos entre consultas consecutivas, e pelas respostas a essas consultas.

Embora 5 consultas representem no máximo 20 KB de tráfego, esse número mínimo de consultas foi estabelecido considerando que o DNSpot esteja sendo usado como um dos vários amplificadores envolvidos em um ataque DDoS. Ainda, de acordo com essa definição, o domínio pode variar durante um ataque, embora essa variação não tenha sido expressiva nas interações com o pelo DNSpot. Foram observados apenas 54 ataques (0,7% do total) envolvendo mais de um domínio, sendo 51 deles com dois domínios, e os três ataques restantes abrangendo 9, 30 e 32 domínios.

De acordo com a definição proposta, foi observado que o DNSpot sofreu cerca de 7.940 ataques DoS distintos. O primeiro ataque foi observado às 13:34 do dia 11/09/2015, menos de 28 h após início da operação do DNSpot. Além disso, pode-se também observar que 3.499 endereços IP únicos estavam ligados a ataques DoS (nesse caso, prováveis vítimas), e 87 RRs diferentes foram usados nas consultas. Por último, o número de transações DNS envolvidas em ataques DoS foram de 4.032.778, ou 99,9% do total observado. A Tabela 19 mostra a porcentagem dessas métricas se comparado aos totais computados no DNSpot.

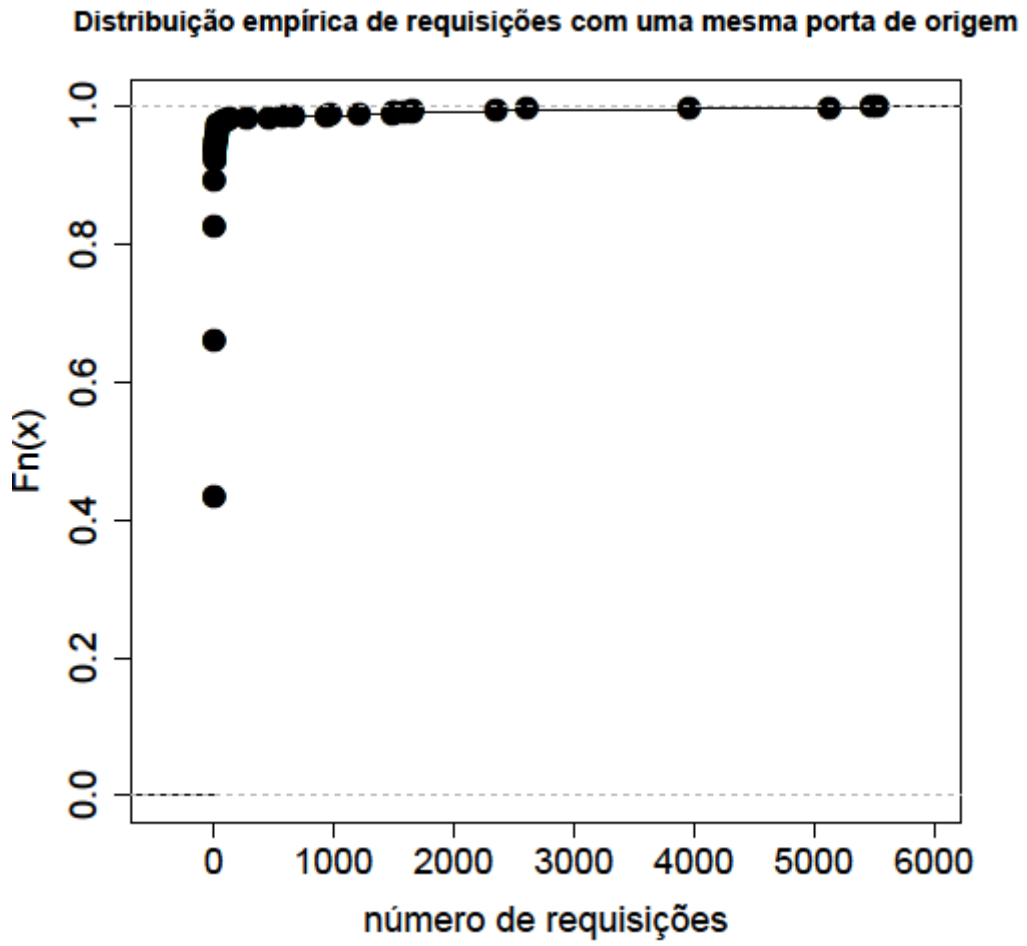
Tabela 19 – Porcentagem do envolvimento em DoS de métricas comparadas aos totais computados no DNSpot.

Métrica	Envolvidos em DoS	Total	% do total
IPs envolvidos	3.499	4.287	81,6%
RRs envolvidos	87	136	64,0%
Numero de consultas	4.032.778	4.035.605	99,9%

FONTE: Tabela elaborada pelo autor.

Dentre os ataques observados que atendem à definição, 1090 deles (13,7%) mantiveram a mesma porta de origem nas consultas durante toda a duração do ataque. Como clientes DNS legítimos tendem a seguir a recomendação de usar portas de origem aleatórias (HUBERT, VAN MOOK; 2009), o uso de uma porta fixa evidencia o uso de uma ferramenta automatizada de ataque. A distribuição empírica do número de consultas associadas a ataques e que usam porta de origem fixa é mostrada na Figura 16. Embora a maioria dos ataques envolva poucas consultas – 1.011 dos 1.090 ataques (93%) tiveram até 10 consultas – três ataques tiveram mais de 5.000 consultas usando a mesma porta de origem. Esses dados demonstram uma ampla tendência de que a porta de origem seja trocada ao longo de um ataque, provavelmente para contornar mecanismos de proteção como *firewalls* e limitação da taxa de consultas (*rate limiting*) no servidor DNS.

Figura 16 - Distribuição empírica de requisições com uma mesma porta de origem associadas a ataques DoS.



. FONTE: Figura elaborada pelo autor.

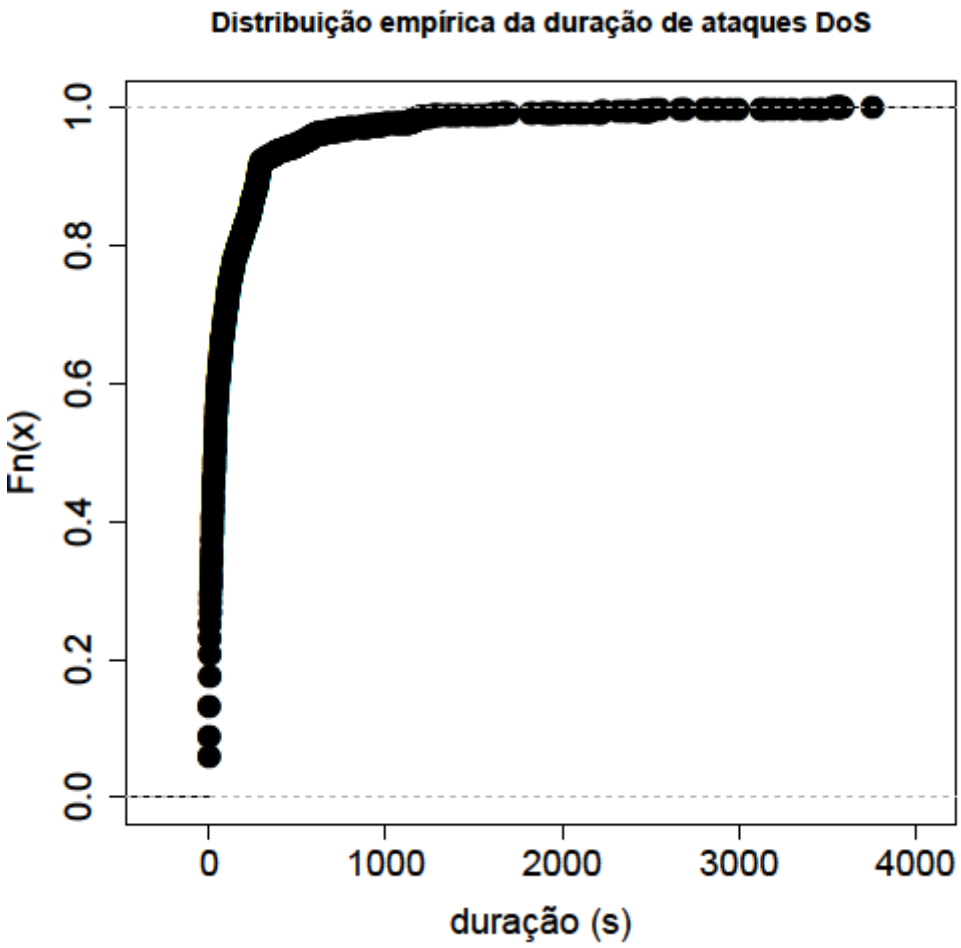
A duração média dos ataques DoS sofridos foi de 131,4 s, com desvio padrão de 323,2 s; a duração mínima foi inferior a 1 s (não é possível precisar mais porque a resolução dos *timestamps* é de 1 s), e a máxima 3.757 s, conforme a Tabela 20. A distribuição empírica da duração dos ataques, que possui assimetria positiva, pode ser observada na Figura 17. A maioria dos ataques teve curta duração: 75% deles duraram pouco mais de 2 minutos (122 s), e 95% menos de 9 minutos (538 s).

Tabela 20 - Estatísticas da duração de ataques DoS.

Estatísticas	Duração (s)
Média	131,4
Desvio padrão	323,2
Mínimo	< 1
1º quartil	6
Mediana	33
3º quartil	122
95º percentil	538
99º percentil	1.648,1
Máximo	3.757

FONTE: Tabela elaborada pelo autor.

Figura 17 - Distribuição empírica da duração de ataques DoS.



. FONTE: Figura elaborada pelo autor.

Foram observadas entre 5 e 25.363 requisições por ataque, com média de 507,9 e desvio padrão de 1.340, conforme a Tabela 21. A Figura 18 mostra a distribuição empírica das requisições por ataque; em (a) o eixo x é linear, o que permite visualizar que alguns poucos ataques tiveram grandes números de requisições, enquanto que em (b) o eixo x é logarítmico, o que dá uma noção melhor da distribuição para a maioria de ataques com poucas requisições. Percebe-se que 75% dos ataques tiveram até 402 requisições e 88,5% tiveram 1.000 requisições ou menos; o 99º percentil atinge apenas 26,2% do número máximo de requisições observadas.

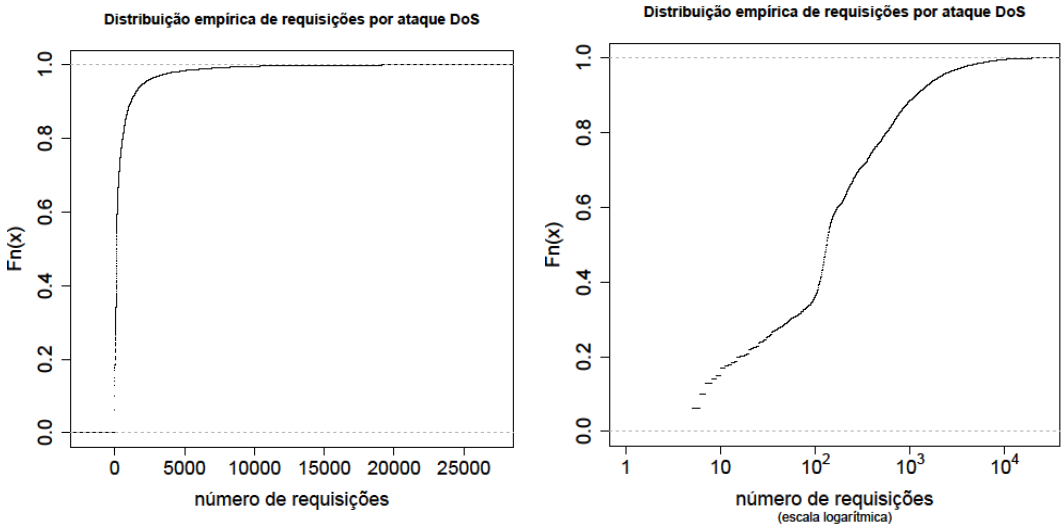
Tabela 21 – Estatistícas de requisições por ataque DoS.

Estatísticas	Número de requisições
Média	507,9
Desvio padrão	1.340
Mínimo	5
1º quartil	30
Mediana	132
3º quartil	402
95º percentil	2.100,2
99º percentil	6.638,9
Máximo	25.363

FONTE: Tabela elaborada pelo autor.

A Tabela 22 apresenta estatísticas do número de ataques DoS por endereço IP, e a Figura 19 mostra a respectiva distribuição empírica. A média de envolvimento em ataques foi de 1,85 por IP, sendo 0 o mínimo e 73 o máximo, e o desvio padrão 3,27. A maioria dos IPs esteve envolvida em poucos ataques: a metade não passou de um único ataque, e aproximadamente 80% tiveram envolvimento em até 3 ataques. Somente 5% e 1% dos clientes estiveram envolvidos em mais de 6 e 14 ataques, respectivamente. Isso mostra que os ataques foram razoavelmente bem distribuídos entre os IPs que interagiram com o DNSpot.

Figura 18 – Distribuição empírica de requisições por ataque DoS. (a) Eixo x linear. (b) Eixo x logarítmico.



(a) (b)

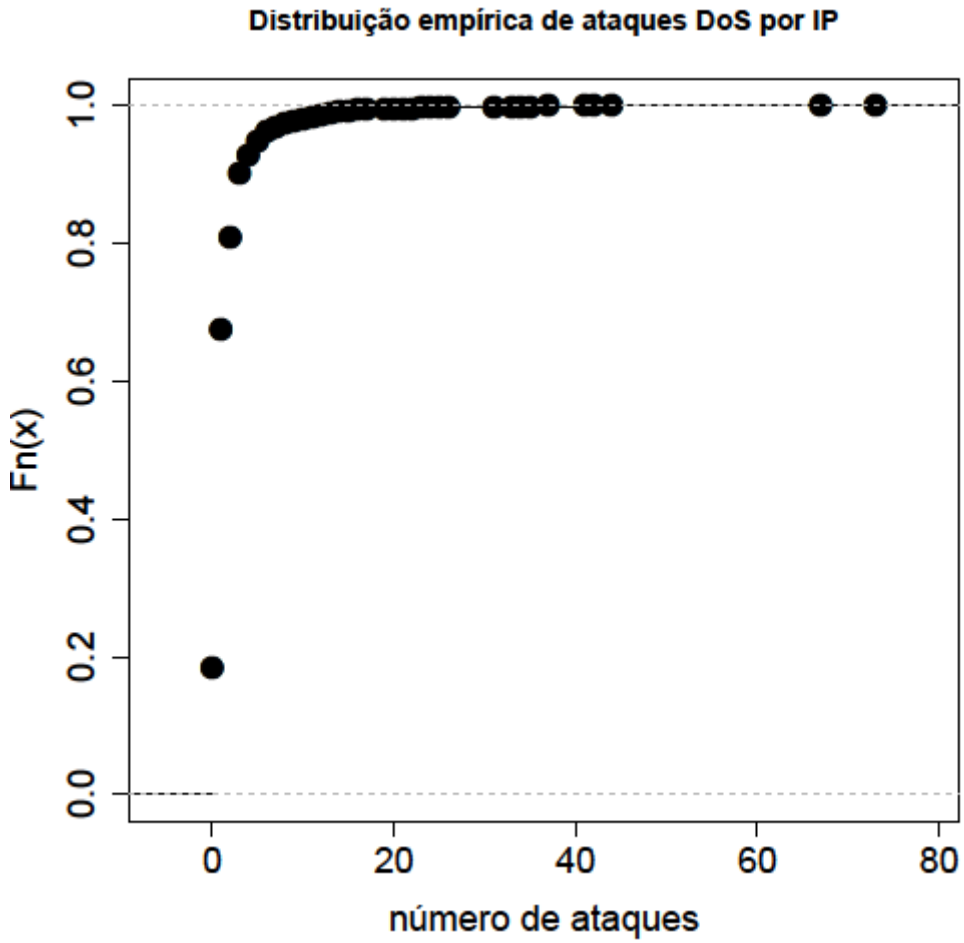
FONTE: Imagens elaboradas pelo autor.

Tabela 22 - Estatistícas de número de ataques DoS por IP.

Estatísticas	Quantidade de ataques DoS
Média	1,852
Desvio padrão	3,271
Mínimo	0
1º quartil	1
Mediana	1
3º quartil	2
95º percentil	6
99º percentil	14
Máximo	73

FONTE: Tabela elaborada pelo autor.

Figura 19 - Distribuição empírica de ataques DoS por IP.



FONTE: Imagem elaborada pelo autor.

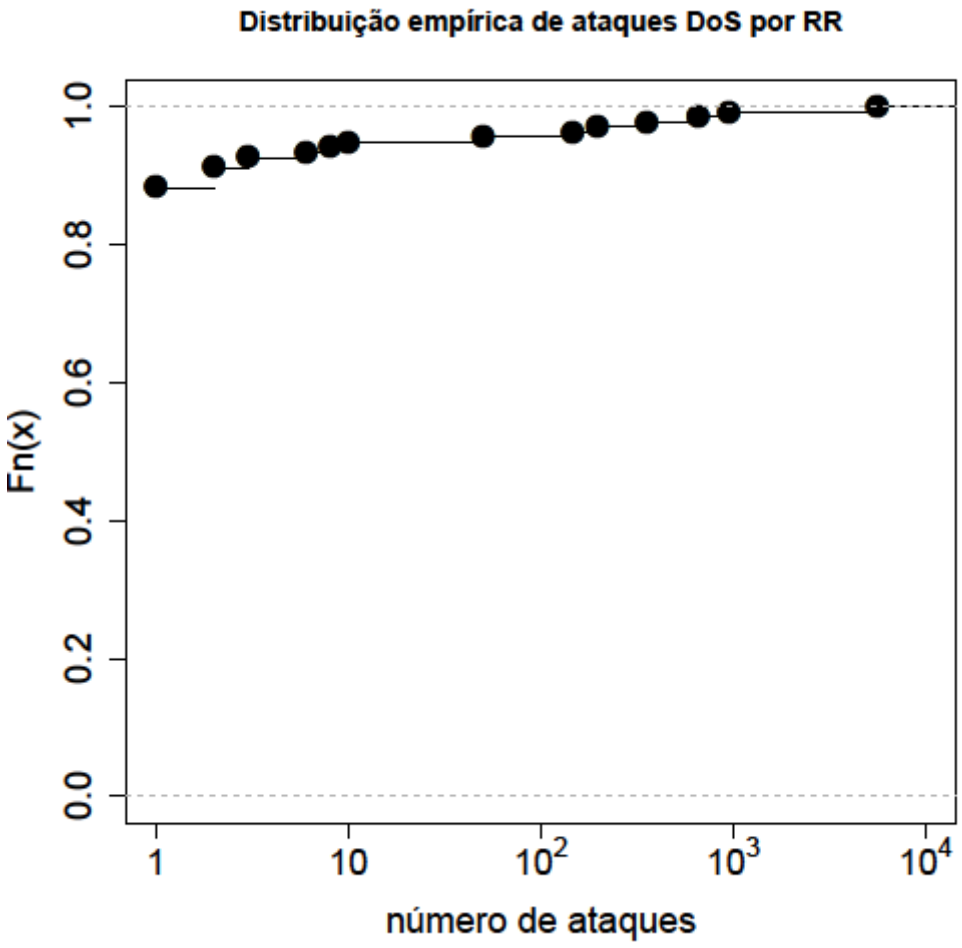
Com relação ao número de ataques DoS em que cada RR foi usado, a Tabela 23 apresenta alguns dados estatísticos, e a Figura 20 apresenta a respectiva distribuição empírica. O número de participação de RR em ataques variou de 0 a 5.594, com média de 59,3 e desvio padrão de 489,5. A distribuição é assimétrica positiva, com um número muito pequeno de RRs sendo responsáveis pela gigantesca maioria dos ataques de DoS. Enquanto 95% dos RRs observados foram usados em no máximo 20 ataques, o RR hehehey.ru. ANY sozinho foi usado em 5.594 ataques, ou 70% do total.

Tabela 23 - Estatistícas de número de ataques DoS por RR.

Estatísticas	Quantidade de ataques DoS
Média	59,3
Desvio padrão	489,5
Mínimo	0
1º quartil	0
Mediana	1
3º quartil	1
95º percentil	20
99º percentil	848,1
Máximo	5.594

FONTE: Tabela elaborada pelo autor.

Figura 20 - Distribuição empírica de ataques DoS por RR.



FONTE: Imagem elaborada pelo autor.

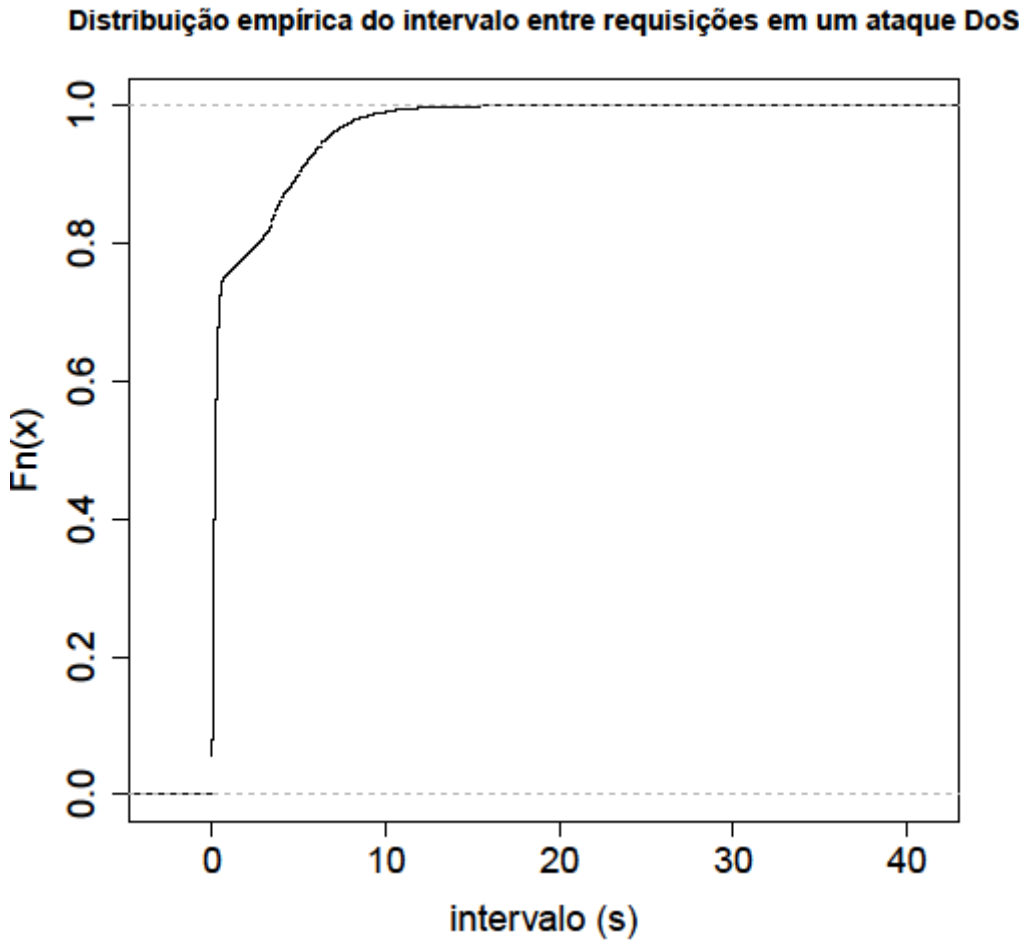
Dados estatísticos sobre o intervalo entre as requisições DNS durante um ataque DoS podem ser observados na Tabela 24, e a distribuição empírica pode ser vista na Figura 21. O intervalo médio entre as requisições foi de 1,3 s, sendo que o mínimo foi inferior a 1 s e o máximo foi 38,2 s, com um desvio padrão de 2,4 s. Devido à mediana de 0,184 s e o 3º quartil de 0,673 s serem menores que a média, pode-se concluir que a distribuição é assimétrica positiva. O intervalo foi inferior a 700 ms para 75% dos ataques, o que demonstra que os ataques se caracterizam mais por enviar requisições em rápida sucessão.

Tabela 24 - Estatistícas do intervalo entre requisições em um ataque DoS.

Estatísticas	Intervalo (s)
Média	1,3
Desvio padrão	2,4
Mínimo	< 1
1º quartil	0,051
Mediana	0,184
3º quartil	0,673
95º percentil	6,5
99º percentil	9,8
Máximo	38,2

FONTE: Tabela elaborada pelo autor.

Figura 21 - Distribuição empírica do intervalo médio entre requisições em um ataque DoS.



FONTE: Imagem elaborada pelo autor.

Diante dessa constatação, decidiu-se caracterizar também as rajadas de requisições DNS que foram realizadas durante os ataques DoS. A definição de rajada adotada nesta caracterização foi a seguinte:

> Uma rajada é formada por um conjunto com no mínimo 5 requisições com o mesmo endereço IP e porta de origem, e com espaçamento máximo de 5 segundos entre requisições consecutivas.

Observa-se que, em adição ao critério de intervalo, uma rajada é interrompida quando houver uma consulta para o mesmo IP mas com outra porta de origem. O número mínimo de requisições e o intervalo máximo entre requisições consecutivas

com as mesmas características foram determinados com base em uma análise empírica do tráfego referente a DoS observado.

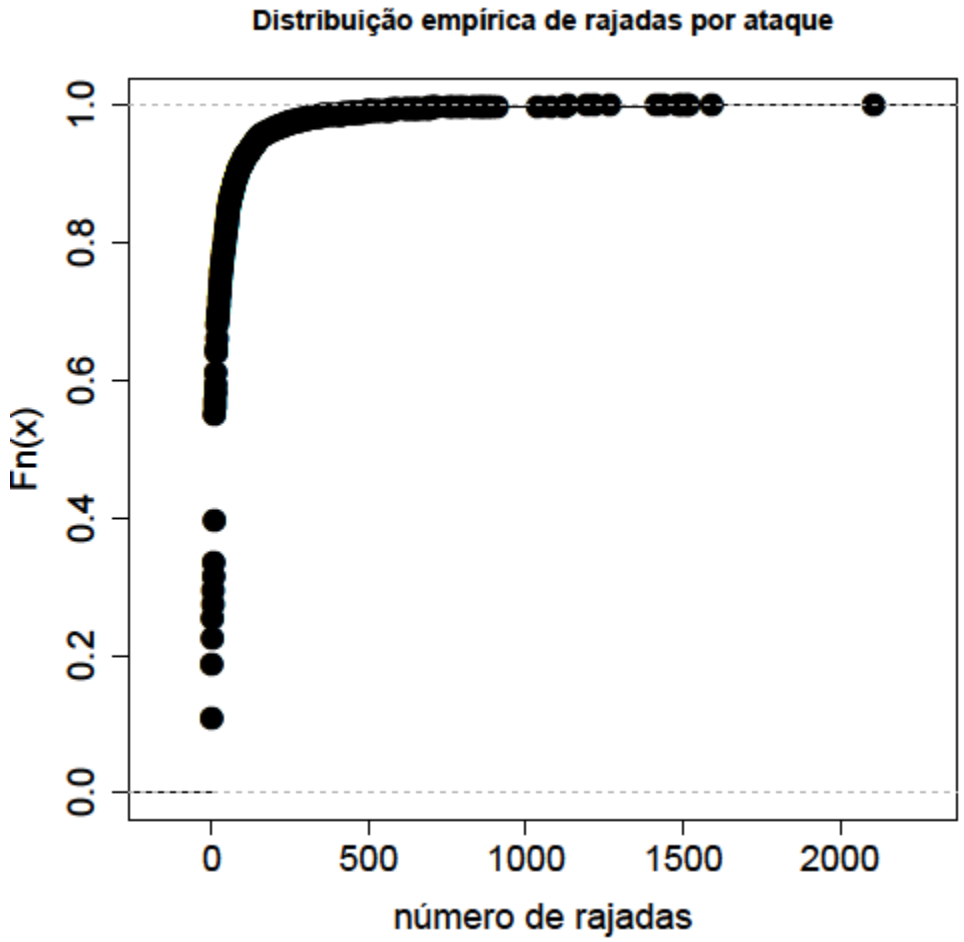
A Tabela 25 apresenta estatísticas de rajadas por ataque DoS, enquanto que a Figura 22 apresenta a distribuição empírica correspondente. A média de rajadas por ataque DoS foi de 37,1 e o desvio padrão foi de 99,2, com o mínimo de zero e o máximo de 2.107 rajadas. Três quartos dos ataques DoS tiveram até 29,2 rajadas e menos de 5% tiveram mais que 148 rajadas (7% do máximo observado), o que evidencia a natureza assimétrica positiva da distribuição das rajadas.

Tabela 25 – Estatísticas das rajadas por ataque Dos.

Estatísticas	Quantidade de rajadas
Média	37,1
Desvio padrão	99,2
Mínimo	0
1º quartil	3
Mediana	9
3º quartil	29,2
95º percentil	148
99º percentil	494,2
Máximo	2.107

FONTE: Tabela elaborada pelo autor.

Figura 22 - Distribuição empírica de rajadas por ataque Dos.

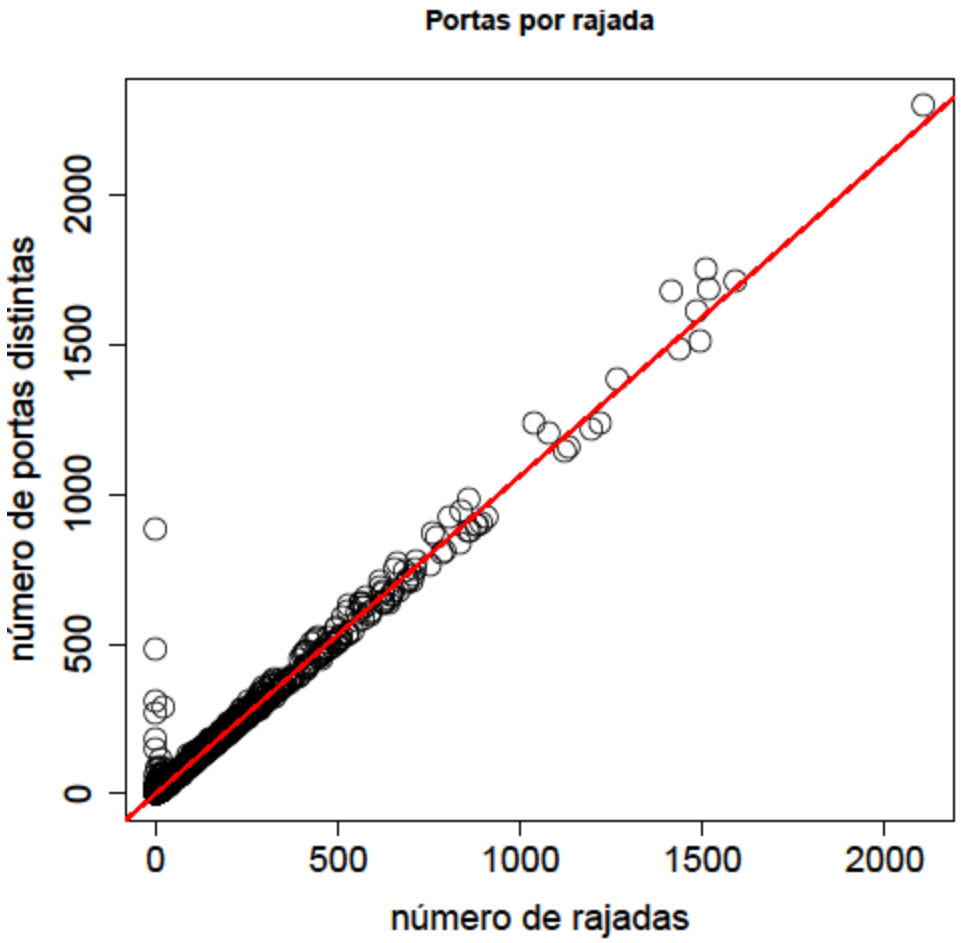


FONTE: Imagem elaborada pelo autor.

A Figura 23 relaciona a quantidade de portas distintas usadas em ataques DoS com o número de rajadas dos ataques; os pontos representam os dados observados, e a reta a equação de regressão linear correspondente. Percebe-se uma forte correlação linear positiva entre as variáveis, ou seja, com o aumento do número de rajadas aumenta o número de portas distintas utilizadas no ataque. Isso indica que a maioria dos ataques DoS utilizaram portas distintas em cada rajada, pois o número de portas distintas aumenta com o número de rajadas, basicamente acompanhando a reta de regressão. Os pontos acima da reta, especialmente aqueles mais à esquerda no gráfico, correspondem a ataques em que o mesmo número de porta foi usado em menos de 5 requisições consecutivas de um dado IP,

ou o intervalo entre requisições consecutivas foi superior aos 5 segundos estabelecidos na definição de rajada.

Figura 23 - Relação entre o número de portas distintas usadas em ataques DoS e o número de rajadas por ataque.



FONTE: Imagem elaborada pelo autor.

O número de consultas por rajada variou entre 5 e 5.176, com uma média de 13,5 consultas por rajada e desvio padrão de 20,8; a Tabela 26 traz um sumário das estatísticas. A distribuição é razoavelmente simétrica (a mediana é próxima da média), embora tenha havido observações destoantes – menos de 1% das rajadas tiveram mais de 25 consultas associadas (quantidade inferior a 0,5% do máximo observado). Isso permite concluir que o padrão observado foi de poucas consultas por rajada, e que rajadas com um número grande de consultas foram exceções.

Tabela 26 - Estatísticas de consultas por rajada.

Estatísticas	Quantidade de consultas
Média	13,5
Desvio padrão	20,8
Mínimo	5
1º quartil	9
Mediana	13
3º quartil	17
95º percentil	22
99º percentil	25
Máximo	5.176

FONTE: Tabela elaborada pelo autor.

Embora tenha-se adotado um limiar de 5 segundos entre requisições consecutivas para a caracterização de uma rajada, 99% das rajadas tiveram duração de 1 segundo ou menos, como pode ser observado na Tabela 27. A duração mínima foi um valor inferior a 1 segundo e a máxima foi de 298 segundos. Devido à resolução dos *timestamps* ser de 1 s, durações inferiores a esse valor são computadas como zero, o que influencia a média e o desvio padrão; por isso, a Tabela 27 não apresenta essas estatísticas. As estatísticas também demonstram que rajadas com duração alta foram exceções, como a rajada de 298 segundos. Cruzando os dados da Tabela 26 e da Tabela 27, pode-se concluir que, de modo geral, as rajadas se caracterizam por serem curtas e com poucas consultas, com intervalo de no máximo 1 segundo.

Tabela 27 - Estatísticas da duração das rajadas.

Estatísticas	Duração em segundos
Mínimo	<1
1º quartil	<1
Mediana	<1
3º quartil	<1
95º percentil	<1
99º percentil	1
Máximo	298

FONTE: Tabela elaborada pelo autor.

5.2.6 Discussão dos resultados

Uma análise dos dados estatísticos coletados pelo DNSpot, considerando de forma conjunta as diferentes métricas observadas, permite chegar a algumas conclusões importantes:

- O principal abuso envolvendo servidores DNS recursivos abertos é para ataques DDoS baseados em amplificação. A escolha adequada dos nomes consultados permite obter um fator de amplificação próximo a 100, consideravelmente maior que os fatores típicos reportados para o DNS, entre 28 e 54 (CERT.br, 2015). Nesse sentido, o mecanismo de limitação diária de consultas, embora pouco sofisticado, foi eficaz em restringir o tráfego de ataque obtido com o abuso do DNSpot a menos de 10% do volume pretendido.
- O volume de requisições maliciosas, especialmente considerando o fato de ser um servidor não anunciado publicamente, é significativo: o DNSpot recebeu em média uma requisição por segundo, com um tráfego médio diário potencial de 298 MB. O DNSpot começou a receber as primeiras requisições segundos após o início do seu funcionamento, e menos de 28h depois já estava sendo usado em ataques DDoS.
- Embora o volume total de tráfego seja significativo, os ataques são em geral de curta duração (93,3% duram até 5 minutos), envolvem poucas requisições (75% dos ataques têm até 402 requisições), e poucos *hosts* são atacados múltiplas vezes (apenas 5% dos endereços IP estiveram relacionados a mais de 6 ataques). Isso sugere que a estratégia de ataque é que um dado refletor envie uma quantidade relativamente pequena de tráfego para diversos alvos, e não uma quantidade massiva de tráfego para um ou poucos alvos. Essa estratégia minimiza as chances de detecção da atividade maliciosa e o impacto da eventual perda de refletores durante um ataque. Em contrapartida, requer o uso concomitante de um grande número de refletores para gerar tráfego suficiente para causar uma negação de serviço.

- A assimetria dos dados observados complica a tarefa de obter modelos representativos das características do tráfego DNS que chega até o DNSpot. Tais modelos poderiam ser úteis para entender os aspectos essenciais desse tipo de tráfego, e contribuir no projeto e avaliação de mecanismos para mitigar ataques contra o DNS.

Dada a sua característica de armazenar dados detalhados sobre as requisições DNS processadas, o DNSpot permite uma análise mais aprofundada, indo além da descrição estatística. Na Seção 5.3 são discutidas algumas das anomalias encontradas no tráfego processado pelo DNSpot; embora não tenha sido identificado nenhum ataque direto contra o servidor DNS, essa análise ajuda a compreender melhor como os atacantes interagem com o DNS, o que afinal de contas é o propósito de um *honeypot* de pesquisa.

5.3 ANOMALIAS DE TRÁFEGO

O DNSpot enquanto esteve em produção recebeu diversas mensagens DNS. Algumas dessas consultas recebidas apresentaram anomalias que foram detectadas durante a análise dos resultados. Essas anomalias incluem mensagens anômalas mal-formatadas que não seguem a formatação DNS e mensagens de protocolo SIP enviadas ao DNSpot; multiplos domínios diferentes que referenciam os mesmos registros nas respostas, e que parecem ter sido projetados para serem usados em ataques DoS; e um domínio que durante o ciclo de vida do DNSpot simplesmente deixou de existir, passando a respondar com erro.

5.3.1 Mensagens anômalas

Durante o período de monitoração, foram observadas 9 mensagens anômalas, ou seja, que não puderam ser interpretadas por não respeitarem o formato definido nas especificações do DNS (Seção 2.4). A Tabela 28 sumariza essas mensagens anômalas. Uma mensagem, formatada incorretamente, tinha como conteúdo “wwwKcpscgov\n”, que sugere o nome de domínio “www.cpsc.gov”;

o domínio cpsc.gov (sem “www.”) aparece em 490 consultas válidas. Duas mensagens com conteúdo vazio foram classificadas como varredura de portas UDP, como as realizadas pela ferramenta Nmap (NMAP, 2015). As seis mensagens restantes eram mensagens do protocolo SIP (*Session Initiation Protocol*), usado em telefonia IP (CAMARILLO *et al*, 2002; HANDLEY *et al*, 1999), e foram categorizadas como varredura de SIP; o conteúdo sugere que essas mensagens foram geradas pela ferramenta SIPvicious (GAUCI, 2007), que realiza esse tipo de varredura. A classificação foi baseadas na inspeção manual do conteúdo das mensagens que não puderam ser interpretadas pela DNSLib, que fica armazenado na tabela DNS_RECV_RAWDATA justamente para permitir esse tipo de análise. Mesmo anômalas, as mensagens foram repassadas ao Unbound para processamento, e a tabela mostra o RCODE retornado pelo servidor DNS; o RCODE devolvido para mensagens SIP dependia do tipo específico de mensagem.

Tabela 28 - Mensagens Anômalas observadas pelo DNSpot.

Tipo de mensagem	Quantidade	RCODE
Malformada	1	*FormErr*
Varredura UDP	2	*ServFail*
Varredura SIP	6	*FormErr, NotImp*

FONTE: Tabela elaborada pelo autor.

5.3.2 Domínios projetados para DoS

Os RRs consultados por clientes apresentaram certas características que podem ser identificadas como anomalias. Dentre os 10 RRs mais requisitados apresentados na Tabela 17, alguns deles apresentaram mensagens DNS de resposta basicamente idênticas. Os RRs hehehey.ru., mototrazit.ru., lifemotodrive.ru. e nhl.msk.su. possuem uma resposta DNS muito parecida e sempre com cerca de 40-41 RRs de respostas seguindo o mesmo padrão, exemplificado na Figura 24:

- Quatro registros NS apontando para os mesmos servidores de nomes;
- Registro SOA indicando a mesma autoridade;
- Dois registros MX apontando para os mesmos servidores de *email*;
- Um único registro A para um endereço IP 77.222.56.X;

- Vários registros TXT cujo prefixo é o caractere ‘x’ repetido 99 vezes, seguido de um sufixo numérico que varia de 1 até 33, para tornar cada RR único e distinto. Esses registros não possuem nenhum significado ou utilidade para uma consulta normal DNS, sendo simplesmente usados para aumentar o tamanho das respostas associadas a esses RRs.

Figura 24 - Resposta à consulta dig por hehehey.ru IN ANY.

```
C:dig @127.0.0.1 +edns=0 hehehey.ru. ANY

; <<>> DiG 9.10.3 <<>> @127.0.0.1 +edns=0 hehehey.ru. ANY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11143
;; flags: qr rd ra; QUERY: 1, ANSWER: 40, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;hehehey.ru.                    IN      ANY

;; ANSWER SECTION:
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx9"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx10"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx11"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx12"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx13"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx14"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx15"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx16"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx17"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx18"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx19"
hehehey.ru.             600     IN      TXT     "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx20"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx21"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx23"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx24"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx25"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx26"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx27"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx28"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx29"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx30"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx31"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx32"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx33"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx1"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx2"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx3"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx4"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx5"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx6"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx7"
hehehey.ru.          600    IN     TXT    "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx8"
hehehey.ru.          600    IN     SOA    ns1.spaceweb.ru. dns1.sweb.ru. 2
015072656 28800 7200 604800 600
hehehey.ru.          600    IN     A      77.222.56.130
hehehey.ru.          600    IN     NS     ns1.spaceweb.ru.
hehehey.ru.          600    IN     NS     ns4.spaceweb.pro.
hehehey.ru.          600    IN     NS     ns3.spaceweb.pro.
hehehey.ru.          600    IN     NS     ns2.spaceweb.ru.
hehehey.ru.          600    IN     MX     20 mx2.spaceweb.ru.
```

```
hehehey.ru.          600    IN     MX     10 mx1.spaceweb.ru.

;; Query time: 646 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Nov 08 03:27:23 E. South America Daylight Time 2015
;; MSG SIZE  rcvd: 3850
```

FONTE: Figura elaborada pelo autor.

Um outro conjunto de RRs semelhantes são vp47.ru, oi69.ru, l3x.ru e 6z2.ru (esse último caso fosse usado ANY). Eles geram cerca de 244-250 RRs na seção de resposta no formato apresentado na Figura 25 (por questão de tamanho, 234 registros A foram omitidos) e definido abaixo:

- Dois registros NS apontando para os mesmos servidores de nomes;
- Registro SOA indicando a mesma autoridade;
- Vários registros A para endereços IP distintos, sendo que alguns desses IPs são usados em dois ou mais dos nomes citados, enquanto outros são exclusivos.

Figura 25 - Resposta à consulta dig por vp47.ru. IN ANY.

```
C:\ >dig @127.0.0.1 +edns=0 vp47.ru. ANY

; <<>> DiG 9.10.3 <<>> @127.0.0.1 +edns=0 vp47.ru. ANY
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 40447
;; flags: qr rd ra; QUERY: 1, ANSWER: 244, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;vp47.ru.                IN      ANY

;; ANSWER SECTION:
vp47.ru.          20889   IN      A       31.31.204.59
vp47.ru.          20889   IN      A       85.97.130.229
vp47.ru.          20889   IN      A       85.97.66.84
vp47.ru.          20889   IN      A       85.97.190.213
vp47.ru.          20889   IN      A       85.98.141.65
vp47.ru.          20889   IN      A       85.97.7.40
```

```
(.....................234 RRs omitidos.....................)
vp47.ru.            20889   IN      A       85.98.208.75
vp47.ru.            20889   IN      NS      ns1.reg.ru.
vp47.ru.            20889   IN      NS      ns2.reg.ru.
vp47.ru.            20889   IN      SOA     ns1.reg.ru. hostmaster.ns1.reg.r
u. 1444560077 14400 3600 604800 10800

;; Query time: 5 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Nov 08 03:31:51 E. South America Daylight Time 2015
;; MSG SIZE  rcvd: 3979
```

FONTE: Figura elaborada pelo autor.

A conclusão lógica é que esses grupos de domínios distintos com estruturas idênticas são projetados para serem usados especificamente em ataques de DoS, uma vez que produzem respostas de quase 4 KB (limite usual de EDNS(0)) sem um propósito legítimo aparente.

5.3.3 Desaparecimento de domínio

Uma anomalia verificada foi com relação ao domínio l3x.ru. Inicialmente o DNSpot recebeu ataques de DoS referenciando esse domínio, cujas respostas eram válidas e não vazias, ou seja, RCODE=*NoError* e com 250 RRs na seção de resposta. Porém, em algum momento entre 21/10/2015 e 23/10/2015, esse domínio deixou de existir, pois a partir da segunda data todas as consultas a ele passaram a responder com *NxDomain*. Isso é comprovado realizando uma consulta ao domínio utilizando o dig, conforme Figura 26.

Figura 26 - Resposta à consulta dig por l3x.ru. IN A.

```
C:\>dig @127.0.0.1 +edns=0 l3x.ru. A

; <<>> DiG 9.10.3 <<>> @127.0.0.1 +edns=0 l3x.ru. A
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 48734
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
```

```
;; QUESTION SECTION:
;l3x.ru.                        IN      A

;; AUTHORITY SECTION:
ru.                 1785    IN      SOA     a.dns.ripn.net. hostmaster.ripn.
net. 4024299 86400 14400 2592000 3600

;; Query time: 2 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Sun Nov 08 06:08:37 E. South America Daylight Time 2015
;; MSG SIZE  rcvd: 96
```

FONTE: Figura elaborada pelo autor.

A Tabela 29 mostra o momento em que o domínio passou a ser não existente. Anteriormente ao dia 10/21/2015 todas as 2.970 consultas válidas foram respondidas com sucesso e *NoError.* A partir de 23/10/2015 o restante das 2.316 consultas válidas processadas foram respondidas com *NxDomain*. O tamanho das consultas permaneceu 35 bytes, enquanto que a resposta passou de 3.875 para 96 bytes.

Tabela 29 - Transição do RR l3x.ru. A para não existente.

RR	l3x.ru. A	l3x.ru. A
RCODE	*NoError*	*NXDomain*
Respostas válidas	2970	2316
Primeira consulta	02/10/2015	23/10/2015
Última consulta	21/10/2015	25/10/2015
Tamanho consulta (bytes)	35	35
Tamanho resposta (bytes)	3875	96

FONTE: Tabela elaborada pelo autor.

Domínios DNS não são eternos e podem deixar de existir, embora isso não seja comum. Realizando uma investigação mais aprofundada, de acordo com o WHOIS[16], o domínio foi registrado em 02/07/2015 com o *registrar*[17] REG.RU, que reporta o domínio como expirado, embora o registro esteja pago até 02/07/2016, conforme Figura 27.

[16] https://en.wikipedia.org/wiki/WHOIS
[17] https://en.wikipedia.org/wiki/Domain_name_registrar

Figura 27 – whois do domínio l3x.ru.

```
$ whois -h whois.reg.ru l3x.ru
domain:    l3x.ru
nserver:   ns1.reg.ru
nserver:   ns2.reg.ru
state:     domain expired
person:    Private Person
registrar: REGRU-RU
created:   2015.07.02
paid-till: 2016.07.02
source:    REGRU-RU
```

FONTE: Figura elaborada pelo autor.

Os servidores de autoridade indicados no WHOIS não estão respondendo pelo domínio, conforme visto na Figura 28 (a resposta do ns2.reg.ru é idêntica ao do ns1.reg.ru). Curiosamente, é retornada uma resposta vazia porém com RCODE=*NoError*. Uma hipótese (não verificável) é que o domínio tenha sido suspenso por envolvimento em atividades escusas.

Figura 28 - Resposta à consulta dig por @ns1.reg.ru l3x.ru. ns.

```
C:\>dig @ns1.reg.ru l3x.ru. ns

; <<>> DiG 9.10.3 <<>> @ns1.reg.ru l3x.ru. ns
; (5 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 35322
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;l3x.ru.                        IN      NS

;; Query time: 272 msec
;; SERVER: 194.58.117.11#53(194.58.117.11)
;; WHEN: Sun Nov 08 06:37:01 E. South America Daylight Time 2015
;; MSG SIZE  rcvd: 24
```

FONTE: Figura elaborada pelo autor.

5.4 ESTATÍSTICAS DE BANCO DE DADOS

Ao final do período em que o DNSpot ficou em produção, o banco de dados do Sqlite3 alcançou o tamanho de 6,28 GB – dividido em três arquivos com tamanhos de 2,05 GB, 1,62 GB e 2,61 GB – para o total de 4.035.605 transações recebidas. Dessa maneira, cada transação consumiu 1,63 KB de espaço no banco de dados. Esse volume de transações resultou em um total de 46.309.640 registros inseridos nas tabelas que armazenam dados de transações (e não as tabelas de estatísticas e controle do sistema). Dividindo-se esse número pelo total de transações, verifica-se que foram gerados 11,48 registros para cada transação. Considerando o tempo que o DNSpot ficou em produção (Tabela 4), foram inseridos 10,80 registros por segundo, ou 945.094,69 registros por dia.

A tabela DNS_STATISTICS armazena algumas estatísticas como o tempo médio para realizar o processo de resolução de nomes e o tempo médio para inserir os dados no banco de dados. Em média, o DNSpot levou 1,72 segundos para terminar o processo de resolução de nomes; esse é o tempo decorrido entre a chegada da requisição ao DNSpot e o envio da resposta de volta ao cliente, considerando somente as transações respondidas e não filtradas. Já o tempo médio de inserção no banco ficou em 0,141 segundos por transação; esse é o tempo decorrido desde o ponto que uma das *threads* que processa cada transação individualmente obtém acesso exclusivo ao banco (*LOCK*/MUTEX), realiza todas as consultas e inserções em todas as tabelas necessárias, e libera o acesso exclusivo ao banco para outra *thread* realizar a inserção. Além disso, foi registrado o tempo médio que cada *thread* ficou em estado de espera (*sleep*) para obter o acesso exclusivo ao banco, que foi de 66,324 segundos por transação, o que revela o alto grau de concorrência da aplicação.

O Sqlite3 apesar de ter bem menos recursos que um SGBD mais completo, como MySQL ou PostgreSQL, atendeu às necessidades de armazenamento para este trabalho. Porém foi encontrada uma deficiência durante a operação do DNSpot: conforme o arquivo que guarda os armazenamentos do banco crescia, o tempo médio para inserir todos os registros relacionados a uma transação aumentava consideravelmente ao ponto de prejudicar o funcionamento do DNSpot. Foi observado que, quando o arquivo chegava a aproximadamente 2,2 GB, o tempo de

inserção de uma transação ficava tão grande que algumas requisições acabavam não sendo armazenadas. Isso ocorria em momentos em que o DNSpot estava recebendo ataques DoS, gerando um montante muito grande de registros por segundo que precisavam ser inseridos rapidamente para continuar recebendo as outras requisições. O tempo de inserção se tornava um problema quando ultrapassava o tempo do intervalo entre as requisições recebidas, ou seja, passava a demorar mais para inserir do que para receber uma nova requisição, gerando uma fila de transações a serem inseridas que aumentava constantemente e consumia recursos físicos do servidor instalado (memória RAM principalmente).

Para uma comparação, quanto um novo arquivo vazio do Sqlite3 é colocado em produção ele mantém uma média entre 0,03 e 0,06 segundos de inserção por transação, no primeiro dia em que ainda não chegou a 500 MB de tamanho. Ao ultrapassar os 2 GB, a média passa a ser cerca de 0,13 a 0,16 segundos por transação, até cinco vezes o tempo inicial. É um aumento significativo no tempo de interação com o banco, principalmente quando se lida com ataques DoS.

Dessa maneira, para evitar a perda de informações, adotou-se a estratégia de rotacionar o arquivo do banco de dados: quando chega a 2 GB, o arquivo existente é salvo em um diretório de histórico, e um novo arquivo vazio passa a ser usado. Isso não causa problemas para o DNSpot, pois novas inserções não dependem fortemente dos registros inseridos anteriormente, mas dificulta o processo de análise das transações pois mais de um arquivo deve ser consultado e pesquisado. Por isso um único arquivo para acesso de todas as informações é sempre preferível.

5.5 CONSIDERAÇÕES PARCIAIS

Neste capítulo é apresentada a implementação do DNSpot, e foram discutidos os resultados da análise de mais de 4 milhões de requisições DNS observadas pelo DNSpot durante 49 dias em operação. Cerca de 99% do tráfego consistiu de ataques DDoS baseados em reflexão, que usavam o DNSpot como amplificador. A análise do perfil estatístico do tráfego e uma investigação mais aprofundada de algumas anomalias encontradas revelaram aspectos do *modus operandi* dos atacantes que eram até então desconhecidos, como grupos de domínios com a

mesma estrutura de resposta e RRs adicionados em respostas com o único objetivo de aumentar o fator de amplificação, o que mostra que o DNSpot cumpriu bem o seu papel como um *honeypot* de pesquisa.

As escolhas de implementação da ferramenta revelaram-se, na maior parte, acertadas. O mecanismo de restrição diária de consultas mostrou eficácia na limitação do tráfego gerado pelo DNSpot durante ataques DDoS. O banco de dados Sqlite3 mostrou algumas limitações de escalabilidade que tiveram de ser contornadas de forma que não prejudicasse a coleta já em andamento. A otimização ou mesmo a substituição desse componente deverão ser avaliadas para a evolução futura da ferramenta.

Para um trabalho futuro, a rotina de inserção dos registros poderia ser otimizada para utilizar a concorrência entre as *threads* que processam as transações da melhor forma possível para eficiência do tempo de inserção dos registros. Além disso, pode-se alterar as configurações de funcionamento e armazenamento do Sqlite3 (PRAGMA) para melhor atender as necessidades específicas do sistema, melhorando o desempenho de inserção.

.

6 CONCLUSÃO

O DNS é um sistema distribuído bem sucedido em sua aplicação. É um dos componentes de maior importância para o funcionamento da Internet, sendo utilizado de maneira imperceptível por qualquer pessoa que use a rede. Existem, porém, vulnerabilidades estruturais do sistema DNS que comprometem a sua segurança. O sistema DNS pode ser atacado ou explorado para ataques e outras atividades fraudulentas. A virtual invisibilidade do DNS, que dificulta a detecção de problemas por parte de usuários comuns, e o fato de ser um protocolo pouco filtrado por *firewalls* e outros mecanismos de controle de acesso de rede o tornam um alvo atraente. Por conta disso, ele tem sido um protocolo cada vez mais visado na Internet.

Uma das maneiras de se proteger de ataques e usuários maliciosos é entender como os mesmos se comportam, como os ataques são realizados e quais são seus objetivos. Ou seja, observar seu inimigo, colher informação sobre o mesmo e analisar. Um *honeypot* é uma infraestrutura, sistema, programa ou serviço que é colocado em produção na rede para ser sondado e atacado. Como não tem outro propósito, qualquer interação com o mesmo é uma atividade suspeita e maliciosa. Com seu uso é possível detectar atividades maliciosas, colher informações sobre o ataque e seus autores, e também confundi-los ou atrasá-los fornecendo respostas lentas ou erradas.

Diante disso, este trabalho apresentou uma arquitetura e implementação de um *honeypot* que simula um servidor DNS recursivo, denominado DNSpot. A arquitetura e implementação desenvolvidas permitem efetuar a resolução de nomes, modificar as respostas e gravar as informações das transações DNS para análise posterior.

A arquitetura proposta foi colocada em produção durante 49 dias, coletando dados das interações de atacantes com o sistema na forma de consultas DNS. O serviço DNS, mesmo não tendo sido anunciado externamente, foi sondado e descoberto muito depressa pelos clientes, e os primeiros ataques DDoS usando amplificação foram realizados já no segundo dia em produção. No total foram observadas mais de 4 milhões de requisições DNS, mais de 99% das quais relacionadas a ataques DDoS. Isso fornece evidências concretas do risco oferecido por servidores DNS recursivos abertos, reforçando a importância de não permitir que

consultas vindas da rede externa sejam respondidas por servidores de organizações. O DNSpot também possibilitou identificar a existência de domínios especificamente projetados para proporcionar um fator de amplificação de tráfego da ordem de 100:1, e constatar a ocorrência de varreduras de SIP usando a porta UDP 53, tradicionalmente alocada para o DNS. Assim, é possível constatar que tanto o objetivo geral quanto os objetivos específicos definidos na Seção 1.1 foram alcançados.

A maior dificuldade encontrada foi relacionada ao armazenamento dos dados coletados, especialmente durante ataques DoS. Como o Sqlite apresentou um incremento no tempo de inserção com o crescimento do banco de dados, adotou-se a estratégia de rotacionar o arquivo de banco de dados, o que permitiu manter o tempo de inserção baixo e evitar a perda de dados. Essa estratégia, porém, é um paliativo, sendo necessário buscar uma solução mais perene para garantir escalabilidade.

Este trabalho pode ser estendido de diversas formas. Algumas perspectivas futuras incluem:

- Implantação de um novo banco de dados ou otimização do banco atual, de forma a resolver os problemas de escalabilidade encontrados;
- Implementação de suporte ao TCP como protocolo de transporte (a versão atual oferece suporte apenas a UDP);
- Permitir que o DNSpot possa capturar e armazenar os dados das camadas de rede (IP) e de transporte (TCP/UDP);
- Integração do DNSpot com sistemas de detecção de intrusões (por exemplo, para gerar alertas sobre *hosts* internos que estejam participando de ataques DDoS usando servidores DNS externos).

REFERÊNCIAS

ALBITZ, P.; LIU, C. **DNS and BIND**. 5ª edição. Sebastopol: O'Reilly, 2006. 640 p. ISBN 0-596-10057-4.

ARENDS R.; AUSTEIN R.; LARSON, M; MASSEY D.; ROSE S. **DNS Security Introduction and Requirements**. The Internet Society, RFC4033, p. 8, 2005a. Disponível em: <http://tools.ietf.org/html/rfc4033>. Acesso em: 01 Mar. 2015.

ARENDS R.; AUSTEIN R.; LARSON, M; MASSEY D.; ROSE S. **Resource Records for the DNS Security Extensions**. The Internet Society, RFC4034, 2005b. Disponível em: < https://www.ietf.org/rfc/rfc4034.txt>. Acesso em: 01 Mar. 2015.

ARENDS R.; AUSTEIN R.; LARSON, M; MASSEY D.; ROSE S. **Protocol Modifications for the DNS Security Extensions**. The Internet Society, RFC4035, 2005c. Disponível em: < https://www.ietf.org/rfc/rfc4035.txt>. Acesso em: 01 Mar. 2015.

ARENDS, R; BLACKA D.; LAURIE B.; SISSON G. **DNS Security (DNSSEC) Hashed Authenticated Denial of Existence**. The Internet Society, RFC5155, 2008. Disponível em: < https://tools.ietf.org/html/rfc5155>. Acesso em: 01 Mar. 2015.

BELLIS, R. **DNS Proxy Implementation Guidelines**. The Internet Society, RFC5625, 2009. Disponível em: < https://tools.ietf.org/html/rfc5625Acesso em: 01 Mar. 2015.

BELLIS, R. **DNS Transport over TCP - Implementation Requirements**. The Internet Society, RFC5966, 2010. Disponível em: < https://tools.ietf.org/html/rfc5966>. Acesso em: 01 Mar. 2015.

BLACKA D.; WEILER S. **Clarifications and Implementation Notes for DNS Security (DNSSEC)**. The Internet Society, RFC6840, 2013. Disponível em: < https://tools.ietf.org/html/rfc6840>. Acesso em: 01 Mar. 2015.

CAMARILLO, G.; HANDLEY, M.; JOHNSTON, A; ROSENBERG, J.; SCHULZRINNE, H..; PETERSON, J.; SPARKS, R.; SCHOOLER, E. **SIP: Session Initiation Protocol**. The Internet Society, Internet Engineering Task Force, RFC3261, 2002. Disponível em: < https://www.ietf.org/rfc/rfc3261.txt>. Acesso em: 01 Out. 2015.

CERT.br. **Honeypots e Honeynets: Definições e Aplicações**. CERT – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, 2007. Disponível em: < http://www.cert.br/docs/whitepapers/honeypots-honeynets/>. Acesso em: 01 Mar. 2015.

CERT.br. **Técnicas e Tendências nos Ataques de Negação de Serviços**. In: XXXIII SBRC, Simpósio Brasileiro de Redes de Computadores, maio de 2015, Vitória, ES. Disponível em: <http://www.cert.br/docs/palestras/certbr-sbrc2015.pdf>. Acesso em: 01 Nov. 2015.

CERT.br. **Recomendações para Evitar o Abuso de Servidores DNS Recursivos Abertos**. CERT – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil, 2013. Disponível em: <http://www.cert.br/docs/whitepapers/dns-recursivo-aberto/>. Acesso em: 01 Mar. 2015.

CERT.br. **Resultados Preliminares do Projeto SpamPots: Uso de Honeypots de Baixa Interatividade na Obtenção de Métricas sobre o Abuso de Redes de Banda Larga para o Envio de Spam** . 2007b. Disponível em: < http://www.cert.br/docs/whitepapers/spampots/>. Acesso em: 01 Mar. 2015.

CERT.br. **SpamPots Project**. 2015b. Disponível em: < http://honeytarg.cert.br/spampots/>. Acesso em: 01 Mar. 2015.

CERT.br. **Golpes na Internet**. Cartilha de Segurança para Internet. 2012. Disponível em: < http://cartilha.cert.br/golpes/>. Acesso em: 01 Mar. 2015.

CERT–SEI. **Buffer Overflows in Multiple DNS Resolver Libraries**. CERT-SEI - Software Engineering Institute, 2002a. Disponível em: < http://www.cert.org/historical/advisories/CA-2002-19.cfm>. Acesso em: 01 Mar. 2015.

CERT–SEI. **Vulnerability Note VU#844360: Domain Name System (DNS) stub resolver libraries vulnerable to buffer overflows via network name or address lookups**. Vulnerability Notes Database - CERT-SEI - Software Engineering Institute, 2003. Disponível em: <http://www.kb.cert.org/vuls/id/844360>. Acesso em: 01 Mar. 2015.

CERT–SEI. **Vulnerability Note VU#542971: Multiple vendors Domain Name System (DNS) stub resolvers vulnerable to buffer overflow via network name and address lookups**. Vulnerability Notes Database - CERT-SEI - Software Engineering Institute, 2002b. Disponível em: <http://www.kb.cert.org/vuls/id/542971>. Acesso em: 01 Mar. 2015.

CHANDRAMOULI, R; ROSE, S. **Secure Domain Name System (DNS) Deployment Guide**. NIST Special Publication 800-81-2, 2013, Computer Security, NIST-National Institute of Standards and Technology. Disponível em: < http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-81-2.pdf/>. Acesso em: 01 Mar. 2015.

CISCO. **A Cisco Guide to Defending Against Distributed Denial of Service Attacks**. 2013. Disponível em: < http://www.cisco.com/web/about/security/intelligence/guide_ddos_defense.html#_Toc374453053>. Acesso em: 01 Mar. 2015.

CONRAD, D. **Towards Improving DNS Security, Stability, and Resiliency**. Internet Society, 2012. Disponível em: <http://www.internetsociety.org/towards-improving-dns-security-stability-and-resiliency-0>. Acesso em: 01 Mar. 2015.

COOKE, E.; JAHANIAN F.; McPHERSON, D. **The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets**. USENIX, The Advanced Computing Systems Association, SRUTI'05 – Steps to Reducing Unwanted Traffic on the Internet Workshop, 2005. Disponível em: < https://www.usenix.org/legacy/event/sruti05/tech/full_papers/cooke/cooke_html/>. Acesso em: 01 Mar. 2015.

CULPAN, T. **Hackers Step Up Attacks in Southeast Asia Amid Tensions: FireEye**. 2015. Disponível em: < http://www.bloomberg.com/news/articles/2015-10-01/hackers-step-up-attacks-in-southeast-asia-amid-tensions-fireeye>. Acesso em: 01 Mar. 2015.

CVE. **CVE-2002-0684**. 2002. Disponível em: <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0684>. Acesso em: 01 Mar. 2015.

DAMAS, J.; GRAFF, M.; VIXIE, P. **Extension Mechanisms for DNS (EDNS(0))**. The Internet Society, RFC6891, 2013. Disponível em: < http://tools.ietf.org/html/rfc6891>. Acesso em: 01 Mar. 2015.

DE GROOT, G., J.; KARRENBERG, D.; LEAR, E.; MOSKOWITZ, B.; REKHTER, Y. **Address Allocation for Private Internets**. The Internet Society, RFC1918, 1996. Disponível em: <https://tools.ietf.org/html/rfc1918>. Acesso em: 01 Mar. 2015.

DIETRICH, C. J.; ROSSOW, C.; FREILING, F. C; BOS, H.; VAN STEEN, M.; POHLMANN, N. **On Botnets That Use DNS for Command and Control**. 2011 Seventh European Conference on Computer Network Defense (EC2ND). **Anais**... In: 2011 SEVENTH EUROPEAN CONFERENCE ON COMPUTER NETWORK

DEFENSE (EC2ND), Set. 2011. Disponível em: <http://kerstin.christian-rossow.de/publications/dnscnc2011.pdf>. Acesso em: 01 Mar. 2015.

ARBOR NETWORKS. **Digital Attack Map. Top Daily DDoS Attacks Worldwide**. 2015. Disponível em: < http://www.digitalattackmap.com>. Acesso em: 01 Mar. 2015.

DYER, S. P. **The Hesiod* Name Server.** In: Proceedings of the USENIX Winter Technical Conference, p. 183-189, 1988**.**

EASTLAKE, D. **Domain Name System Security Extensions**. The Internet Society, RFC2535, 1999. Disponível em: < https://tools.ietf.org/html/rfc2535>. Acesso em: 01 Mar. 2015.

EASTLAKE, D. **Domain Name System (DNS) IANA Considerations**. The Internet Society, RFC6895, 2013. Disponível em: < http://tools.ietf.org/html/rfc6895#section-2.3>. Acesso em: 01 Mar. 2015.

EASTLAKE, D. **Secret Key Establishment for DNS (TKEY RR)**. The Internet Society, RFC2930, 2000. Disponível em: < http://tools.ietf.org/html/rfc2930>. Acesso em: 01 Mar. 2015.

ENISA. **ENISA Threat Landscape 2014**., 2014. Disponível em: <https://www.enisa.europa.eu/activities/risk-management/evolving-threat-environment/enisa-threat-landscape/enisa-threat-landscape-2014>. Acesso em: 01 Mar. 2015.

ENISA. **Threat Landscape and Good Practice Guide for Internet Infrastructure**., 2015. Disponível em: < https://www.enisa.europa.eu/activities/risk-management/evolving-threat-environment/enisa-thematic-landscapes/threat-landscape-of-the-internet-infrastructure/iitl>. Acesso em: 01 Mar. 2015.

FIFIELD, A. **Seoul seeks hacker troops to fend off North Korean cyberattacks**. 2015. Disponível em: < https://www.washingtonpost.com/world/asia_pacific/south-korea-seeks-hackers-to-defend-against-north-korean-cyberattacks/2015/10/24/88bcbca0-7682-11e5-a5e2-40d6b2ad18dd_story.html>. Acesso em: 01 Mar. 2015.

FREEGEOIP.NET. **Freegeoip.net**. 2015. Disponível em: < http://freegeoip.net/?q=179.222.232.93>. Acesso em: 01 Mar. 2015.

GALLOIS, F. **Uma Avaliação de Desempenho do DNSSEC**. 2010. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Tecnológicas, Universidade do Estado de Santa Catarina, Joinville, 2010.

GAUCI, S. **Introduction to svmap**. Sipvicious.org, 2007. Disponível em: <http://blog.sipvicious.org/2007/11/introduction-to-svmap.html>. Acesso em: 01 Mar. 2015.

GONT, F. **Security Assessment of the Internet Protocol**. Centre for the Protection of National Infrastructure, 2008. Disponível em: < http://www.gont.com.ar/papers/InternetProtocol.pdf>. Acesso em: 01 Mar. 2015.

GONT, F. **Security Assessment of the Transmission Control Protocol (TCP)**. Centre for the Protection of National Infrastructure, 2009. Disponível em: < http://www.gont.com.ar/papers/tn-03-09-security-assessment-TCP.pdf>. Acesso em: 01 Mar. 2015.

GRIMES, R. A. **Honeypots for Windows**. Apress, 2005. 424 p. ISBN 1-59059-335-9.

GITHUB. **Kippo**. 2015. Disponível em: https://github.com/desaster/kippo>. Acesso em: 01 Mar. 2015.

HANDLEY, M.; ROSENBERG, J.; SCHOOLER, E.; SCHULZRINNE H. **SIP: Session Initiation Protocol**. The Internet Society, Internet Engineering Task Force, RFC2543, 1999. Disponível em: < https://tools.ietf.org/html/rfc2543>. Acesso em: 01 Out. 2015.

HOLZ, T. **A Short Visit to the Bot Zoo**. IEEE Computer Society, IEEE Security & Privacy, p. 76-79, 2005. Disponível em: < https://www1.informatik.uni-erlangen.de/filepool/publications/a-short-visit-to-the-bot-zoo.pdf>. Acesso em: 01 Mar. 2015.

HONEYD. **Developments of the Honeyd Virtual Honeypot**. 2008. Disponível em: < http://www.honeyd.org/index.php/>. Acesso em: 01 Mar. 2015.

HONEYD. **Honeyd Frequently Asked Questions**. 2007. Disponível em: < http://www.honeyd.org/faq.php#what/>. Acesso em: 01 Mar. 2015.

HONEYD. **Honeyd General Information**. 2004. Disponível em: < http://www.honeyd.org/general.php/>. Acesso em: 01 Mar. 2015.

HUBERT, A.; VAN MOOK, R. **Measures for Making DNS More Resilient against Forged Answers**. The Internet Society, RFC5452, 2009. Disponível em: < https://tools.ietf.org/html/rfc5452>. Acesso em: 01 Nov. 2015.

IANA. **Domain Name System (DNS) Parameters**. Internet Assigned Numbers Authority, 2015. Disponível em: < http://www.iana.org/assignments/dns-parameters/dns-parameters.xhtml#dns-parameters-6>. Acesso em: 01 Mar. 2015.

INACON. **DNS Message Format**. 2010. Disponível em: < http://www.inacon.de/ph/data/DNS/DNS-Message-Format_OS_RFC-1035.htm>. Acesso em: 01 Mar. 2015.

INETSIM. **INetSim: Internet Services Simulation Suite**. 2014. Disponível em: < http://www.inetsim.org/>. Acesso em: 01 Mar. 2015.

INTERNET ENGINEERING TASK FORCE. **Internet Protocol**. The Internet Society, RFC791, 1981. Disponível em: < http://tools.ietf.org/html/rfc791>. Acesso em: 01 Mar. 2015.

ISC. **BIND 9 Administrator Reference Manual**. Internet Software Consortium, USA, California, Redwood City, 2015. Disponível em: < ftp://ftp.isc.org/isc/bind9/cur/9.10/doc/arm/Bv9ARM.pdf>. Acesso em: 01 Out. 2015.

JAKOBSSON, M; RAMZAN, Z.; STAMM, S. **Drive-By Pharming**, Indiana University Bloomington, 2006. Disponível em: < http://www.symantec.com/avcenter/reference/Driveby_Pharming.pdf>. Acesso em: 01 Mar. 2015.

JHART. **Adventures in Empty UDP Scanning**. Information Security, 2014. Disponível em: < https://community.rapid7.com/community/infosec/blog/2014/10/03/adventures-in-empty-udp-scanning>. Acesso em: 01 Out. 2015.

KAÂNICHE, M.; ALATA, E.; NICOMETTE, V.; DESWARTE, Y.; DACIER, M. **Empirical Analysis and Statistical Modeling of Attack Processes based on Honeypots**, 2005. Disponível em: < http://homepages.laas.fr/~kaaniche/documents/Kaaniche-DSN-WEEDS2006/Kaaniche-WEEDS-DSN06-final.pdf>. Acesso em: 01 Mar. 2015.

KREIBICH, C.; CROWCROFT, J. **Honeycomb – Creating Intrusion Detection Signatures Using Honeypots**, 2004. Disponível em: < http://dpnm.postech.ac.kr/research/04/nsri/papers/honypot.pdf>. Acesso em: 01 Mar. 2015.

MAZIERO, C. A. **O Serviço DNS**. 2010. Disponível em < http://dainf.ct.utfpr.edu.br/~maziero/doku.php/espec:servico_dns>: . Acesso em: 01 Mar. 2015.

MEDEIROS, H. **ANÁLISE DE DESEMPENHO DO USO DE TCP PARA CONSULTAS DNS**. 2011. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Tecnológicas, Universidade do Estado de Santa Catarina, Joinville, 2011.

MOCKAPETRIS, P. **DOMAIN NAMES - CONCEPTS AND FACILITIES**. The Internet Society, RFC1034, 1987a. Disponível em: < http://tools.ietf.org/html/rfc1034>. Acesso em: 01 Mar. 2015.

MOCKAPETRIS, P. **DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION**. The Internet Society, RFC1035, 1987b. Disponível em: < http://tools.ietf.org/html/rfc1035>. Acesso em: 01 Mar. 2015.

MOON, D. A. **Chaosnet.** Massachusetts Institute of Technology Artificial Intelligence Laboratory, A.I Memo No. 628, 66 p., jun. 1981**.** Disponível em:< http://web.archive.org/web/20070930201300/http://bitsavers.org/pdf/mit/AIM-628_chaosnet.pdf>. Acesso em: 01 Mar. 2015.

NMAP. **Nmap Network Scanning**. Port Scanning Techniques. Capítulo 15 no Guia de Referência Nmap, 2015. Disponível em: <https://nmap.org/book/man-port-scanning-techniques.html>. Acesso em: 01 Out. 2015.

NETWORKS ASIA. **Hong Kong, Taiwan most exposed regionally to cyberattacks**. 2015. Disponível em: < http://www.networksasia.net/article/hong-kong-taiwan-most-exposed-regionally-cyberattacks.1444616464>. Acesso em: 01 Mar. 2015.

OPEN RESOLVER PROJECT. **Open Resolver Project**., 2015. Disponível em: < http://openresolverproject.org/>. Acesso em: 01 Mar. 2015.

PESCATORE, J. **Securing DNS to Thwart Advanced Targeted Attacks and Reduce Data Breaches**. A SANS Whitepaper, SANS Institute InfoSec Reading Room, 2014. Disponível em: <https://www.sans.org/reading-room/whitepapers/analyst/securing-dns-thwart-advanced-targeted-attacks-reduce-data-breaches-35597/>. Acesso em: 01 Mar. 2015.

RAISANEN, V.; GROTEFELD, G.; MORTON, A. **Network performance measurement with periodic streams**. The Internet Society, RFC3432, 2002. Disponível em: < https://tools.ietf.org/html/rfc3432>. Acesso em: 01 Mar. 2015.

SANTCROOS, M.; KOLKMAN, O. M. **DNS Threat Analysis**. NLnet Labs Document, 2007. Disponível em: <http://nlnetlabs.nl/downloads/se-consult.pdf>. Acesso em: 01 Mar. 2015.

SATELLITE. **Satellite Measuring the Internet's Stars**. University of Washington CSE Networks Lab, 2015. Disponível em: < http://satellite.cs.washington.edu/>. Acesso em: 01 Mar. 2015.

SCHUBA, C. L. **Addressing Weaknesses in The Domain Name System Protocol**. 99 p. Dissertação (Mestrado) — University of Purdue, West Lafayette, 1993.

SECURE64. **DoS Attack Knocks Out Microsoft Sites**. 2001. Disponível em: < http://www.secure64.com/news-hackers-microsoft-dns-switch>. Acesso em: 01 Mar. 2015.

SECURITY WEEK. **Reports of Massive DNS Outages in Germany**. 2010. Disponível em: <http://www.securityweek.com/content/reports-massive-dns-outages-germany>. Acesso em: 01 Mar. 2015.

SLAYER, P. **DDoS Attack on DNS Hits Amazon and Others Briefly**. PcWorld, 2009. Disponível em: <http://www.pcworld.com/article/185458/article.html>. Acesso em: 01 Mar. 2015.

SPITZNER, L. **Honeypots: Catching the Insider Threat**, 2003. Disponível em: < https://acsac.org/2003/papers/spitzner.pdf>. Acesso em: 01 Mar. 2015.

SPITZNER, L. **Honeypots: Tracking Hackers**. Addison Wesley, 2002.

STEDING-JESSEN, K.; VIJAYKUMAR, N. L.; MONTES, A. **Using low-interaction honeypots to study the abuse of open proxies to send spam.** INFOCOMP Journal of Computer Science , v. 7, n. 1, p. 45-53, 2008**.** Disponível em:<http://www.dcc.ufla.br/infocomp/index.php?option=com_content&view=article&id=362&Itemid=73>. Acesso em: 01 Mar. 2015.

TEAM CYMRU. **Secure BIND Template Version 7.3 07 Aug 2012**. 2012. Disponível em: <http://www.cymru.com/Documents/secure-bind-template.html>. Acesso em: 01 Out. 2015.

THE HONEYNET PROJECT. **HOW FAST-FLUX SERVICE NETWORKS WORK**. 2008a. Disponível em: < http://www.honeynet.org/node/132>. Acesso em: 01 Mar. 2015.

THE HONEYNET PROJECT. **Projects**. 2015b. Disponível em: < https://www.honeynet.org/project/>. Acesso em: 01 Mar. 2015.

THE HONEYNET PROJECT. **Sebek**. 2008b. Disponível em: https://www.honeynet.org/project/sebek >. Acesso em: 01 Mar. 2015.

THE HONEYNET PROJECT. **Welcome to the Honeywall project site**. 2015a. Disponível em: < https://projects.honeynet.org/honeywall/>. Acesso em: 01 Mar. 2015.

THE HONEYNET PROJECT AND ALLIANCE. **Know Your Enemy - A Profile**, 2003. Disponível em: < http://old.honeynet.org/papers/profiles/cc-fraud.pdf>. Acesso em: 01 Mar. 2015.

THE TCP/IP GUIDE. **DNS Message Header and Question Section Format**. The TCP/IP GUIDE, 2005. Disponível em: < http://www.tcpipguide.com/free/t_DNSMessageHeaderandQuestionSectionFormat.htm>. Acesso em: 01 Mar. 2015.

UNBOUND. **Requirements for Recursive Caching Resolver**. 2006. Disponível em: < https://www.unbound.net/documentation/requirements.html>. Acesso em: 01 Out. 2015

UNBOUND. **New Open Source DNS Server Released Today**. Press Release, 2008. Disponível em: < https://www.unbound.net/documentation/Unbound_Press_Release.pdf>. Acesso em: 01 Out. 2015

US-CERT. **Alert (TA13-088A) - DNS Amplification Attacks**. US-CERT – United States Computer Emergency Readiness Team, 2013. Disponível em: < https://www.us-cert.gov/ncas/alerts/TA13-088A>. Acesso em: 01 Mar. 2015

VIXIE, P.; THOMSON, S.; REKHTER, Y.; BOUND, J. **Dynamic Updates in the Domain Name System (DNS UPDATE)**. The Internet Society, RFC2136, 1997. Disponível em: < http://tools.ietf.org/html/rfc2136>. Acesso em: 01 Mar. 2015.

VIXIE, P. **Extension Mechanisms for DNS (EDNS0)**. The Internet Society, RFC2671, 1999. Disponível em: < https://tools.ietf.org/html/rfc2671>. Acesso em: 01 Mar. 2015.

VIXIE, P.; GUDMUNDSSON, O.; EASTLAKE, D.; Wellington, B. **Secret Key Transaction Authentication for DNS (TSIG)**. The Internet Society, RFC2845, 2000. Disponível em: < http://tools.ietf.org/html/rfc2845>. Acesso em: 01 Mar. 2015.

VOID-INFO. **SIP**. Voip-info.org, A reference guide to all things VOIP, 2015. Disponível em: < http://www.voip-info.org/wiki/view/SIP>. Acesso em: 01 Out. 2015.

VOID-INFO. **IETF Working Groups related to VOIP**. Voip-info.org, A reference guide to all things VOIP, 2012a. Disponível em: < http://www.voip-info.org/wiki/view/IETF>. Acesso em: 01 Out. 2015.

VOID-INFO. **SIP method options**. Voip-info.org, A reference guide to all things VOIP, 2012b. Disponível em: <http://www.voip-info.org/wiki/view/SIP+method+options>. Acesso em: 01 Out. 2015.

VOID-INFO. **SIP method options**. Voip-info.org, A reference guide to all things VOIP, 2012. Disponível em: < SIP method invite>. Acesso em: 01 Out. 2015.

APÊNDICE A - ESTRUTURA DO BANCO DE DADOS

I. DNS_CLIENT

A tabela DNS_CLIENT é utilizada para armazenar dados referentes aos clientes que interagem de alguma forma com o DNSpot, realizando consultas DNS (ou consultas anômalas não DNS). Cada cliente do DNSpot (representado pelo endereço IP) é identificado unicamente no sistema por um identificador único númerico atribuído ao mesmo de maneira incremental (coluna CLIENT_ID). A tabela armazena alguns dados estátisticos para cada cliente, tais como a quantidade total de requisições recebidas do cliente e respostas enviadas ao mesmo (o sistema pode escolher não responder uma requisição do cliente), quantidade de vezes que o sistema ignorou uma requisição do cliente (políticas de segurança), e as quantidades de vezes que os principais RCODEs foram respondidos ao cliente (*NoError, ServFail, NXDomain, FormErr* e outros.). Com essa tabela, é possível ter uma noção geral das quantidades de interações realizadas por um cliente com o DNSpot, e também identifica-lo unicamente no sistema em outras tabelas referenciado a chave primária da tabela (CLIENT_ID).

II. DNS_DOMAIN_SEARCHED e DNS_DOMAIN_DATA

As tabelas DNS_DOMAIN_SEARCHED e DNS_DOMAIN_DATA guardam informações referents aos nomes de domíios requisitados nas mensagens DNS. A primeira armazena todos os domínios recebidos em mensagens de requisição DNS, enquanto a segunda armazena todas as respostas associadas à um domínio. Em ambos os casos existem contadores para determinar a quantidade de ocorrência de um dado domínio e suas respostas associadas.

III. DNS_TRANSACTION

A tabela DNS_TRANSACTION representa uma transação DNS processada pelo sistema DNSpot. Uma transação DNS é interpretada pelo sistema como todo o

processo de resolução de nomes, para gerar a resposta da requisição recebida por um cliente. Cada transação de um dado cliente recebe um identificador único que é incrementado automaticamente (coluna TRAN_ID). A tabela armazena informações gerais de uma transação DNS, tais como:

- Porta utilizada pelo cliente para a transação DNS
- Tempo de recebimento da requisição do cliente e do envio da resposta.
- Tamanho das mensagens de requisição e de resposta DNS.
- Código de resposta (RCODE) enviado na resposta para o cliente.
- Se a transação foi bem sucedida, ou seja, se não houveram erros durante o processamento das mensagens DNS, se as mensagens estavam formatadas corretamente, e se o cliente foi respondido com sucesso.
- Se a transação foi ignorada devido a alguma política de segurança do DNSpot, ou seja, se a requisição não foi respondida ao cliente.

Com essa tabela, é possível fazer uma análise geral de uma dada transação DNS realizada por um cliente. Além disso, a chave primária desta tabela (TRAN_ID e CLIENT_ID) é usada como chave estrangeira em outras tabelas que guardam dados mais espécificos e detalhados relacionados ao processamento de uma transação DNS pelo sistema.

IV. DNS_RECV_HEADER e DNS_SENT_HEADER

As tabelas DNS_RECV_HEADER e DNS_SENT_HEADER armazenam as informações de todos os campos presentes no cabeçalho (*header*) de uma mensagem DNS de requisição e de resposta, respectivamente, conforme revisados na Seção 2.4.1. Através dessas tabelas é possível analisar os dados de cabeçalho de uma mensagem DNS.

V. DNS_RECV_QUESTION e DNS_SENT_QUESTION

As tabelas DNS_RECV_QUESTION e DNS_SENT_QUESTION armazenam as informações de todos os campos presentes na seção de pergunta de uma mensagem DNS de requisição e de resposta, respectivamente, conforme revisados

na Seção 2.4.2. Através dessa tabela é possível analisar quais foram os domínios, classes e tipos solicitados nas requisições DNS.

VI. TABELAS DE RRs

Os RRs associados as seções de resposta, autoridade e adicional de uma mensagem DNS possuem a mesma formatação dos campos, conforme definido na Seção 2.4.3. Dessa maneira, as tabelas que armazenam as informações dessas seções possuem a mesma definição, diferenciando somente o nome. São elas a DNS_RECV_ANSWER, DNS_RECV_AUTHORITY, DNS_RECV_ADDITIONAL, DNS_SENT_ANSWER, DNS_SENT_AUTHORITY e DNS_SENT_ADDITIONAL, onde as três primeiras armazenam os dados das seções de pergunta, autoridade e adicional de uma requisição DNS, enquanto as três últimas armazenam os de respostas DNS. Essas tabelas permitem a análise de todos os dados associados aos RRs enviados em mensagens DNS.

VII. DNS_RECV_RAWDATA e DNS_SENT_RAWDATA

Todas as mensagens DNS recebidas pelo *proxy* do DNSpot estão em formato de *bytes*, antes de serem processadas e seus dados extraídos. Essas mensagens em em formatação de *bytes* não processadas são armazenadas nas tabelas DNS_RECV_RAWDATA e DNS_SENT_RAWDATA, onde na primeira são armazenadas as requisições recebidas do cliente, e na segunda são armazenadas as respostas recebidas do servidor DNS real. Com essas tabelas, é possível reprocessar as mensagens DNS novamente a qualquer momento para extrair as informações, se for necessário (se ocorrer erros em um processamento anterior, devido a uma mensagem mal-formatada). Ou se caso for recebido uma mensagem não DNS anômala, investiga-lá.

VIII. DNS_RECV_BADFORMAT

As mensagens recebidas pelo sistema DNSpot que não estarem formatadas corretamente para processamento e extração dos dados de uma mensagem DNS

são armazenadas no formato em *bytes* recebido na tabela DNS_RECV_BADFORMAT, para análise posterior. Mensagens mal formatadas também incluem mensagens que não são DNS, ou seja, são mensagens anômalas de outros protocolos que foram enviadas a porta configurada para o DNSpot.

IX. DNS_TRAN_PROBLEM

A tabela DNS_TRAN_PROBLEM indica se houve algum problema durante o processamento de uma transação DNS, tais como cliente não respondido, conexão perdida com o cliente (*socket* e porta fechados), mensagem DNS mal formatada e erros de conexão ou funcionamento do servidor real interagido pelo *proxy*.

X. DNS_DATABASE_ERROR

A tabela DNS_DATABASE_ERROR indica se houve problemas de conexão com o banco de dados durante a rotina de armazenamento dos dados de uma transação. Determina quais tabelas foram gravadas com sucesso e quais não foram. Dessa maneira pode-se saber quais dados estão consistentes para análise e quais precisam ser reprocessados.

XI. DNS_EXCEPTION_RAISED

A tabela DNS_EXCEPTION_RAISED armazena todas as exceções e erros ocorridos durante a execução do DNSpot, de modo a permitir detectar e corrigir *bugs* na implementação do sistema.

XII. DNS_STATISTICS

A tabela DNS_STATISTICS armazena estatísticas gerais do funcionamento do DNSpot, tais como o tempo em média do processo de resolução de nomes pelo servidor e o tempo médio de armazenamento dos dados em tabela.

XIII. DNS_PARAMETROS_GERAIS

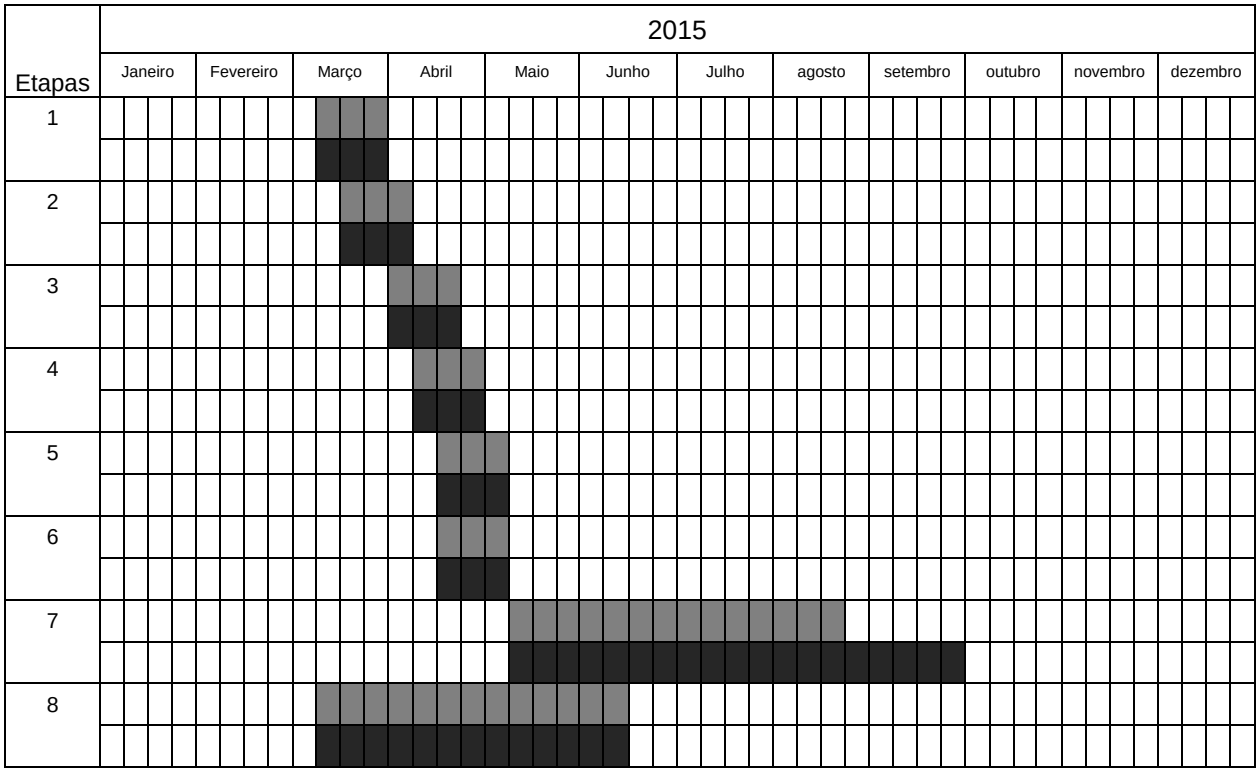
A tabela DNS_PARAMETROS_GERAIS armazena parametros gerais utilizados pelo sistema, tais como a resposta a ser enviada à uma requisição *VERSION.BIND*, a porta e a *interface* configuradas para o DNSpot, a porcentagem de *Server Fail* (RCODE=2) enviadas pelo DNSpot e a quantidade máxima de requisições diárias processadas e respondidas com sucesso para um dado clinete.

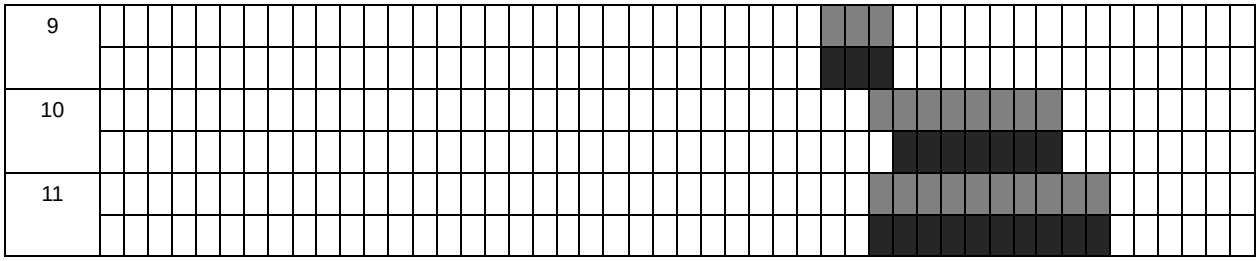
XIV. DNS_IGNORED_IP e DNS_IGNORED_SEARCHED_DOMAIN

O DNSpot permite que transações associadas a certos endereços IP (clientes) e domínios sejam ignoradas, ou seja, as requisições não são processadas e resolvidas pelo servidor real, e o cliente não recebe nenhuma resposta. Esses IPs e domínios a serem ignorados são armazenados nas tabelas DNS_IGNORED_IP e DNS_IGNORED_SEARCHED_DOMAIN respectivamente.

APÊNDICE B - CRONOGRAMA

1) Revisão bibliográfica sobre DNS
2) Revisão bibliográfica sobre aspectos de segurança do DNS.
3) Revisão bibliográfica sobre *honeypot*s e *honeynets*.
4) Revisão bibliográfica de tabalhos correlatos.
5) Escolha e obtenção de um servidor DNS recursivo.
6) Especificação de requisitos e definição de funcionalidades desejadas para o DNSpot.
7) Implementação de funcionalidades desejadas.
8) Escrita da parte conceitual da monografia.
9) Implantação do DNSpot em produção.
10) Coleta e análise dos dados obtidos.
11) Finalização da escrita da monografia e conclusões.





Legenda:

Cronograma Proposto

Cronograma Real