

Universidade Federal de Pernambuco
Graduação em Ciência da Computação

Centro de Informática
2008.2



DETECÇÃO DE ANOMALIAS NO PROTOCOLO DNS

Trabalho de Graduação

RODRIGO DIEGO MELO AMORIM

Orientador: Prof. Djamel Sadok (jamel@cin.ufpe.br)

Co-orientador: Prof. Eduardo Feitosa (efeitosa@gprt.ufpe.br)

Recife,
2008

Universidade Federal de Pernambuco
Graduação em Ciência da Computação

Centro de Informática
2008.2

DETECÇÃO DE ANOMALIAS NO PROTOCOLO DNS

Trabalho de Graduação

RODRIGO DIEGO MELO AMORIM

Projeto de Graduação apresentado
no Centro de Informática da Universidade
Federal de Pernambuco por Rodrigo Diego
Melo Amorin, orientado pelo Prof. PhD.
Djamel Sadok, como requisito parcial para
a obtenção do grau de Bacharel em
Ciência da Computação

Orientador: Prof. Djamel Sadok (jamel@cin.ufpe.br)

Co-orientador: Prof. Eduardo Feitosa (efeitosa@gprt.ufpe.br)

Recife,
2008

FOLHA DE APROVAÇÃO

DETECÇÃO DE ANOMALIAS NO PROTOCOLO DNS

RODRIGO DIEGO MELO AMORIM

APROVADO EM 04 DE DEZEMBRO DE 2008

BANCA EXAMINADORA:

Prof. Djamel Fawzi Hadj Sadok, PhD –
UFPE (Orientador)

Prof. Ruy José Guerra Baretto de Queiroz, PhD –
UFPE (Avaliador)

“Não há caminho fácil para a liberdade”

Nelson Mandela

“Sim, nós podemos!”

Barack Obama

“Cacildis!”

Mussum

Agradecimentos

O término desta empreitada é devido a vários personagens que estiveram presentes e que contribuíram mesmo em pensamento, para que eu pudesse chegar a este final, esta é a oportunidade de agradecê-los.

Quero agradecer primeiramente a quem deu início a esta caminhada, meu pai Marcos Antônio que sempre acreditou na educação e que nunca abdicou de oferecer tal tesouro, deixando uma herança para as futuras gerações maior do que tudo o que ele poderia deixar: cultura e educação. Agradecer também à minha madrastra, Ana Rita, e a minha avó, Maria de Lourdes, que foram como mães e que nunca desistiram de me apoiar e me dar forças quando para continuar. Agradecer a todos os meus parentes, irmãos, primos, primas, tios, tias e à lembrança do meu avô, Heleno Louro, que esteve presente durante esta graduação, mas que nos deixou antes de ver o final chegar.

Agradecer a todos os professores desta graduação que, através desta profissão belíssima que é a de compartilhar o conhecimento com o próximo, contribuíram para a minha formação acadêmica, principalmente aos meus guias acadêmicos e profissionais, Prof^o Djamel e Prof^a Judith, que me deram a oportunidade de crescer como profissional e de aprender cada vez mais, a cada dia que passa. Gostaria de agradecer em especial ao meu co-orientador Eduardo Feitosa, que com sua tamanha paciência esteve presente contribuindo para a qualidade excepcional deste trabalho.

Gostaria de agradecer também aos amigos que fiz e que continuo fazendo ao longo da minha graduação, e da estadia nesta cidade. Seria um clichê estranho dizer que sem eles eu não teria chegado até aqui. Teria chegado sim, talvez não fosse tão divertido ou não tivesse a mesma graça, mas chegaria.

Por fim gostaria de agradecer também a todos que deram a sua contribuição para a continuidade deste trabalho, mesmo que pequena, como Jorge e todos os funcionários do CIn, Maninho e sua família por proporcionarem um ambiente de socialização no cavanhaque, e por aí vai uma lista de nomes.

A todos, um muito obrigado!

Resumo

O protocolo DNS constitui uma importante parte da infra-estrutura da Internet, fornecendo um serviço que se tornou essencial ao longo dos anos. Sua disponibilidade é crucial para o bom andamento de quase todos os serviços Internet. Uma vez que novas técnicas de ataque a serviços na Internet não param de ser criadas e o DNS tornou-se um dos principais alvos, este trabalho levanta as deficiências encontradas atualmente no protocolo DNS, analisa novas técnicas de detecção de anomalias nesse protocolo e faz um estudo de caso da técnica de detecção de anomalia, bem atual e preocupante, chamada *Fast Flux Domains*, muito utilizada para proliferação de tráfego não desejado.

Sumário

1	Introdução	11
1.1	Objetivos.....	12
1.2	Estrutura do Trabalho.....	12
2	Conceitos Básicos	13
2.1	O Protocolo DNS.....	13
2.1.1	Elementos do DNS	15
2.1.2	Consultas ao DNS.....	16
2.1.2.1	Cache	17
2.1.2.2	Consulta reversa	18
2.1.3	Formato do Protocolo	19
2.2	Anomalias	20
2.2.1	<i>Fast-Flux Domains</i>	21
2.2.1.1	Anatomia do Fast Flux Domain.....	22
2.2.1.2	Round Robin DNS.....	24
2.2.1.3	CDN (Content Delivery Network).....	24
2.2.2	Typo-Squatter Domains	24
2.2.3	Uso de Endereços Privados	25
2.2.3.1	DNS Rebind	26
2.2.3.2	Darknets.....	27
3	Trabalhos Relacionados.....	28
3.1	Trabalhos propostos	28
3.2	Ferramentas.....	31
3.2.1	<i>DNSwatch</i>	31
3.2.2	<i>DNSlogger</i>	31
3.2.3	honeyDNS	32
3.2.4	DNStop e DNScap	32
3.2.5	DSC.....	32
4	Detecção de Anomalias no Protocolo DNS.....	33
4.1	Algoritmo	33
4.1.1	<i>Fluxiness</i>	34
4.1.2	<i>Flux-Score</i>	35
4.2	Implementação.....	35

4.2.1	Captura.....	36
4.2.2	Base	37
4.2.3	Análise.....	37
4.3	Execução	37
5	Resultados e Discussões	39
5.1	Validação	39
5.2	Ambiente de Teste.....	41
5.3	Resultados	42
5.3.1	TTL.....	43
5.3.2	<i>Flux-score</i>	43
5.3.3	Discussões.....	45
6	Conclusão.....	47
6.1	Dificuldades Encontradas	47
6.2	Trabalhos Futuros	48
	Referências	49
	Glossário	52
	Apêndice A – Parâmetros DNS	54

Índice de Figuras

Figura 2.1 - Estrutura Hierárquica do Espaço de endereçamento de nomes.....	14
Figura 2.2 - Divisão do domínio <i>ufpe.br</i> em zonas	16
Figura 2.3 - Resolvendo <i>garanhuns.ufpe.br</i> iterativamente na Internet	17
Figura 2.4 - Domínio in-addr.arpa	18
Figura 2.5 - Formato da mensagem DNS.....	19
Figura 2.6 - Formato dos Registros de Recurso	20
Figura 2.7 - Arquitetura de Ataque utilizando <i>Fast Flux Domains</i>	23
Figura 2.8 - Arquitetura do Ataque DNS <i>Rebind</i>	27
Figura 3.1 - Arquitetura de software do <i>dnslogger</i>	30
Figura 4.1 - Arquitetura do software desenvolvido.....	35
Figura 4.2 - Duas fases de execução do <i>software</i> : (a) Modelagem e Persistência e (b) Análise	38
Figura 5.1 - Resultado da Análise dos domínios maliciosos	41
Figura 5.2 - Arquitetura do caso estudado	42
Figura 5.3 - Distribuição das respostas nas faixas de <i>TTL</i> 's de 0 a 1900	43
Figura 5.4 - Falsos positivos do domínio <i>akamai.net</i>	44
Figura 5.5 - Resultado total de falsos positivos	45

Índice de Tabelas

Tabela 2.1 - Exemplo de registros DNS de <i>Fast Flux Domains</i>	22
Tabela 2.2 - Tipos de erros de domínios.....	25
Tabela 3.1 - Discriminação das categorias de consultas	29
Tabela 4.1 - Exemplo de registros persistidos na base.....	36
Tabela 5.1 - Lista de domínios maliciosos (24/11)	40
Tabela 5.2 - Falso positivo do domínio <i>freenode.net</i>	44
Tabela 5.3 - Falha do <i>flux-score</i>	45

1 Introdução

A área de segurança em redes de computadores apresenta-se como uma das maiores preocupações das empresas atualmente. Os ataques estão cada vez mais complexos e é necessário encontrar novas estratégias que ajudem a distinguir anomalias do tráfego normal de uma rede [23] .

Novas e bem conhecidas vulnerabilidades nos protocolos e infra-estruturas de redes são exploradas constantemente. As técnicas de ataque que até pouco tempo atrás eram conhecidas por poucos, hoje estão disponíveis, a qualquer curioso, na Internet. Além disso, elas vêm se tornando mais complexas e críticas para o bom funcionamento de servidores na rede. É nesse conturbado cenário que um dos alicerces da Internet [22] tornou-se um alvo fácil de ataques, principalmente os de negação de serviço.

O serviço fornecido pelo *Domain Name Service* (DNS) é de fundamental importância para o funcionamento da Internet e de outros serviços providos nela. Um servidor DNS, em estado normal, trabalha resolvendo os nomes dos domínios da Internet (www.cin.ufpe.br) para endereços IP (150.161.2.9) e vice-versa. Sem o DNS, qualquer pessoa que desejasse acessar um domínio teria memorizar seu endereço IP. É óbvio que devido à enorme quantidade de domínio, essa tarefa seria bastante desgastante e infactível.

Assim como boa parte do que foi projetado e desenvolvido para uso na Internet, o DNS também não contemplou aspectos de segurança durante sua implementação e por isso é alvo de ataques ou anomalias. Entre as principais pode-se destacar: os ataques de negação de serviço (DoS ou DDoS) diretos provenientes de *botnet*, vírus, *worms* e *spams*. Outro tipo com conseqüências piores é o envenenamento de cache (*DNS cache poisoning*), onde o objetivo é corromper um determinado servidor DNS com o intuito de direcionar qualquer requisição feita a ele para um site com um servidor DNS malicioso [25] .

No que diz respeito a soluções e mecanismos de detecção de anomalias ao DNS, as principais são técnicas baseadas em assinaturas e através de entropia, mas poucas soluções ainda existem. Entretanto, as ferramentas existentes falham, em sua maioria, por não conseguirem entender o comportamento do tráfego DNS.

Nesse contexto, a captura de tráfego DNS e sua posterior análise para identificação de padrões e conseqüentemente de anomalias é um alvo interessante de pesquisa.

1.1 Objetivos

Este trabalho tem por objetivo estudar o protocolo DNS e implementar uma ferramenta para detecção de anomalias. Para tanto, o funcionamento e as principais falhas de segurança do protocolo DNS serão estudadas, além das possíveis ferramentas existentes.

A idéia central do trabalho é elaborar uma ferramenta de detecção de anomalias *Fast-Flux Domains*, específica do protocolo DNS bastante em voga na Internet, visto que é utilizada na proliferação de tráfego não desejado como ataques, tentativas de fraude e *spam* [40].

Para assegurar a robustez, precisão e eficiência dessa ferramenta, serão realizados testes de validação e análises com dados reais.

1.2 Estrutura do Trabalho

O restante deste trabalho está dividido da seguinte forma: o Capítulo 2 descreve alguns conceitos básicos necessários para o correto entendimento do problema e sua solução, como o funcionamento do protocolo DNS e suas principais anomalias. O Capítulo 3 apresenta alguns trabalhos realizados nesta área de estudo com o propósito de analisar as anomalias e tentar detectá-las. O Capítulo 4 descreve o algoritmo utilizado, no escopo deste trabalho, o *Fast Flux Domains*. O Capítulo 5 detalha os experimentos realizados e seus resultados. Por fim as conclusões e trabalhos futuros encontram-se no Capítulo 6.

Também são disponibilizadas (ao final deste texto) as referências utilizadas, um glossário de termos importantes encontrados ao longo do documento e um apêndice contendo os principais parâmetros utilizados nas *flags* e cabeçalhos da mensagem do protocolo DNS.

2 Conceitos Básicos

Este capítulo visa proporcionar o entendimento dos conceitos relacionados ao DNS. A primeira seção descreve o funcionamento do protocolo e do serviço DNS, elucidando alguns pontos importantes como estruturas e mensagens do protocolo. A segunda seção descreve os diversos tipos de comportamentos anômalos relacionados ao tráfego do DNS.

2.1 O Protocolo DNS

O *Domain Name System* (DNS) [36] é um sistema para atribuição e distribuição de nomes para computadores e serviços de redes baseado na pilha TCP/IP. A nomeação oferecida pelo DNS é usada para localização de computadores e serviços através da utilização de nomes amigáveis. Por exemplo, é muito mais fácil e interessante acessar o *site* do Centro de Informática através de um nome amigável como *www.cin.ufpe.br* do que usando o endereço IP do servidor HTTP, no caso 150.161.2.9.

O conceito de DNS foi criado por Mokapetris [36] [37] para acompanhar o grande crescimento da Internet na época de 1987. Antes do DNS, todos os nós (*hosts* e servidores) integrantes da rede ARPANET possuíam um arquivo local (e.g. */etc/hosts* no UNIX) onde existia um mapeamento (tabela) entre nomes e os respectivos endereços dos integrantes da rede. Esses arquivos eram gerenciados centralmente, pelo SRI-NIC (*Stanford Research Institute - Network Information Center*), e cada computador na rede atualizava seu arquivo periodicamente.

Evidentemente essa solução não era escalável e tornou-se insuficiente e incapaz de suportar o crescimento exponencial da quantidade de endereços. A solução encontrada então foi à criação de um sistema hierárquico de gerenciamento de nomes, capaz de operar distribuidamente e resolver nomes de servidores e serviços em endereços IP. Esta solução foi a criação do protocolo DNS.

De modo geral, o DNS pode ser visto como um imenso banco de dados distribuído e hierárquico, cuja resolução de nomes ocorre consultando-se diversos servidores DNS e esta consulta pode ocorrer em vários níveis ("servidor-filho" para "servidor-pai"), diminuindo assim a cobertura de cada consulta. A estrutura hierárquica do DNS é ilustrada na figura 2.1.

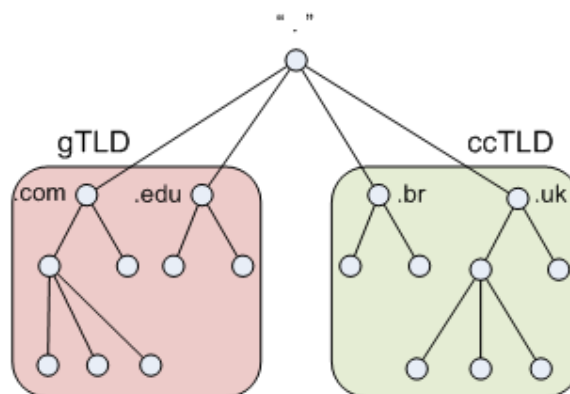


Figura 2.1 - Estrutura hierárquica do espaço de endereçamento de nomes

A base de dados do DNS é indexada por nomes de domínio [14], onde cada domínio é um conjunto de endereços que um determinado servidor pode armazenar (por exemplo, *ufpe.br*). Esses endereços estabelecem uma estrutura lógica de árvore denominada Espaço de nome de domínio. A raiz da árvore é o domínio raiz chamado *root* e representado por “.” (topo da figura 2.1). O segundo nível da árvore é formado pelos domínios principais chamados TLD (*Top Level Domain*), que podem ser agrupados em dois tipos: os gTLD (*Generic Top Level Domain*) que representam domínios como *.com*, *.net*, *.org*, *.edu* e etc; e os ccTLD (*Country Code Top Level Domain*) que representam os países como, por exemplo, *.br* para o Brasil, *.uk* para o Reino Unido, *.us* para os Estados Unidos.

Os **nomes de domínios absolutos** [37] (por exemplo, *ufpe.br*) representam o caminho da folha até a raiz e são chamados de FQDN (*Fully Qualified Domain Name*). Esses nomes de domínio têm a função de identificar: (i) o nó raiz do domínio, ou seja, o servidor local de nomes responsável pelo domínio, (ii) outros subdomínios como, por exemplo, *cin.ufpe.br* e *ctg.ufpe.br*.

O servidor local de nomes pode delegar parte dos endereços dos seus domínios para um subdomínio, que passa a ser responsável pelos endereços relativos a este subdomínio. Os nomes dos subdomínios de um determinado domínio podem ser chamados de **nomes de domínios relativos** [37].

O uso dessa estrutura hierárquica resolve dois problemas do antigo serviço de atribuição de nomes: unicidade e consistência dos nomes, e o tamanho do espaço de busca. Em relação ao primeiro problema, o DNS permite que um domínio possa ter n outros subdomínios e, desta forma, é possível haver dois nomes de domínios relativos iguais (*www.cin.ufpe.br* e *webmail.cin.ufpe.br*), contanto que estejam em subdomínios diferentes.

Para o segundo problema, o uso da hierarquia de domínios do DNS permite reduzir o espaço de busca consideravelmente, uma vez que a consulta é muito mais rápida quando realizada numa estrutura hierárquica.

2.1.1 Elementos do DNS

O DNS é composto basicamente por três elementos [36] [37] : registro de recursos, servidores de nomes e resolvidor.

Os **Registros de Recursos** (RR - *Resource Record*) correspondem ao formato dos dados que descrevem as propriedades de um domínio e seus *hosts* e que são armazenados nos servidores DNS. Cada registro de recurso é um tupla de quatro elementos que contém os seguintes campos *<nome, valor, tipo, TTL>*, onde *nome* corresponde ao nome que está sendo mapeado; *valor* é a tradução deste nome; *tipo* é o tipo de endereço que esta tradução retorna (esses tipos estão todos discriminados no anexo A); e *TTL (Time-To-Live)* é o tempo de vida deste registro, ou seja, é o tempo para o qual esta informação continua válida.

Por exemplo, uma consulta ao domínio *www.cin.ufpe.br* sobre seu endereço IP (consulta tipo A) retorna o seguinte RR *<www.cin.ufpe.br, cabo.cin.ufpe.br, CNAME, 320>*, onde o tipo CNAME redireciona a consulta a outro nome de domínio cuja resposta será *<cabo.cin.ufpe.br, 151.161.2.4, A, 320>*. No caso de existir mais de um endereço IP, seria retornado um registro para cada IP.

Servidores de Nomes são programas que mantêm as informações sobre determinada parte da árvore DNS, chamada de zona. Um servidor de nomes corresponde a um nó na árvore e responde com autoridade para os endereços daquela zona. Caso haja subdomínios, ele pode delegar para outro servidor de nomes os endereços de tal subdomínio com o intuito de descentralizar a administração dessas informações (figura 2.2). A tarefa básica de um servidor de nomes é responder as requisições usando dados da sua zona.

Resolvidor é a implementação, do lado do cliente, que acessa o servidor de nomes. Normalmente esta implementação é feita no nível de Sistema Operacional [1] , ou seja, todo e qualquer programa executando em um *host* que necessite de informação do espaço de nomes de domínio utiliza o resolvidor. Na prática, é um conjunto de rotinas que faz a consulta DNS ao servidor de nomes mais próximo, interpreta as respostas e retorna a informação ao programa que a solicitou.

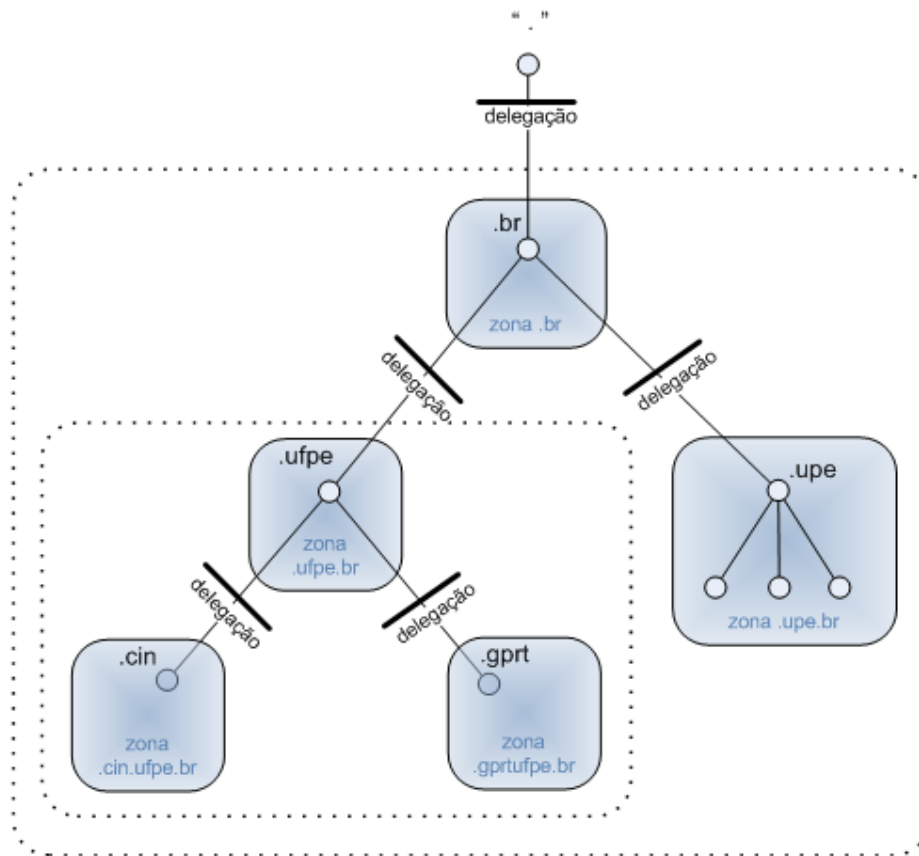


Figura 2.2 - Divisão do domínio *ufpe.br* em zonas

2.1.2 Consultas ao DNS

O processo de obter informação do espaço de nomes de domínio é chamado de resolução de nomes. Supondo, como cenário de exemplo, que um usuário tente acessar o portal *www.cin.ufpe.br*, o processo de resolução de nomes ocorre da seguinte forma:

1. O usuário usando um navegador *Web* tenta acessar determinado serviço na Internet através da sua URL;
2. A aplicação (navegador) repassa a solicitação para o resolvedor da máquina que faz a consulta (*query*) DNS para o servidor mais próximo, geralmente o servidor local;
3. Caso não tenha, faz uma solicitação ao servidor raiz mais próximo, que por sua vez a repassa ao servidor de nomes que tem autoridade sobre o domínio solicitado.

Neste ponto a resolução pode ser realizada de duas maneiras: recursiva ou iterativa. Na resolução iterativa (figura 2.3) o servidor de nomes local faz a requisição ao servidor de

nomes raiz que, caso não tenha a resposta, repassa o endereço do servidor de nomes que tem autoridade sobre aquele domínio. Assim, o servidor de nomes local refaz a requisição que agora será direcionada ao servidor com autoridade. Este processo se repete até encontrar a resposta requerida ou uma mensagem de erro – normalmente uma mensagem do tipo *Non-Existent Domain* (NXDOMAIN).

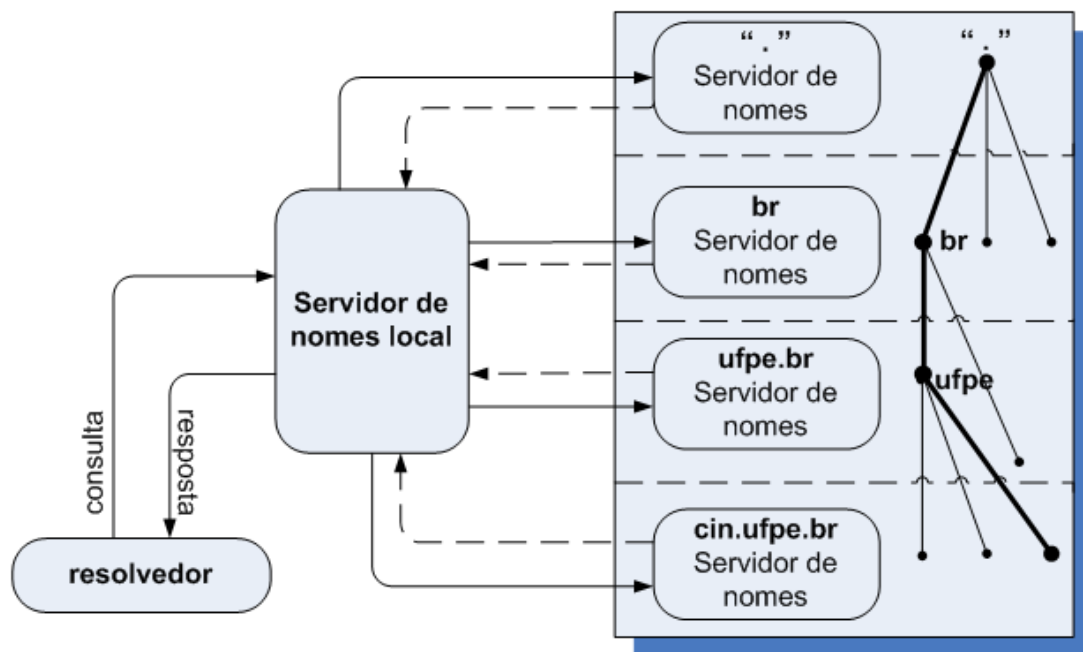


Figura 2.3 - Resolvendo *garanhuns.ufpe.br* iterativamente na Internet

Já na resolução recursiva, o servidor de nomes local faz a consulta a um determinado servidor de nomes que caso não tenha a resposta faz a requisição para o correto servidor de nomes em nome do servidor de nomes originalmente requisitante. Caso obtenha a resposta, retorna-a ao servidor de nomes local do *host* requisitante. Este processo se repete de tal forma que o servidor de nomes local faz apenas uma requisição e espera por uma resposta positiva ou uma mensagem de erro.

De modo geral, todas as consultas são recursivas, exceto a consulta que parte do servidor de nomes local ao servidor de nomes raiz, que é iterativa no intuito de evitar sobrecargas.

2.1.2.1 Cache

Durante o processo de resolução de nomes, o servidor arquiva na memória *cache* local todo mapeamento DNS que recebe como resposta. Desta forma, quando uma consulta já feita ao

servidor de nomes pode ser respondida pelo mapeamento que está na *cache*, mesmo que não possua autoridade para este nome. Esta funcionalidade do DNS é chamada de *caching* e permite maior eficiência e velocidade nas respostas das requisições. A garantia de que a informação armazenada na *cache* foi validada através de um esquema de tempo de limite de vida (TTL).

2.1.2.2 Consulta reversa

Uma opção de resolução não muito comum, mas também existente, é a resolução de endereços IP pelos seus respectivos nomes (URL) chamada de consulta reversa (*reverse query DNS*). Para realizar este tipo de consulta, existe um domínio do espaço de nomes de domínios que indexa todos os *hosts* pelos seus endereços IP, diferentemente do restante dos domínios que indexam pelos nomes de domínios, que é o domínio **IN-ADDR.ARPA**. A resolução ocorre igual a uma normal, o que a diferencia é o argumento passado para consulta, que ao invés de ser um nome de domínio é um endereço IP, e que recebe como resposta um nome de domínio associado a este endereço IP (caso exista). O tipo de dado armazenado é diferente, e é um dado do tipo PTR (*Domain Name Pointer*).

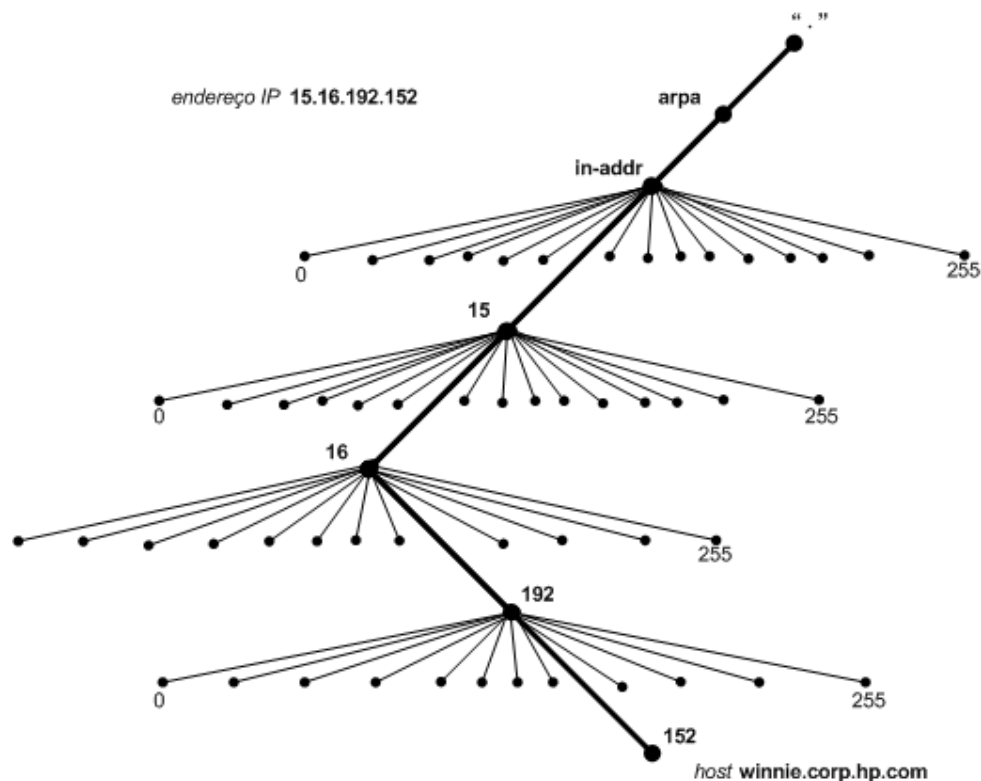


Figura 2.4 - Domínio in-addr.arpa

2.1.3 Formato do Protocolo

Como mencionado anteriormente, o DNS utiliza dois tipos de mensagens para resolução de nomes: consultas e respostas. Ambos os tipos seguem um formato único composto por um cabeçalho, contendo um número fixo de campos e até quatro secções que transportam os parâmetros de consulta, respostas, autoridade e adicionais (figura 2.5). O formato das mensagens é o seguinte:

- **Cabeçalho**- possui um conjunto de *flags* que indicam se a mensagem é uma consulta ou uma resposta (*QR*), se é uma resposta com autoridade (*AA*), se é uma mensagem truncada (*TC*), se deseja recursão (*RD*), se há recursão disponível (*RA*), o tipo da consulta (descrito pelo campo *OPCODE*), e o código da resposta (*no error, format terror, server error*) no campo *RCODE*. Também indica a quantidade de consulta, a quantidade de respostas (caso tenha), a quantidade de servidores de nomes retornados na resposta e a quantidade de registros de recurso (*RR*).
- **Consulta** – contém os parâmetros da consulta (nome, tipo e etc).
- **Resposta** – contém os *RR*'s que respondem diretamente à consulta.
- **Autoridade** – contém os *RR*'s que descrevem outros servidores com autoridade.
- **Adicional** – contém os *RR*'s que não são explicitamente solicitados, mas podem ser úteis como, por exemplo, os endereços IP dos servidores de nome retornados no campo de autoridade.

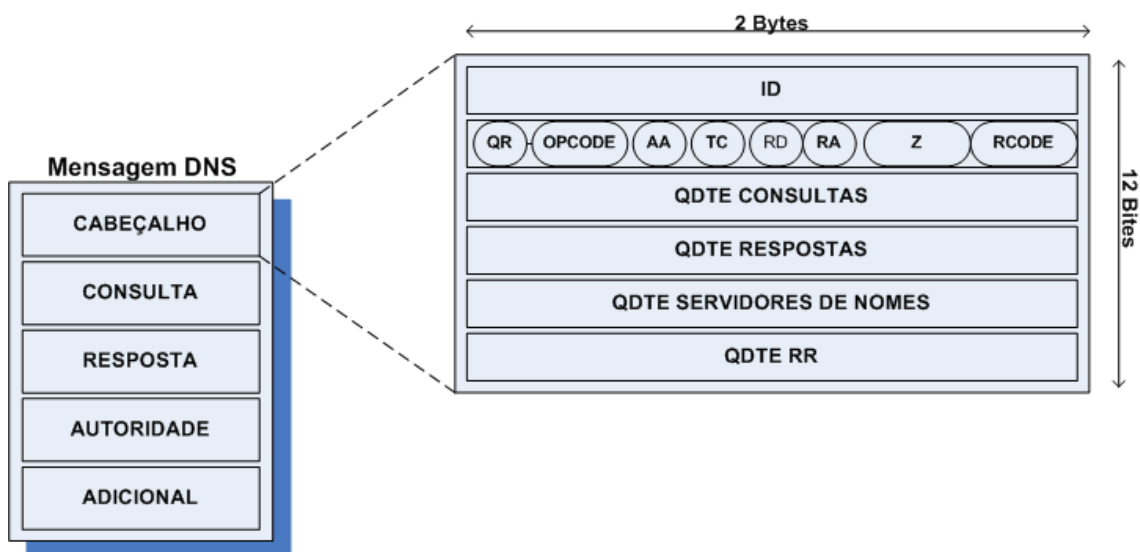


Figura 2.5 - Formato da mensagem DNS

A Figura 2.5 ilustra como é o formato de uma mensagem DNS. Quando a mensagem é de consulta os demais campos ficam vazios. Quando a mensagem é de resposta o campo de questão é repetido e o campo de resposta contém os RR's correspondentes à resposta. Caso necessário os campos de autoridade e adicional são utilizados. Mais detalhes dos campos do cabeçalho e das *flags* podem ser encontradas no Apêndice A sobre os parâmetros do DNS.

Uma vez que os registros de recursos (RR) são a unidade padrão das informações disponibilizadas e transmitidas pelo servidor de nomes, possuem uma organização padrão (figura 2.6). Estão dispostas informações como o nome (*name*) da consulta, o tipo do RR (*type*), a classe da informação (*class*), o tempo de vida de tal informação (TTL), o tamanho do campo de dados (*rdlength*) e o campo de dados (*rdata*) contendo a informação de retorno, caso seja o RR de uma mensagem de resposta, ou vazio caso seja o RR de uma mensagem de consulta.

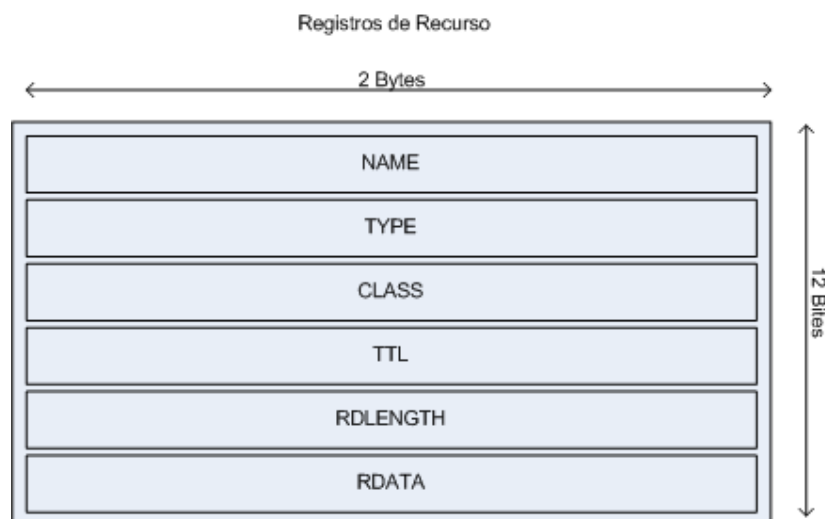


Figura 2.6 - Formato dos registros de recurso

2.2 Anomalias

O crescimento explosivo da Internet tem proporcionado maior acessibilidade a uma enorme quantidade de dados valiosos, tornando, assim, o papel do DNS ainda mais importante. Entretanto, muitas vulnerabilidades estão expostas e o número de incidentes aumenta ao longo do tempo, sobretudo as tentativas maliciosas recentes, cada vez mais ousadas e destinadas a obter benefícios financeiros através da grande quantidade de máquinas comprometidas na rede.

Portanto, o serviço de nomes da Internet está suscetível a quase todos os tipos de combinações ataques existentes na web. Devido à criticidade da informação fornecida pelo DNS, esses ataques não passam despercebidos pela rede, criando, quase sempre, características anômalas no tráfego ou no protocolo DNS.

Por definição, uma anomalia [35] é todo e qualquer comportamento fora da normalidade que apareça no tráfego. Dado que todo o funcionamento do protocolo seja bem conhecido, é possível determinar comportamentos e ações que não correspondem ao comum. Estes comportamentos podem acontecer por vários motivos: má configuração do servidor de nomes, acidentes no tráfego da Internet e, o mais perigoso, ações maliciosas.

Recentemente vários casos de anomalias e vulnerabilidades no protocolo DNS foram identificados e foram remediados. O caso mais comentado foi à vulnerabilidade [11] encontrada por Kaminsky [6] que utilizava um ataque já conhecido (*Cache Poisoning*) contra servidores DNS, para explorar uma nova vulnerabilidade de implementação encontrada em um dos softwares mais usados, o BIND [19] .

Vale lembrar que servidores DNS já foram vítimas desse mesmo tipo de ataque há certo tempo [5] também, por causa de uma vulnerabilidade na implementação já corrigida, o que prova quão bem-conhecido é este ataque. Dan Kaminsky percebeu que a implementação do BIND possuía uma brecha na randomização das portas de comunicação feitas durante a comunicação do protocolo DNS. Tal brecha tornava a randomização de portas fraca, o que aumentava a chance de “adivinhar” o ID da transação durante a tentativa de roubar a identidade do servidor de nomes [3] . O *Spoofing*, ou falsa personificação, de um servidor de nomes é o gargalo de toda a operação de *Cache Poisoning*, tornando todo o restante da operação trivial.

Nos próximos tópicos serão descritos as anomalias mais comumente conhecidas e encontradas na Internet.

2.2.1 Fast-Flux Domains

Domínios *Fast-Flux* [40] são domínios que têm como característica mudar rapidamente os dados dos registros de recurso e, por este motivo, tipicamente apresentam um TTL baixo. *Fast Flux Service Network* (FFSN) são redes, normalmente compostas por máquinas comprometidas, que hospedam e oferecem serviços para domínios *Fast-Flux*. Dessa forma, um domínio *Fast-Flux* pode ter múltiplos (centenas ou até milhares) de endereços IP associados a ele, que serão trocados em uma frequência muito alta e, por conseguinte, com

TTL baixo (minutos ou até segundos). A Tabela 2.1 exemplifica registros de *Fast Flux Domains*.

Tabela 2.1 - Exemplo de registros DNS de *Fast Flux Domains*

Nome	Endereço	Tipo do registro	TTL (segundos)
614a3p875.com.	4.225.174.243	1	180
614a3p875.com.	24.0.250.74	1	180
614a3p875.com.	24.208.243.227	1	180
614a3p875.com.	41.244.113.69	1	180
614a3p875.com.	71.56.128.14	1	180
614a3p875.com.	71.57.171.189	1	180
614a3p875.com.	76.20.166.51	1	180
614a3p875.com.	83.227.51.100	1	180
614a3p875.com.	97.81.205.120	1	180
614a3p875.com.	128.253.141.133	1	180

Essa técnica de evasão, o *Fast Flux Domains*, é muito utilizada por atacantes e criminosos para dificultar a identificação, rastreamento e o combate a domínios utilizados para propósitos ilegais. Fatos recentes [2] sinalizam um crescente uso dessa técnica por servidores de Comando e Controle (C&C) para controlar *bots* e redes zumbis¹ [40], como percebido no caso do *Storm Worm* [10] e do *myspace* [9].

Devido ao crescente uso e ao potencial dano à estrutura da Internet, domínios *Fast-Flux* têm sido estudados e vários métodos e soluções estão sendo propostas. A ICANN (*Internet Corporation Assigned Names and Numbers*) publicou um relatório [32] descrevendo os aspectos técnicos dos FFSN's. Também existe um draft [30] da IETF (*Internet Engineering Task Force*) que sugere novos procedimentos para implementações do DNS de forma a prevenir abusos com o uso dessa técnica.

2.2.1.1 Anatomia do *Fast Flux Domain*

O funcionamento e arquitetura de ataques que utilizam domínios *Fast-Flux* é ilustrada na Figura 2.7 e explicada em seguida.

¹ Redes zumbis fazem parte de uma arquitetura utilizada pelos *worms* onde as máquinas infectadas se comunicam com a central de comando e controle para posteriormente obter informações.

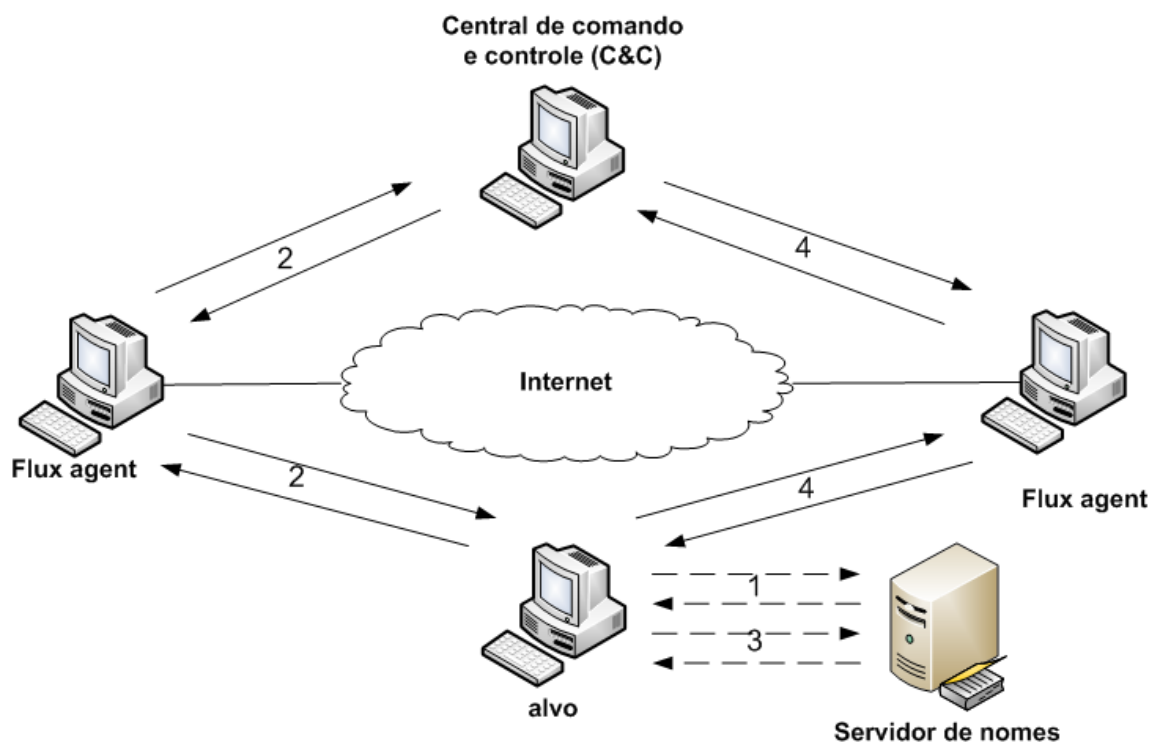


Figura 2.7 - Arquitetura de Ataque utilizando *Fast Flux Domains*

1. A máquina alvo faz uma consulta ao servidor DNS requisitando o endereço de um determinado domínio malicioso. Existe várias formas de fazer a máquina alvo acessar o domínio malicioso, tudo depende do objetivo do atacante. Normalmente são usadas técnicas de *phishing*, onde o atacante constrói um *website* falso, muito semelhante ao legítimo, fazendo com que a vítima o acesse pensando estar acessando o *site* original. O uso de *SPAM's* com *links* para tais domínios também é bastante empregado.
2. A vítima então acessa o endereço respondido que está hospedado em uma das máquinas corrompidas pelo atacante, chamada *flux agent*. Esse agente atua como um *proxy* de redirecionamento das requisições que chegam para o servidor central.
3. Após um tempo TTL_{0+1} a vítima volta a solicitar o endereço do domínio malicioso ao servidor de nomes. Desta vez, um novo conjunto de endereços é retornado, direcionando a vítima para outra máquina comprometida.
4. A vítima volta a acessar o domínio malicioso, acessando um endereço diferente do anterior, porém com conteúdo e serviços iguais ao primeiro.

Redes *Fast-Flux* podem ser facilmente confundidas com técnicas utilizadas por servidores legítimos, como *Round Robin DNS* (RRDNS) e redes de distribuição de conteúdo (CDN - *Content Delivery Network*), visto que seus requisitos de disponibilidade e controle de carga se assemelham aos de FFSN's.

2.2.1.2 *Round Robin DNS*

Round Robin DNS [38] é uma técnica de distribuição e balanceamento de carga redundante dos serviços existentes na Internet, como servidores *Web* e FTP, que utilizam o gerenciamento das respostas DNS para os endereços requisitados pelos clientes. Funciona respondendo não somente com um endereço IP, mas sim com uma lista de endereços de vários servidores que hospedam o mesmo serviço. A ordem em que os endereços são distribuídos na lista é o que determina o termo *Round Robin*. Em cada resposta ao cliente o endereço IP é permutado.

2.2.1.3 *CDN (Content Delivery Network)*

CDN [12] é um sistema de redes de computadores interligados através da Internet que cooperam entre si para entregar conteúdo freqüentemente e assim melhorar questões como desempenho, escalabilidade e custo para os usuários finais. Baseia-se no fato de que a soma da capacidade de servidores estrategicamente localizados pode ser maior do que a capacidade do *backbone* da rede. Isso pode resultar em um aumento no número de usuários usando concorrentemente o mesmo sistema. .

Por exemplo, em um *backbone* de rede de 10 Gbit/s onde a capacidade do servidor central é de 100 Gbit/s, somente 10 Gbit/s podem ser utilizados. Mas quando 10 servidores são movidos para 10 localizações extremas, a capacidade pode ser $10 * 10$ Gbit/s. Os CDN's comerciais mais conhecidos são Akamai Technologies Inc². e Limelight Networks³.

2.2.2 **Typo-Squatter Domains**

Typo-squatting, também chamada de seqüestro de URL, faz parte de uma técnica de *phishing* [8] onde o atacante tira proveito de erros cometidos pelo usuário ao colocar URL's incorretas nos seus navegadores, seja por desconhecimento do real endereço, seja por erro de digitação ou por enganos na pronúncia de tal endereço. Desta forma, atacantes podem hospedar domínios que ainda não foram alocados, mas que têm a grafia semelhante a algum outro endereço na Internet de grande acesso, mudando apenas uma letra, ou um subdomínio (de

² <http://www.akamai.com/>

³ <http://www.limelightnetworks.com/>

.com para .net ou .org, por exemplo) e esperar que os usuários acessem este endereço falso, que são extremamente semelhantes ao *site* originalmente desejado.

Esse tipo de ação, conhecida por ser uma forma de *cybersquatting* [4] , também é muito comum para a transferência de vírus, *worms*, *adwares* e *spywares*, além de também ser utilizado como técnicas de *domain parking* [7] (prática de reservar certos domínios para a sua posterior negociação ou para o uso de programas de publicidade, como o Google AdSense). A Tabela 2.2 apresenta alguns exemplos de *typo squatting*.

Tabela 2.2 - Tipos de erros de domínios

Tipo do erro	Exemplo
Pronúncia	www.sin.ufpe.br
Digitação	www.con.ufpe.br
Hierarquia de domínio	www.cin.ufpe.br.com

Bojan [17] identificou que muitos dos *typo squatting domain* encontrados estão hospedados em um único endereço IP, mantido pelo atacante, que contém vários anúncios e propagandas. Também foi percebido que, por medida de precaução de alguns administradores do DNS, uma parte dos *typo squatting domain* encontrados fazem uso de registros DNS do tipo *wildcard*. Uma zona DNS *wildcard* permite ao administrador configurar a resolução de qualquer consulta nessa zona para uma resposta padrão. Por exemplo, um *wildcard* “*.cin.ufpe.br” irá retornar a mesma resposta para qualquer consulta feita ao *host* ou a um subdomínio no domínio *cin.ufpe.br*.

Esse problema não é recente e pesquisas já vêm sendo feitas no intuito de criar ferramentas e técnicas que possam identificar possíveis URL’s, com grau de semelhança grande a outras URL’s bem conhecidas [45] .

2.2.3 Uso de Endereços Privados

De acordo com a RFC 1918 [44] existem espaços de endereçamento que não podem ser usados externamente, ou seja, não podem ser roteados e somente devem ser utilizados internamente, dentro do domínio da rede. Desta forma, uma característica que pode ser considerada anômala no funcionamento do DNS são respostas contendo endereços não roteáveis ou endereços que não tenham nenhum domínio alocado na Internet.

2.2.3.1 DNS Rebind

Essa anomalia consiste em iludir o navegador de uma determinada máquina de uma rede local, fazendo-o executar arbitrariamente *scripts* maliciosos contra outras máquinas da mesma rede [21] .

Basicamente o ataque faz uso do processo de resolução de nomes onde o atacante registra um domínio que é delegado para um servidor DNS que ele controla. O servidor é configurado para responder com um TTL muito baixo, o que previne que a resposta seja mantida na *cache* por muito tempo. A primeira resposta contém o endereço IP do servidor que está hospedando o código malicioso. Respostas subseqüentes conterão endereços IP falsificados de rede privada, presumivelmente atrás de um *firewall*, sendo o alvo do ataque.

A Figura 2.8 ilustra e exemplifica o processo de ataque usando DNS *rebind*:

1. Uma máquina corrompida de dentro da rede-alvo solicita um domínio do servidor de nomes malicioso mantido pelo atacante. O servidor responde com um endereço arbitrário, porém com um valor de TTL baixo.
2. A máquina então faz o acesso ao serviço desejado normalmente.
3. Após o tempo TTL_{0+1} a máquina corrompida então faz outra solicitação de endereço para o domínio. Desta vez o servidor de nomes retorna um endereço de uma máquina dentro da rede-alvo (endereço privado), juntamente com uma rotina maliciosa a ser executada pelo navegador da máquina que está servindo de porta de entrada para o ataque.
4. A máquina corrompida então acessa o domínio, porém enviando as requisições para uma máquina que se encontra dentro da rede e que será o alvo do ataque, através da execução automática da rotina maliciosa que está na máquina corrompida. Como a máquina alvo está na mesma rede da máquina que está atacando então, os mecanismos de segurança da rede não podem impedir o ataque.

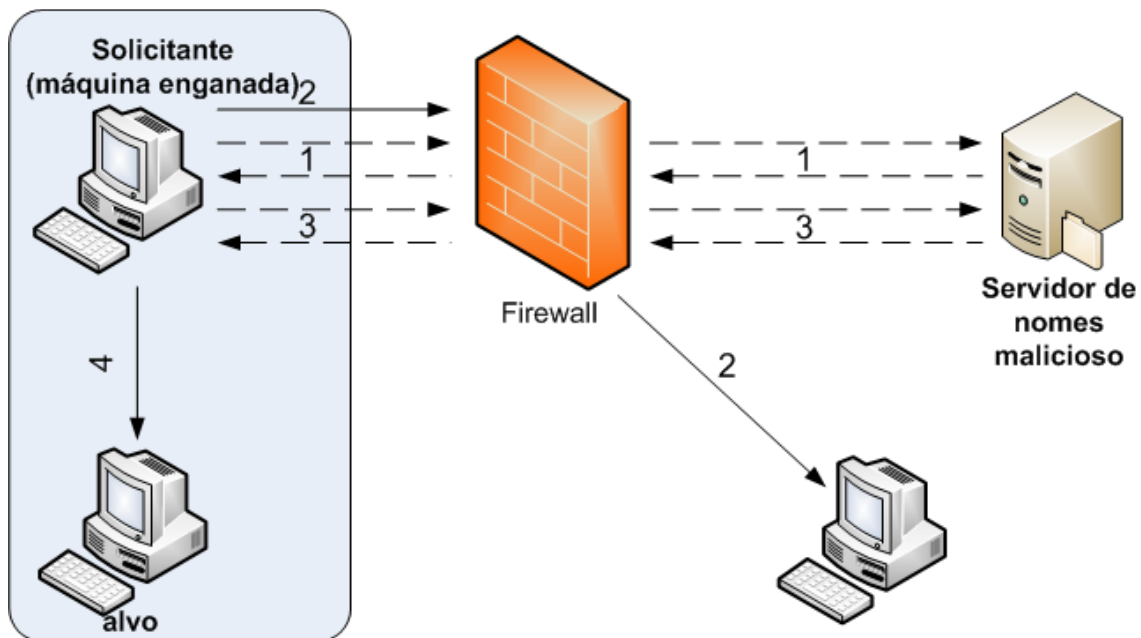


Figura 2.8 - Arquitetura do Ataque DNS Rebind

2.2.3.2 Darknets

Também é possível, através do monitoramento do DNS *Rebind*, identificar redes que utilizam faixas de endereçamento IP indevidas ou supostamente não alocadas, chamadas **darknets** [34]. Qualquer pacote que é originário de uma *darknet* pode ser considerado anômalo. Nenhum pacote legítimo deveria ser enviado para uma *darknet*. Tais pacotes podem ter chegado por erro ou má-configuração, mas a maioria desses pacotes é enviada por *malwares*, buscando vulnerabilidades na rede.

Esse tipo de rede pode ser utilizado de forma maliciosa na criação de redes zumbi, mais especificamente, para atuar em tentativas de ataques DDoS (*Distributed Denial of Service*). O seu monitoramento pode aumentar a ciência e facilitar a mitigação dos ataques, tornando mais fácil determinar a quantidade de tráfego malicioso na rede e a sua origem.

3 Trabalhos Relacionados

Este capítulo traz uma breve análise das soluções mais relevantes nas áreas de detecção e prevenção de anomalias (ataques e/ou ações maliciosas) no protocolo DNS. Também analisa alguns estudos sobre o monitoramento do tráfego de rede focados na busca de informações importantes na identificação de possíveis ataques e deficiências contra esse protocolo.

3.1 Trabalhos propostos

Desde sua idealização até os dias atuais, o DNS vem ganhando importância à medida que a Internet cresce em número de usuários e serviços. Porém, como não foi projetado nem implementado levando em consideração mecanismos de segurança e/ou robustez, existem diversas vulnerabilidades (inerentes à especificação, implementação e até configuração do protocolo) que podem levá-lo a falhas.

O primeiro trabalho a mostrar as fraquezas do protocolo foi feito por Schuba [22], onde foram exemplificadas como algumas características do protocolo DNS podem ser exploradas e utilizadas para abrir caminhos para outros tipos de ataques como *spoofing* ou DDoS. Diversos cenários foram montados e simulados, expondo, de forma clara, as potenciais fraquezas do protocolo, quais situações favorecem a exploração dessas fraquezas e de que forma elas podem ser utilizadas.

Seguindo a mesma linha de raciocínio, Bellovin [18] mostra, na prática, os ataques que podem e foram feitos ao protocolo DNS, além de explicar como executar tais ataques.

Na tentativa de prover segurança foi criado o DNSSEC [29] (*DNS Security Extension*) que adiciona autenticação e integridade de dados ao protocolo DNS. Esta extensão permite ao usuário verificar, criptograficamente, se uma resposta validada é realmente o dado configurado no servidor de nomes autoritário. Uma característica importante de tal extensão é que ela não modifica o protocolo DNS, apenas especifica campos RR's adicionais para chaves criptográficas e assinaturas digitais, que não restringem resolvers antigos de continuarem funcionando, permitindo, assim, que a segurança seja inserida incrementalmente e não requer que todos na Internet passem a usar uma nova versão do protocolo DNS.

O DNSSEC consegue prevenir uma quantidade razoável de problemas e ataques ao protocolo DNS, como respostas forjadas, ataques de *phishing*, *man-in-the-middle* e redirecionamento para um endereço malicioso. Entretanto, a segurança provida por esta

extensão não é completa, além de não garantir segurança, contra eventos como má configuração ou informações incorretas no servidor. O DNSSEC ainda traz consigo uma série de vulnerabilidades [15] que permitem ataques ainda mais prejudiciais como *buffer overrun* e ataques DDoS. Algumas questões relacionadas ao gerenciamento da chave criptográfica como a configuração da chave inicial, autenticação e verificação da mesma ainda precisam ser resolvidas a nível operacional, para permitir o DNSSEC ser utilizado em escala mundial.

Atualmente, outra linha de pesquisa relacionada ao DNS vem ganhando bastante espaço: o monitoramento e observação constante dos servidores de nomes da Internet. Essa ação é considerada interessante porque visa manter o serviço DNS da Internet em funcionamento contínuo, uma vez que a avaliação do comportamento do tráfego DNS, em determinados pontos da Internet, podem indicar vários sintomas de anormalidades.

Wessels [23] acompanhou o fluxo de consultas que chegavam a um dos treze servidores de nomes raiz do DNS durante um período de 24 horas. Os dados coletados foram analisados usando um modelo simples do DNS e as consultas foram classificadas em nove categorias, detalhadas na Tabela 3.1. O resultado mostrou que muitas das consultas são repetidas e que somente uma pequena porcentagem é legítima.

Tabela 3.1 - Discriminação das categorias de consultas

Categorias	Discriminação
Consulta desconhecida	Consultas com valores para o campo <i>class</i> diferente das cinco definidas pelo DNS
<i>A</i> para <i>A</i>	Consultas do tipo <i>A</i> para nomes que já estão na forma numérica
<i>TLD</i> desconhecido	Consultas para nomes não conhecidos como <i>Top Level Domains</i>
Caracteres irregulares	Consultas que possuem caracteres inválidos no nome
RFC 1918 no PTR	Consultas do tipo PTR para endereços dentro do espaço de endereçamento especificado pela RFC 1918
Consultas idênticas	Consultas idênticas, inclusive o ID, enviadas pela mesma fonte em um curto espaço de tempo
Consultas repetidas	Semelhante às <i>consultas idênticas</i> , porém com o valor do ID diferente
Orientação não armazenada	Consultas sequenciais feitas iterativamente, onde ao subir um nível na hierarquia de servidores a consulta se diferencia
Legítimos	São as consultas que não se enquadram em nenhuma das categorias supracitadas

Sánchez [24] fez uma análise mais detalhada das consultas “falsas” que chegam ao servidor de nomes raiz, através da caracterização mais sucinta das categorias as quais elas se enquadram, levantando estatísticas e determinando possíveis causas e soluções. Diferentemente das duas últimas abordagens, citadas anteriormente, que analisa as consultas recebidas por um servidor de nomes, o trabalho de Sánchez propõe um estudo das respostas trafegadas pela rede.

Nesta mesma linha de abordagem, Weimer [31] projetou uma arquitetura de captura de pacotes do protocolo DNS (figura 3.1) que utilizou sensores localizados em pontos estratégicos da Internet e capturou respostas de consultas DNS provenientes de clientes web, por um determinado tempo. Como resultado desse projeto, o software *dnslogger* foi desenvolvido e um *website* foi construído para permitir ao público acesso aos dados coletados do DNS, para posterior referência.

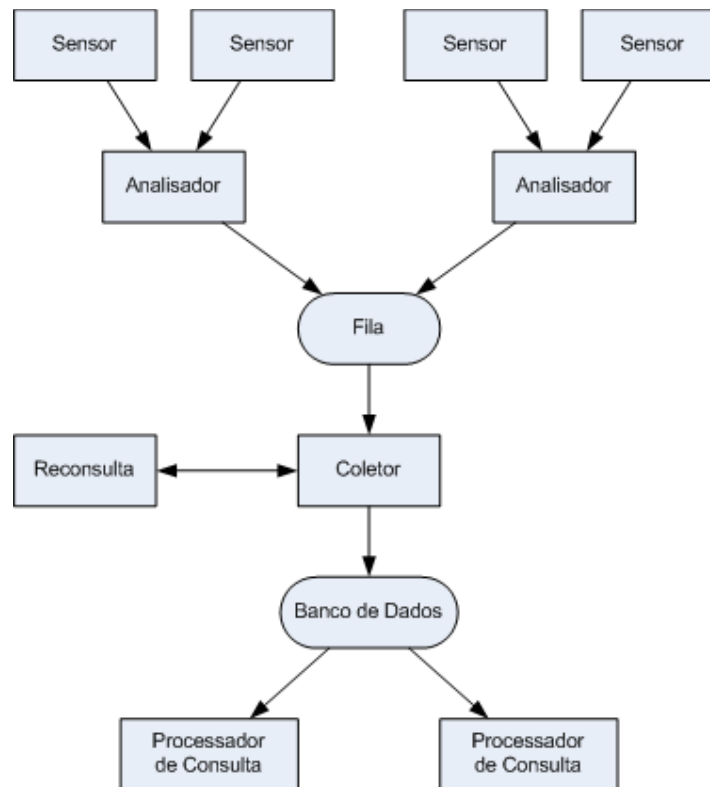


Figura 3.1 - Arquitetura de software do *dnslogger*

Kristoff [33] , no intuito de identificar e acompanhar máquinas infectadas por *bots* e *worms*, criou uma ferramenta denominada DNSwatch. Esse *software* funciona comparando as respostas das consultas a uma lista (*black list*) com domínios previamente conhecidos como

maliciosos. Essa mesma necessidade foi a motivação para o trabalho de Schonewille [13] que, utilizando uma arquitetura semelhante à utilizada por Weimer, capturou pacotes do tráfego da Universidade de Amsterdam, na tentativa de identificar máquinas comprometidas com *bots* e/ou tráfegos anômalos. Este último foi o primeiro trabalho a encarar o DNS como um IDS (*Intrusion Detection System*).

Bojan [20] fez um estudo parecido na Universidade de Auckland, onde monitorou passivamente o tráfego da rede por um determinado tempo. Foram identificados novos tipos de anomalias até então não encontradas no tráfego como *Typo Squatter* e *Fast Flux Domains*. Também foi avaliado o impacto e a dependência dos sistemas anti-spam instalados na Universidade.

3.2 Ferramentas

Esta seção apresenta algumas ferramentas empregadas na verificação do comportamento do DNS. Uma análise de suas características, assim como suas vantagens e desvantagens serão discutidas.

3.2.1 DNSwatch

O *DNSwatch* é um *script* que monitora o comportamento de alguns domínios contidos em uma lista, ditos maliciosos, podendo assim identificar possíveis máquinas infectadas dentro da rede. À medida que novos domínios maliciosos são identificados a lista é atualizada.

Este *script* possui a desvantagem de atuar de maneira muito restrita por, apenas, acompanhar o comportamento de alguns domínios maliciosos, alertando a atuação desses domínios na rede e não necessariamente observando o comportamento do protocolo DNS, como é a proposta deste trabalho de graduação. Além disso, precisa de uma *black list* conhecida para que funcione da maneira esperada.

3.2.2 DNSlogger

Aplicação desenvolvida com o intuito de registrar todos os pacotes DNS que chegam a uma determinada rede, sem alterar o tráfego e disponibilizando os resultados para posteriores consultas. Este *software* surgiu da primeira arquitetura de captura passiva a utilizar sensores em pontos estratégicos na rede. Porém esse *software* utiliza uma base muito simples, armazena pouca informação e não faz nenhuma inferência ou classificação em cima dos dados coletados, apenas registra os dados coletados para posterior consulta.

3.2.3 honeyDNS

Oberheide [34] , em 2007, introduziu o conceito de *dark* DNS (campos e domínios DNS associados à *darknets*). Na tentativa de evitar possíveis ameaças, desenvolveu um sensor capaz de responder a consultas reversas (consultas do tipo PTR) feitas a grandes *darknet's* com respostas apropriadas e, assim, rastrear e monitorar as consultas feitas a esses endereços.

Este sensor atua passivamente sobre o tráfego da rede, não coleta ou analisa o tráfego, apenas responde a requisições a *darknet's* previamente estabelecidas, atuando como um *honeypot* e identificando tentativas de acesso a redes, a princípio, não roteáveis.

3.2.4 DNStop e DNScap

DNStop [41] é uma ferramenta para visualizar várias informações do tráfego DNS da rede que está sendo capturado como tipos de consultas, tipos de respostas TLDs acessados e etc.

De forma semelhante, o DNScap [26] é uma ferramenta de captura de pacotes DNS também baseada na biblioteca libpcap, porém, inclui mais opções e integração com outras ferramentas, como os formatos de saída do DiG. Foi desenvolvido e é mantida pelo DNS-OARC [28] , uma entidade que reúne os principais pesquisadores na área no intuito de compartilhar informações e coordenar respostas a ataques ao DNS.

3.2.5 DSC

DSC [42] é um sistema para coletar e explorar estatísticas de servidores de nomes DNS. Pode ser dividido em dois componentes principais: o *Collector*, que armazena os pacotes DNS recebidos pela interface de rede que está sendo capturada, e o *Presenter*, que apresenta os dados capturados, no formato XML, através de um servidor web.

Esta ferramenta é superior ao **dnstop** e **dnscap**, uma vez que além de coletar dados, levanta estatísticas sobre os dados e as representa através de gráficos e figuras.

4 Detecção de Anomalias no Protocolo DNS

Este capítulo descreve a construção da ferramenta para detecção DNS *Fast Flux Domains*. Primeiro, será discutido o algoritmo e as métricas utilizadas. Em seguida, aspectos referentes a implementação da ferramenta serão apresentados. Por fim, o processo de execução será demonstrado.

4.1 Algoritmo

O algoritmo utilizado para detecção de *Fast Flux Domains* foi proposto e descrito por Holz [39] e baseia-se na extração de características que definem como um CDN ou *Round Robin* DNS (RRDNS) se comportam.

Antes de explicar o algoritmo propriamente dito, é necessário discutir duas restrições que ajudam a definir e identificar mais claramente *Fast Flux Service Network* (FFSN). São elas:

- **Diversidade de endereços IP** - uma FFSN depende de máquinas corrompidas ou comprometidas para executar o *flux-agent*. Sendo assim, existe uma natural diversidade de endereços IP e quase sempre o atacante não tem a possibilidade de escolher um determinado endereço IP.
- **Falta de controle físico sobre o *flux-agent*** - normalmente as máquinas que fazem parte de uma FFSN possuem conexão limitada com a internet ou mesmo uma limitação de recursos computacionais podendo, desta forma, sair da rede a qualquer momento, sem qualquer garantia para o atacante.

Baseado nessas restrições, o algoritmo selecionado enumera um conjunto de parâmetros que podem ser usados para distinguir comportamentos de tráfego DNS benignos dos maliciosos.

1. A ausência de controle físico dos *flux-agent* pode ser medida segundo dois parâmetros. O primeiro é o n_A (número de registros do tipo A retornados para todas as consultas de um domínio). Domínios legítimos comumente retornam entre um e três registros, enquanto que domínios *Fast-Flux* freqüentemente retornam cinco ou mais registros, em uma única consulta, para garantir que pelo menos um dos endereços IP esteja *online*. O segundo é o n_{NS} (número de registros de servidores de nome em uma única consulta). FFSN normalmente retornam vários

registros NS e registros A para os registros NS. Já domínios legítimos retornam poucos registros NS.

2. A restrição na diversidade de endereços IP resulta na consideração do parâmetro n_{ASN} , o número de ASN únicos para todos os registros do tipo A. ASN (Autonomous System Number) são identificadores únicos das redes contidas dentro da Internet. Domínios legítimos tendem a retornar endereços IP de um AS específico, enquanto que redes *Fast-Flux* tendem a estarem localizadas em diferentes AS, uma vez que as máquinas infectadas estão espalhadas através de diferentes redes.

Outro parâmetro importante, porém não essencial, é o *Time-To-Live* (TTL). Seu valor não pode ser considerado como um bom parâmetro, pois domínios legítimos hospedados em CDN possuem requisitos similares aos de FFSN. Entretanto o seu valor pode ser utilizado para distinguir CDNs e FFSN de RRDNS. De modo geral, TTL com valores menores que 1800 segundos representam os dois primeiros enquanto que valores maiores de 1800 segundos representam RRDNS e não podem ser considerados “rápidos” o suficiente para mudanças repentinas.

Com base nesses parâmetros é possível definir duas métricas para avaliar a existência ou não de *Fast Flux Domains* no tráfego DNS: *Fluxiness* e *Flux-score*.

4.1.1 *Fluxiness*

Esta métrica é capaz de distinguir FFSN e CDN em função de n_A , n_{ASN} e n_{NS} . Uma aproximação para essa função é definida na Equação 1.

$$\phi = n_A / n_{single} \quad (\text{Equação 1})$$

Onde o parâmetro n_{single} representa a quantidade de registros A retornada em uma única consulta.

O resultado da Equação 1 é avaliado da seguinte forma. Quando ϕ tem valor 1, significa que o conjunto de registros A permanece constante após consecutivas consultas, o que indica a existência de um domínio legítimo usando o mesmo conjunto de endereços IP. Entretanto, valores de ϕ maiores que 1 indicam que pelo menos um novo registro A foi encontrado após consecutivas consultas, o que vem a ser uma indicação de CDN e FFSN.

4.1.2 Flux-Score

Esta métrica para detecção de domínios *Fast-Flux* pode ser derivada dos parâmetros observados como um vetor x da forma (n_A, n_{ASN}) e usando um vetor de peso w para diferenciar a importância de cada parâmetro no valor final. Assim o *flux-score* é dado por:

$$f(x) = w^T x = w_1 \cdot n_A + w_2 \cdot n_{ASN}$$

Desta forma, o *flux-score* provê uma classificação de domínios. Comparados com um termo limitante b , um valor alto reflete um alto grau de características *Fast-Flux*, enquanto que valores menores correspondem a domínios benignos. Medidas empíricas indicaram os melhores valores de peso dos parâmetros w^T e do limitante b [39] :

$$flux_A(n_A, n_{ASN}) = 1.32 \cdot n_A + 18.54 \cdot n_{ASN}$$

com $b = 142.38$

4.2 Implementação

A solução implementada para detecção de Fast Flux Domains é estruturada em três módulos de controle: Captura, Base e Análise. A figura 4.1 mostra a arquitetura da solução proposta neste trabalho de graduação.

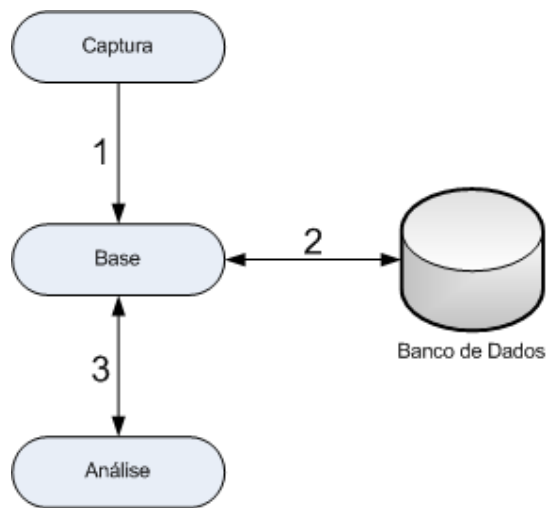


Figura 4.1 - Arquitetura do software desenvolvido

4.2.1 Captura

O módulo de **Captura** é responsável pela captura do tráfego da rede e extração dos dados (pacotes) relacionados ao DNS do *payload* dos pacotes UDP. Usando a proposta de [17] como base, é aplicado um filtro de captura visando obter somente as respostas às consultas DNS, vindas de um servidor autoritário. O filtro utilizado é exemplificado abaixo:

```
udp port 53 and ( udp[10] & 0x04 != 0 )
```

Cada pacote DNS capturado tem suas informações relevantes (aquelas que possam ser usadas para detectar anomalias) extraídas e encaminhadas ao módulo de armazenamento (**Base**) para ser persistida na base de dados criada, veja Tabela 4.1. As informações consideradas relevantes são:

- a. *Nome da consulta (query)*
- b. *Tipo da consulta (type)*
- c. *Resposta (answer)*
- d. *Time-To-Live (TTL)*
- e. *Hora da captura do pacote (timestamp)*

Tabela 4.1 - Exemplo de registros persistidos na base

<i>Query</i>	<i>type</i>	<i>answer</i>	<i>TTL</i>	<i>Timestamp</i>
www.cin.ufpe.br.	2	<i>exu.cin.ufpe.br.</i>	3600	1224251188
www.cin.ufpe.br.	5	<i>jaboatão.cin.ufpe.br.</i>	3600	1224251188
www.cin.ufpe.br.	1	150.161.2.9	3600	1224251188

É importante ressaltar que o módulo de Captura pode ser encarado como um sensor de tráfego e pode, desta forma, ser naturalmente distribuído em diferentes pontos da rede onde existam servidores DNS. Atualmente, opera de forma *off-line*, entretanto não existe restrição para operar de forma *on-line*.

4.2.2 Base

O módulo **Base** controla todo o tipo de acesso à base de dados, seja para consulta ou inserção. Para tanto, foi utilizada uma base de dados MySQL devido ao suporte a grandes consultas, linguagem de scripts SQL e de ser *open source*.

4.2.3 Análise

O módulo de Análise é o elemento principal do esquema de detecção uma vez que fornece a análise dos dados armazenados e implementa os algoritmos de detecção de anomalias, no caso o de detecção de domínios *Fast Flux*.

4.3 Execução

Vale lembrar que o monitoramento do tráfego é *passivo* e *offline*, ou seja, a captura não interfere em momento algum no tráfego, seja inserindo ou retirando pacotes, e a análise é feita após a captura e inserção na base, desta forma o resultado não reflete, necessariamente, o tráfego real que está passando no momento da execução da análise (*on-the-fly*).

Dessa maneira, a execução pode ser dividida em duas fases independentes, ilustrada na figura 4.2: a fase de *modelagem e persistência*, onde o tráfego capturado e salvo no formato de arquivo .pcap é tratado e armazenado na base de dados, e a fase de *análise* onde será retornado para o usuário o resultado da detecção de anomalias.

Primeiramente, é aplicado um filtro no banco de dados, utilizando uma consulta SQL, para reter apenas os valores da base que possuam um TTL abaixo de um limite, passado como argumento no momento de execução desse módulo. Logo em seguida é calculado o valor do *flux-score* de cada domínio, para isso, busca-se a quantidade de respostas do tipo A distintas (utilizando outra consulta na base) e a quantidade de AS's únicos para essas respostas. Por fim, é verificado se o *flux-score* ultrapassa o limitante definido, previamente. A seguir, exemplos de consultas para retornar valores com um TTL menor ou igual a 100 ⁽¹⁾ e para retornar as respostas do tipo A para a consulta www.cin.ufpe.br ⁽²⁾.

```
SELECT Distinct query FROM dns_table WHERE ttl < 100 (1)
```

```
SELECT answer FROM dns_table WHERE query = "cin.ufpe.br" and type = 1  
(2)
```

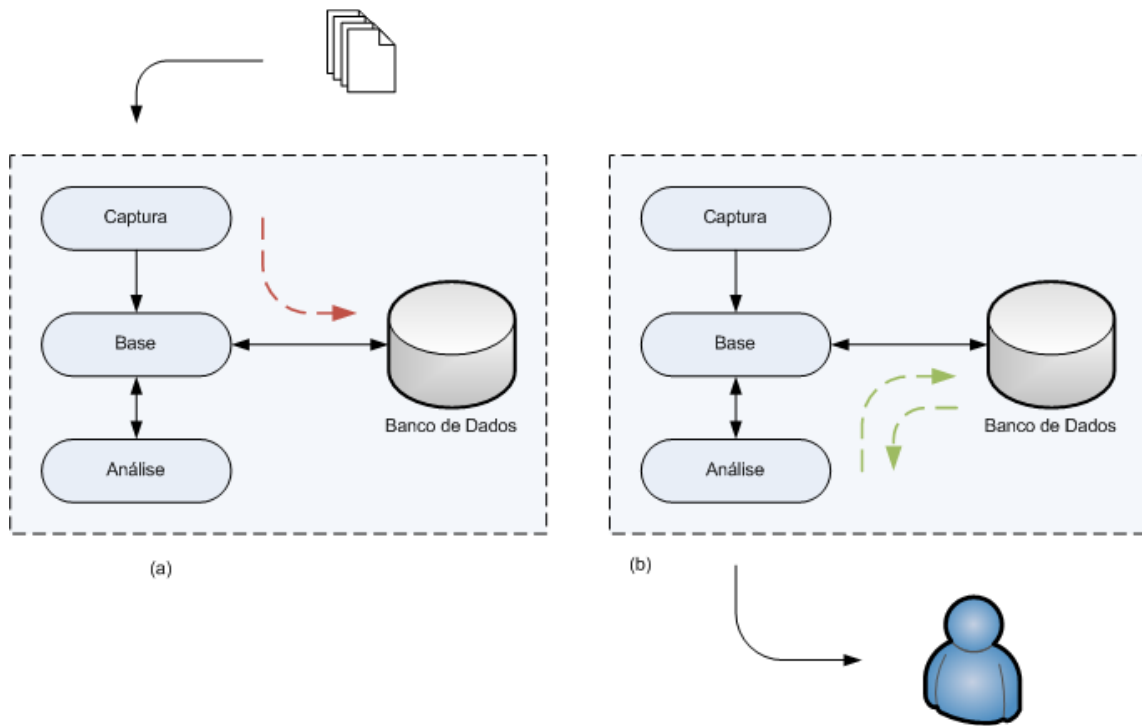


Figura 4.2 - Duas fases de execução do *software*: (a) Modelagem e Persistência e (b) Análise

5 Resultados e Discussões

Este capítulo descreve os resultados obtidos na etapa de análise dos dados, o ambiente de teste utilizado para validar o protótipo implementado e um estudo de caso realizado, nas dependências da Universidade Federal de Pernambuco (UFPE), através de dados capturados na sua rede.

5.1 Validação

Como prova de conceito do protótipo desenvolvido e no intuito de avaliá-lo e validá-lo, foram executados alguns testes de precisão. O primeiro passo foi criar um ambiente de simulação de tráfego anômalo DNS. Foi elaborado um cenário onde um *host* localizado na rede está corrompido por algum *worm* ou *bot*, de tal forma que tenta enviar requisições DNS de domínios anômalos para fora da rede.

Com a ajuda do ATLAS da Arbor *Networks* [16] foi possível obter uma lista de 10 domínios anômalos, recentemente descobertos, que são liberados diariamente. Portanto, foi criado um simulador de tráfego anômalo, baseado nessa lista de domínios, capaz de gerar requisições para esses domínios a partir de uma máquina da rede local. Desta forma, o tráfego gerado pode ser capturado com o auxílio da ferramenta de captura *Wireshark* [43]. O pseudocódigo a seguir exemplifica o simulador de tráfego anômalo DNS.

```
SimuleTrafegoAnomalo(Ld)      Ld = Lista de domínios anômalos

1 WHILE Ld não vazio:
2   FOR k = 1 to 3
3     d := proximo(Ld)         d = domínio
4     r := enviaConsulta(d)    r = resposta
5     k++
6   ENDFOR
7 ENDWHILE
```

O fato da lista disponibilizada pelo Atlas ser dinâmica e bem atualizada foi útil, pois as *black lists* encontrados na Internet não são atualizadas freqüentemente e grande parte dos domínios contidos nelas já não estão mais presentes. A Tabela 5.1 mostra a lista de domínios maliciosos obtidos no dia 24/11/2008.

Tabela 5.1 - Lista de domínios maliciosos (24/11/2008)

Domínio	Data
christizfunz.com	2008-11-22 20:54:12 EST
christinazfunz.com	2008-11-22 20:54:11 EST
christiezfunz.com	2008-11-22 20:54:10 EST
christianzfunz.com	2008-11-22 20:54:09 EST
christazfunz.com	2008-11-22 20:54:07 EST
buyonlinepharma.com	2008-11-22 20:47:09 EST
7d19i14db.com	2008-11-22 20:16:37 EST
6l4a3p875.com	2008-11-22 20:16:29 EST
4import.me	2008-11-22 14:40:14 EST
uswho.cn	2008-11-22 04:46:48 EST

Após a captura, o protótipo desenvolvido foi executado tendo como parâmetro de entrada os tráfegos diários obtidos do Atlas da Arbor Network. Os domínios maliciosos alertados pelo programa foram verificados com a lista esperada (a mesma passada como parâmetro). Esses testes de captura e execução foram realizados nos dias 22, 23 e 24 de novembro de 2008, em uma máquina nas dependências do Grupo de Pesquisa em Redes e Telecomunicações (GPRT) do Centro de Informática da UFPE.

O gráfico da figura 5.1 mostra os domínios previamente conhecidos como maliciosos utilizados para criar o tráfego anômalo. Esse tráfego foi criado através de requisições que partiram de dentro da rede para tais domínios e suas respectivas respostas DNS. De acordo com o gráfico, pode-se perceber que tais domínios foram identificados como anômalos pelo *software* como esperado.

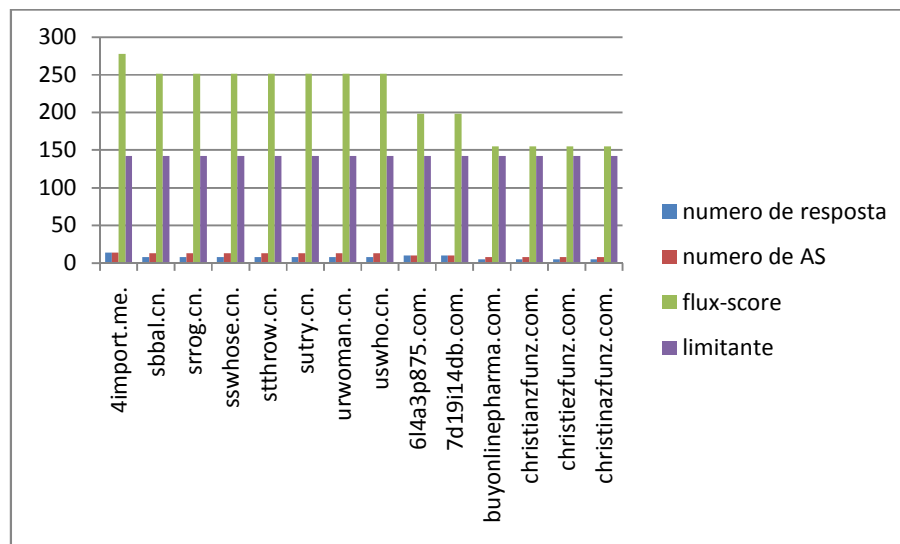


Figura 5.1 - Resultado da Análise dos domínios maliciosos

5.2 Ambiente de Teste

Uma vez que a validação da ferramenta foi realizada com sucesso, iniciou-se a fase de testes com tráfego real. Foi feita a captura do tráfego real da UFPE, durante duas semanas (do dia 17/10/2008 até o dia 02/11/2008), onde todas as informações relativas às respostas das consultas DNS foram armazenadas, para posterior análise. A captura realizada foi passiva. O ponto de captura foi o *gateway* principal do GPRT (Grupo de Pesquisas em Rede e Telecomunicações) localizado dentro da UFPE e com ligação direta ao gateway do Núcleo de Tecnologia da Informação (NTI) da própria universidade. A figura 5.2 exemplifica a topologia de captura.

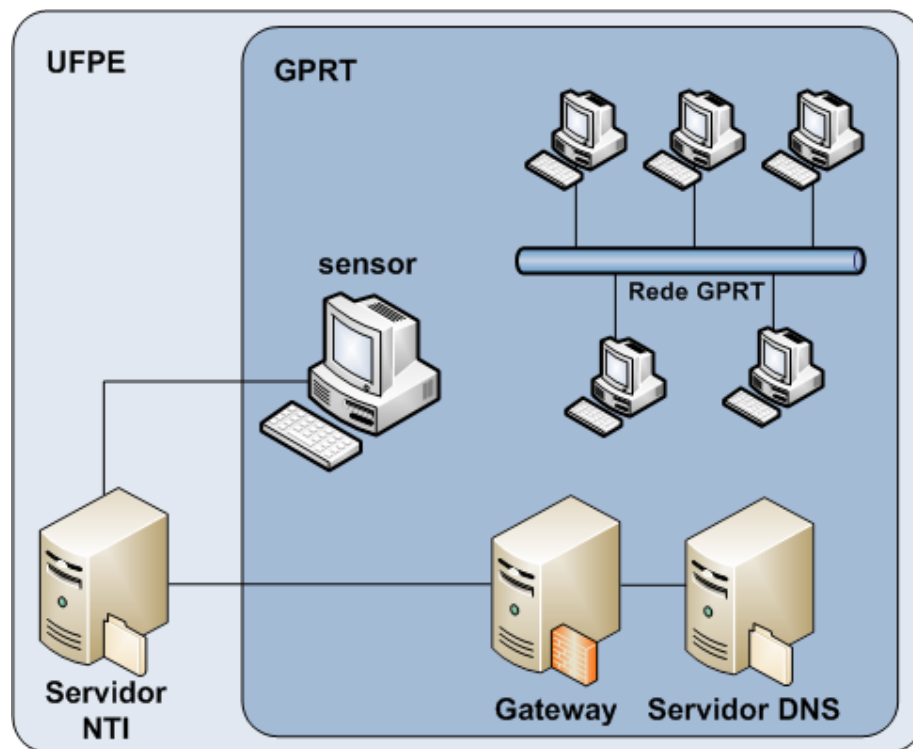


Figura 5.2 - Arquitetura do caso estudado

O tráfego obtido durante essas duas semanas somou 286 MB apenas de respostas a consultas DNS, efetuadas do servidor DNS da rede GPRT para a rede externa. Para facilitar a manipulação dos dados, o tráfego foi dividido em arquivos de 25 MB.

Seguindo o processo de execução explicado no capítulo 4, os dados capturados foram passados como parâmetro para a fase de persistência na base de dados. O software registrou a criação de aproximadamente 2,5 milhões de registros (2.578.481 registros). Na parte de identificação de anomalias, o software foi executado em um computador com processador AMD Athlon de 64 bits e 1.81 Ghz, com 1 Gb de memória RAM, executando *Windows XP SP-3* e com MySQL versão 5.0.45 como base. O tempo de processamento foi de aproximadamente 5 horas e meia.

5.3 Resultados

Para facilitar o entendimento, os resultados obtidos foram divididos em duas categorias, de modo gradativo: TTL e *Flux-Score*.

5.3.1 TTL

Uma das primeiras análises foi feita em cima dos dados utilizando apenas parâmetros passados estaticamente à execução do software, como o TTL. Dentro da faixa de TTL considerada características de Fast Flux, 1800 segundos, foram feitas consultas com diferenças de TTL de 100 segundos. Através da

Figura 5.3 pode-se visualizar a quantidade de pacotes presentes nessas faixas. Constata-se que a grande maioria dos pacotes que podem ser anômalos possui um valor de TTL de até 500 segundos.

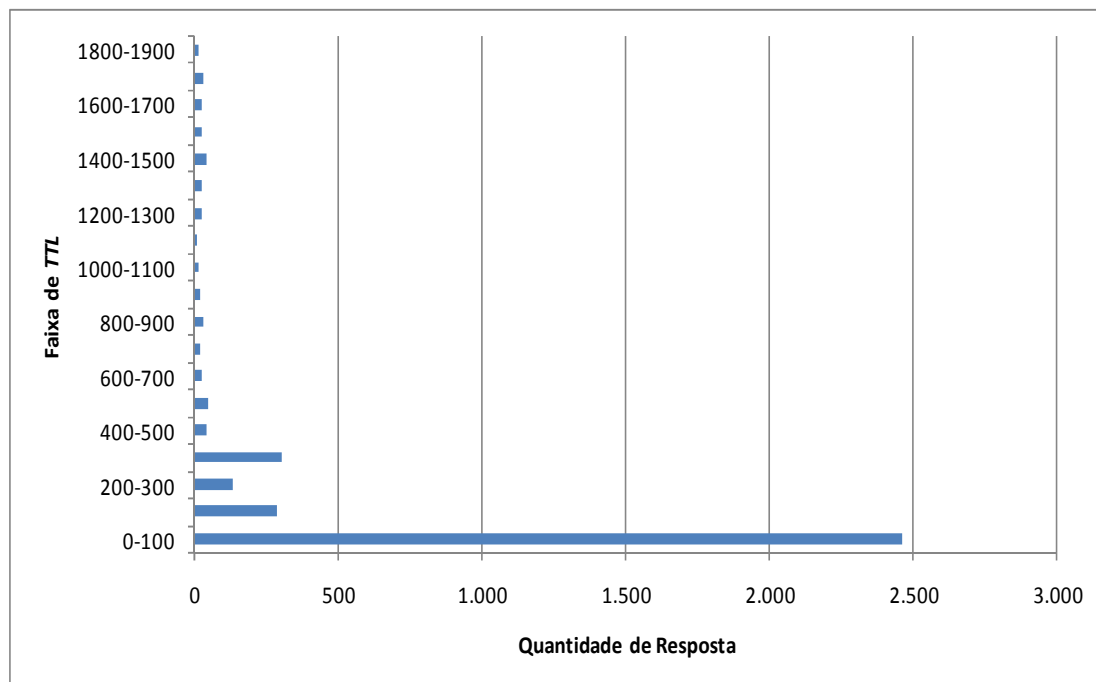


Figura 5.3 - Distribuição das respostas nas faixas de TTL's de 0 a 1900 segundos

Como já previsto anteriormente, o valor do *TTL* por si só não constitui um parâmetro confiável para a constatação de FFSN, apenas pode servir para a diminuição do espaço de busca de um possível domínio malicioso.

5.3.2 Flux-score

Empregando a métrica de *Flux-Score* sobre o tráfego capturado, todos os resultados obtidos do processamento foram notoriamente de falsos positivos. Grande parte dos nomes de domínios sinalizados no experimento usando a métrica *flux-score* foram de domínios CDN

como *akamai.net.* e *freenode.net.* A Tabela 5.2 demonstra o resultado (falso positivo) obtido para o domínio CDN *freenode.net.*

Tabela 5.2 - Falso positivo do domínio *freenode.net.*

Nome de domínio	Nº de resposta	Nº de endereço IP por resposta	Nº de AS	Fluxscore	Limitante
<i>irc.freenode.net</i>	1	14	13	259,5	142,38
<i>chat.freenode.net</i>	1	14	13	259,5	142,38

Os dados mostram que apesar de existir uma única resposta de consulta para cada nome de domínio, 14 endereços IP distintos foram retornados distribuídos em 13 AS diferentes. Aplicando a fórmula do *flux-score*, tem-se que:

$$flux_A = (1.32 \cdot 14 + 18.54 \cdot 13)$$

$$flux_A = 259,5$$

$$flux_A > b$$

O domínio é considerado uma FFSN por ultrapassar o limitante *b*.

A figura 5.4 ilustra os resultados de falso positivo obtidos referente ao domínio *akamai.net.*

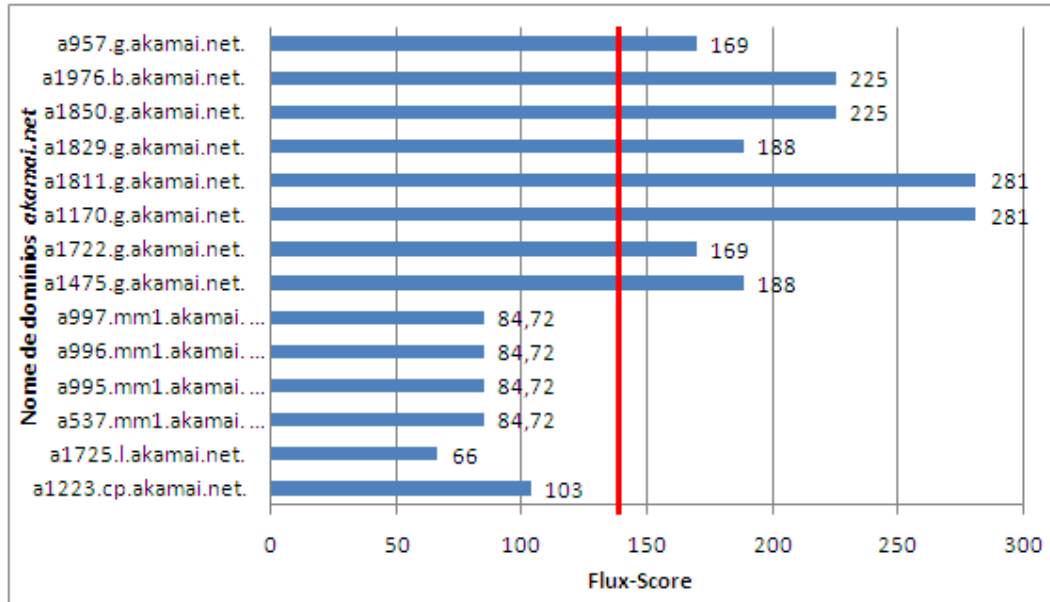


Figura 5.4 - Falsos positivos do domínio *akamai.net.*

A figura 5.5 apresenta todos os domínios legítimos apontados como FFSN.

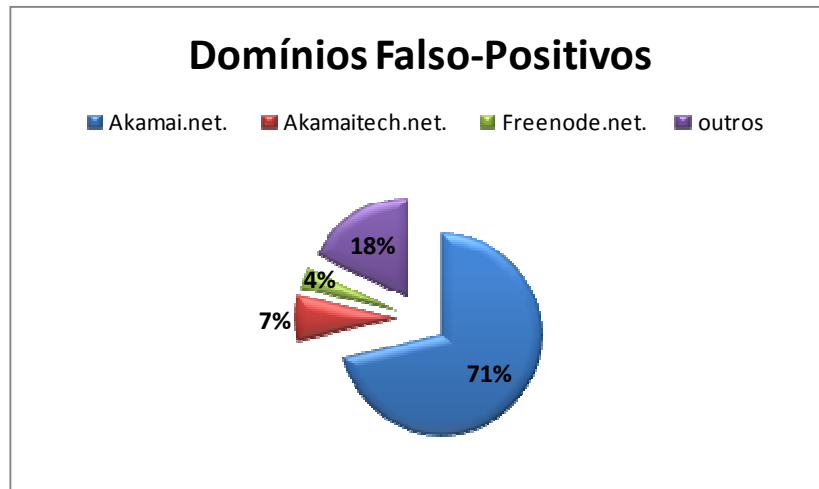


Figura 5.5 - Resultado total de falsos positivos

5.3.3 Discussões

Após uma minuciosa análise sobre os resultados de falsos positivos obtidos, algumas conclusões podem ser retiradas. A fórmula do *flux-score* apresenta uma falha que permite que o valor da função tenha um aumento diretamente proporcional ao aumento do valor de n_A , isto é, o acúmulo de consultas para o mesmo nome de domínio com endereços IP diferentes pode aumentar consideravelmente com o tempo, a ponto da função ultrapassar o valor do limitante.

A Tabela 5.3 ilustra o problema usando o domínio *akamai.net*, como exemplo. A primeira linha representa duas respostas de consulta com apenas dois endereços IP distintos. A segunda linha representa o acúmulo de consultas ao mesmo domínio, agora com 26 endereços IP distintos, o que permitiu ao *flux-score* ultrapassar o limitante.

Tabela 5.3 - Falha do *flux-score*

Nome de domínio	Nº consultas	Nº respostas distintas	NºAS	Flux-score	Limitante
<i>a957.g.akamai.net</i>	2	2	1	21,18	142,38
<i>a957.g.akamai.net</i>	13	26	9	169.499	142,38

Pode-se constatar que no caso da execução do protótipo *online*, ou seja, a análise dos dados fosse realizada enquanto houvesse a captura dos dados, ou mesmo que a execução da

fase de análise fosse realizada em períodos menores, tendo como entrada intervalos de captura menores fosse possível ocorrer uma diminuição do número de falso positivo encontrados.

Uma alternativa possível seria relacionar os valores de n_A e n_{ASN} diretamente entre si, mantendo um comportamento constante, para um determinado domínio independente da quantidade de consultas analisadas. Este fator poderia ser aplicado ao *flux-score* dando mais confiança ao valor retornado. Esse fator também poderia diminuir o valor normalmente atribuído pelo *flux-score* chegando ao ponto de, talvez, obter o resultado contrário, os falsos negativos.

Foi percebido também que o valor de n_{ASN} se mostrou eficiente para determinar o comportamento de FFSN. Domínios com altos números de endereços IP retornados e baixos valores de TTL, mas que tem valores de n_{ASN} baixos (em torno de um ou dois) não são FFSN. Redes *Fast-Flux*, principalmente as utilizadas por *bots* ou *worms*, não tem controle de onde será executada, por isso a dispersão geográfica pode ser considerada um fator de grande peso. Apesar da função de *flux-score* já ter um peso maior, porém, como foi demonstrado neste estudo de caso, para grandes fluxos a função cresce, absorvendo falsos positivos.

6 Conclusão

É notório que hoje em dia ameaças a infra-estrutura da Internet são corriqueiras e o uso de *bots* e *worms* para tal finalidade é quase uma espécie de padrão. Neste obscuro cenário, as redes *Fast-Flux* vêm crescendo e desempenhando um papel cada vez maior. Desta forma, este trabalho contribuiu ao propor a discussão e estudo sobre anomalias do tráfego DNS na Internet focando redes *Fast-Flux*.

De forma sucinta, este documento apresentou uma breve visão geral do protocolo DNS, bem como as mais usuais anomalias e ameaças encontradas e/ou aplicadas ao protocolo, seus efeitos e os principais mecanismos de mitigação.

Também analisou uma técnica de detecção de anomalias, no caso *Fast-Flux Domains*, sobre o protocolo DNS e implementou uma ferramenta capaz de capturar e analisar o tráfego DNS. A correteza funcional desse software foi validada através de testes utilizando tráfego anômalo simulado em laboratório e tráfego real obtido na rede do GPRT.

Como resultado foi observado a aparição de falsos positivos em fluxos de dados muito grandes e de longo tempo acumulado. Isto se dá pela possibilidade de várias consultas ao mesmo domínio serem feitas ao longo de um tempo relativamente grande, podendo, desta forma, domínios legítimos de CDN, que naturalmente têm características semelhantes às de *Fast Flux Domains*, terem seus endereços legitimamente alterados e serem confundidos com redes *Fast-Flux*.

Esta constatação poderá ser importante na medida em que este algoritmo possa ser utilizado para a detecção de anomalias do porte de *Fast Flux Domains* em grandes fluxos e com análise em tempo real.

6.1 Dificuldades Encontradas

Durante a elaboração do presente trabalho foram encontrados vários desafios que, ao mesmo tempo, tornaram o desenvolvimento mais difícil, mas que também, após ultrapassá-los, engrandeceu e trouxe mais qualidade ao trabalho final.

Um desses desafios foi à manipulação de pacotes DNS, que precisou de um estudo detalhado de uma biblioteca específica, o DNSjava [27], o que ao final do trabalho agregou mais conhecimento e experiência. Outro grande desafio foi à validação dos protótipos iniciais sem tráfego malicioso. Existem poucas fontes hoje em dia na Internet, e este obstáculo pôde

ser contornado através da simulação do tráfego necessário, o que também possibilitou um conhecimento adquirido a mais ao projeto.

6.2 Trabalhos Futuros

Alguns trabalhos futuros são:

- Elaborar uma ferramenta de detecção de anomalias baseado no protocolo DNS de maneira *online*, gerando alertas em tempo real. Também se pretende combater os falsos positivos obtidos pelo atual algoritmo e pensar em melhores heurísticas e limitantes dinâmicos assim como novos métodos de detecção de anomalias em tal protocolo.
- Elaborar ou melhorar técnicas de combater os falsos positivos, incluindo o teste e avaliação de um novo fator à equação de *flux-score*.
- Agregar outras técnicas e heurísticas para a detecção de outras anomalias constantes no tráfego DNS.

Referências

- [1] "Domain Name Service (DNS)" by Thomas Lee, Joseph Davies
<http://technet.microsoft.com/en-us/library/bb726935.aspx>, acessado em 27 de Outubro de 2008.
- [2] "Battle Against Fast-Flux Botnets Intensifies",
<http://darkreading.com/security/vulnerabilities/showArticle.jhtml?articleID=211201257>, ultimo acesso em 25 de Novembro de 2008.
- [3] "Dan Kaminsky's 2008 DNS Vulnerability",
http://www.snort.org/vrt/docs/white_papers/DNS_Vulnerability.pdf.
- [4] "DNS Developments Feed Growing Cybersquatting Concerns",
http://www.wipo.int/pressroom/en/articles/2008/article_0015.html, último acesso em 25 de Novembro de 2008
- [5] "DNS hacked again", <http://governanca.cgi.br/noticias/dns-hacked-again/>, último acesso em 18 de Novembro de 2008.
- [6] "Gigantes de internet combateram unidos falha que permitiria controlar a rede",
<http://economia.uol.com.br/ultnot/2008/07/10/ult35u60772.jhtm>, ultimo acesso em 18 de Novembro de 2008.
- [7] "Google AdSense for Domains" <http://www.google.com/domainpark/>, acessado em 04 de Novembro de 2008.
- [8] "Google Profits From Typo Squatting"
<http://blog.wired.com/27bstroke6/2008/10/google-profitin.html>, acessado em 04 de Novembro de 2008.
- [9] "Image attack on MySpace boosts phishing exposure",
<http://www.securityfocus.com/brief/522>, último acesso em 25 de Novembro de 2008.
- [10] "Imperfect Storm aids spammers", <http://www.securityfocus.com/news/11442>, último acesso em 25 de Novembro de 2008.
- [11] "Multiple DNS implementations vulnerable to cache poisoning",
<http://www.kb.cert.org/vuls/id/800113>, ultimo acesso em 18 de Novembro de 2008.
- [12] A. Barbir, B. Cain, R. Nair, O. Spatscheck, "Know content Network (CN) Request-Routing Mechanism", RFC 3568, Julho de 2003.
- [13] A. Schonewille, D. v. Helmond, "The Domain Name Service as an IDS", Research Project for the Master System- and Network Engineering at the University of Amsterdam, Fevereiro de 2006.
- [14] ALBITZ, Paul e LIU, Cricket. **DNS and BIND**. 4ª Edição. EUA: O'Reilly, 1998.
- [15] Ariyapperuma, S. and Mitchell, C. J. 2007. Security vulnerabilities in DNS and DNSSEC. In *Proceedings of the the Second international Conference on Availability, Reliability and Security* (April 10 - 13, 2007). ARES. IEEE Computer Society, Washington, DC, 335-342. DOI= <http://dx.doi.org/10.1109/ARES.2007.139>.
- [16] ATLAS Summary Report: Global Fastflux, Arbor Networks.
<http://atlas.arbor.net/summary/fastflux>, ultimo acesso em 25 de Novembro de 2008.
- [17] B. Zdrnja, N. Brownlee, and D. Wessels. Passive Monitoring of DNS Anomalies. In *Proceedings of the 4th Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, pages 129–39. Springer Berlin, 2007.
- [18] Bellovin, S. M. 1995. Using the domain name system for system break-ins. In *Proceedings of the 5th Conference on USENIX UNIX Security Symposium - Volume 5* (Salt Lake City, Utah, June 05 - 07, 1995). USENIX Security Symposium. USENIX Association, Berkeley, CA, 18-18.

- [19] Berkeley Internet Name Domain – BIND. <https://www.isc.org/software/bind>, ultimo acesso em 18 de Novembro de 2008.
- [20] Bojan Zdrnja, "Security Monitoring of DNS traffic", CompSci780 project, University of Auckland, Maio de 2006, http://www.caida.org/~nevil/Bojan_Zdrnja_CompSci780_Project.pdf
- [21] C. Jackson, A. Barth, A. Bortz, W. Shao e D. Boneh, Protecting Browsers from DNS Rebinding Attacks, In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, Outubro de 2007.
- [22] Christoph L. Schuba and Eugene H. Spafford. *Addressing weaknesses in the domain name system protocol*. Master's thesis, Purdue University, 1993. Department of Computer Sciences.
- [23] D. Wessels and M. Fomenkov, "Wow, That's a Lot of Packets," in Proc. 2003 Passive and Active Measurements Workshop, April 2003.
- [24] Daniel Sánchez e Joost Pijnaker. "What is all that crap?" *Analysis of DNS root server bogus queries*. Master Student Project, University of Amsterdam, 2007.
- [25] DNS Cache Poisoning – The Next Generation. www.lurhq.com/dnscache.pdf, último acesso em 25 de Novembro.
- [26] DNSCAP - DNS traffic capture utility , 2008. <https://www.dns-oarc.net/tools/dnscap>
- [27] Dnsjava, <http://www.dnsjava.org/>, último acesso em 25 de Novembro de 2008.
- [28] DNS-OARC. <https://www.dns-oarc.net/>, ultimo acesso em 25 de Novembro de 2008.
- [29] DNSSEC – Security Extension. <http://www.dnssec.net/>, ultimo acesso em 25 de Novembro de 2008.
- [30] *Double Flux Defense in the DNS Protocol*, <http://tools.ietf.org/html/draft-bambenek-doubleflux-01>, ultimo acesso em 25 de Novembro de 2008.
- [31] F. Weimer, "Passive DNS Replication," FIRST 2005, April 2005.
- [32] ICANN Security and Stability Advisory Committee (SSAC). SAC 025: SSAC Advisory on Fast Flux Hosting and DNS, 2008.
- [33] J. Kristoff, "DNSWatch", <http://aharp.ittns.northwestern.edu/software/dnswatch>, último acesso em 25 de Novembro de 2008.
- [34] J. Oberheide, Manish Karir, Z. Morley Mao. Characterizing Dark DNS Behavior. In *Proceedings of the 4th Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, pages 140–56. Springer Berlin, 2007.
- [35] Lazarevic, A., Ertöz, L., Kumar, V., Ozgur, A. & Srivastava, J. (2003), A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the Third SIAM International Conference on Data Mining*.
- [36] P. Mockapetris, "Domain Names - Concepts and Facilities", RFC 1034, Novembro 1987.
- [37] P. Mockapetris, "Domain Names - Implementation and Specification", RFC 1035, Novembro 1987.
- [38] T. Brisco, "DNS Support for Load Balancing", RFC 1794, Abril de 1995.
- [39] T. Holz, C. Gorecki, K. Rieck, and F. C. Freiling. Measuring and Detecting Fast-Flux Service Networks. In *Proceedings of the 15th Annual Network & Distributed System Security Symposium (NDSS)*, 2008.
- [40] The Honeynet Project. Know Your Enemy: Fast-Flux Service Networks, Julho de 2007. <http://www.honeynet.org/papers/ff/>.

- [41] The Measurement Factory: dnstop tool, 2006. <http://dns.measurement-factory.com/tools/dnstop/>.
- [42] The Measurement Factory: DSC-a dns statistics collector, 2006. <http://dns.measurement-factory.com/tools/dsc/>.
- [43] Wireshark: Go Deep. <http://www.wireshark.org/>, ultimo acesso em 25 de Novembro de 2008.
- [44] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, "Address Allocation for Private Internets", RFC 1918, Fevereiro de 1996.
- [45] Yi-Min Wang, Doug Beck, Jeffrey Wang, Chad Verbowski, and Brad Daniels, Strider Typo-Patrol: Discovery and Analysis of Systematic Typo-Squatting, in *Proc. Usenix SRUTI Workshop*, Julho de 2006.

Glossário

Autoridade – é a responsabilidade de resposta para uma determinada consulta. Vários servidores podem responder a uma consulta, porém apenas um é a fonte original desta informação, o servidor de nomes de tal domínio.

Bot - abreviação para robô, a palavra é utilizada para descrever programas, normalmente rodados em servidores, que automatizam tarefas como, por exemplo, encaminhar ou agrupar e-mail's. Ultimamente, muito utilizado para fins maliciosos, automatizando a execução de grandes ataques.

Caching – É o ato de manter uma determinada informação solicitada armazenada na máquina local, caso seja preciso utilizá-la em um curto espaço de tempo é possível obter a informação local sem requisitar novamente, desde que respeite o seu tempo de vida, dado pelo *TTL*.

Country Code Top Level Domain – TLD's com uma abreviação padrão de dois caracteres que correspondem a um país, território ou localidade geográfica.

Delegar – É a ação de distribuir as informações de um domínio para outros servidores de hierarquia mais abaixo, repassando a responsabilidade de resposta para um determinado subdomínio para outro servidor de nomes.

Domínio – Uma região de jurisdição para atribuição de nome e conteúdo na *web*. Uma porção da internet que tem registros em um servidor de nomes associados a ela.

Espaço de Nomes de Domínio - Um esquema de endereçamento da internet que é hierárquico por natureza e utiliza uma estrutura de árvore para organizar a informação que descreve redes e computadores. É a soma de todos os nomes de domínios que representam redes e computadores, assim como todos os nomes possíveis, ainda não utilizados, que podem potencialmente representar redes de computadores. Um conjunto distribuído de nomes no qual cada um é único.

Generic Top Level Domain – Uma porção do espaço de nomes alocada internacionalmente. Os TLD's com três ou mais caracteres são referenciados como genéricos.

Nome de Domínio – Uma designação alfa-numérica única para facilitar a referência a endereços IP que identificam um computador em particular ou uma rede. Para todo nome de domínio existe um registro de recurso correspondente dentro do DNS.

Resolução de Nomes – O processo de conversão de um nome de domínio para endereço da internet correspondente. Se um servidor no domínio local não pode resolver a requisição do cliente, ele tenta localizar um servidor que pode resolvê-lo através do uso de consultas iterativas a outros servidores.

Resolver - Um conjunto de rotinas contidas nas bibliotecas do sistema operacional que provêem a interface que os programas utilizam para acessar o nome de domínio.

Resource Record – Os dados associados com um nome de domínio específico. A classe Internet (IN) do registro de recurso é a mais popular.

Reverse DNS – É um tipo de consulta onde o parâmetro utilizado é o endereço IP e a resposta esperada é o nome de domínio correspondente.

Root (root domain) – O topo da hierarquia do DNS. Frequentemente referenciado com o ".". Tentativas finais de resolução acontecem aqui, caso os servidores de mais baixo nível não possuam os dados requisitados.

Root Servers - Um servidor que contém os endereços IP de todos os TLD's

Subdomínio - Um domínio que se ramifica a partir de outro.

Time-To-Live – É o tempo determinado pelo servidor de nomes de determinado domínio para o qual uma informação é válida. Quando este tempo expira, é preciso voltar a solicitar tal informação ao servidor responsável.

Top Level Domain – Top level domains são os nomes no topo da hierarquia de nomes do DNS. É o nível mais alto da hierarquia depois do *root*. Em um nome de domínio, a porção do nome que aparece mais a direita.

Zona – A porção do espaço de nomes de domínio que é servida por um servidor de nomes. Pode ser um domínio inteiro, um domínio com todos os seus subdomínios, ou uma porção de um domínio onde um servidor de nomes tem autoridade para manter os dados.

Apêndice A – Parâmetros DNS⁴

Registros inclusos abaixo:

- Tipos de Resource Record (RR)
- DNS OpCodes
- DNS RCODEs
- Flags do Cabeçalho DNS

Nome do Campo: Tipos de Registro de Recursos (RR)

Referência: [RFC5395][RFC1035]

TIPO	Valor e Significado	Referência
A	1 um endereço do host	[RFC1035]
NS	2 um servidor de nomes autoritário	[RFC1035]
MD	3 um e-mail de destino (Obsoleto - use MX)	[RFC1035]
MF	4 um e-mail de origem (Obsoleto - use MX)	[RFC1035]
CNAME	5 o nome canônico para um "alias"	[RFC1035]
SOA	6 delimita o início de uma zona de autoridade	[RFC1035]
MB	7 um nome de domínio de uma caixa de e-mail	[RFC1035]
MG	8 um membro de um grupo de e-mail	[RFC1035]
MR	9 uma renomeação de um domínio de e-mail	[RFC1035]
NULL	10 um registro nulo	[RFC1035]
WKS	11 uma descrição de um service bem conhecido	[RFC1035]
PTR	12 um apontador para um nome de domínio	[RFC1035]
HINFO	13 informação do host	[RFC1035]
MINFO	14 informação de uma lista de e-mail	[RFC1035]
MX	15 troca de e-mail	[RFC1035]
TXT	16 texto	[RFC1035]
RP	17 para a pessoa responsável	[RFC1183]
AFSDB	18 para localização na base de dados AFS	[RFC1183]
X25	19 para um endereço X.25 PSDN	[RFC1183]
ISDN	20 para um endereço ISDN	[RFC1183]
RT	21 para uma rota direta	[RFC1183]
NSAP	22 para um endereço NSAP	[RFC1706]
NSAP-PTR	23 para um apontador de domínio, estilo NSAP	[RFC1348]
SIG	24 para uma assinatura de segurança	[RFC4034][RFC3755][RFC2535]
KEY	25 para uma chave de segurança	[RFC4034][RFC3755][RFC2535]
PX	26 informação de mapeamento de e-mail X.400	[RFC2163]
GPOS	27 posição geográfica	[RFC1712]
AAAA	28 endereço IPv6	[RFC3596]
LOC	29 informação de localização	[RFC1876]
NXT	30 próximo domínio - OBSOLETO	[RFC3755][RFC2535]
EID	31 identificador de ponto final	[Patton]
NIMLOC	32 localizador Nimrod	[Patton]
SRV	33 seleção de servidor	[RFC2782]
ATMA	34 endereço ATM	[ATMDOC]
NAPTR	35 apontador para autoridade de nome	[RFC2915][RFC2168]
KX	36 troca de chave	[RFC2230]
CERT	37 CERT	[RFC2538]
A6	38 A6 (Experimental)	[RFC3226][RFC2874]
DNAME	39 DNAME	[RFC2672]
SINK	40 SINK	[Eastlake]
OPT	41 OPT	[RFC2671]
APL	42 APL	[RFC3123]

⁴ Versão Traduzida de: IANA – Internet Assigned Numbers Authority, [HTTP://www.iana.org/assignments/dns-parameters](http://www.iana.org/assignments/dns-parameters)

DS	43	senalizador de delegação	[RFC3658]
SSHFP	44	impressão digital da chave SSH	[RFC4255]
IPSECKEY	45	IPSECKEY	[RFC4025]
RRSIG	46	RRSIG	[RFC4034] [RFC3755]
NSEC	47	NSEC	[RFC4034] [RFC3755]
DNSKEY	48	DNSKEY	[RFC4034] [RFC3755]
DHCID	49	DHCID	[RFC4701]
NSEC3	50	NSEC3	[RFC5155]
NSEC3PARAM	51	NSEC3PARAM	[RFC5155]
Unassigned	52-54		
HIP	55	protocolo de identificação de host	[RFC5205]
Unassigned	56-98		
SPF	99		[RFC4408]
UINFO	100		[IANA-Reservado]
UID	101		[IANA-Reservado]
GID	102		[IANA-Reservado]
UNSPEC	103		[IANA-Reservado]
Unassigned	104-248		
TKEY	249	chave de transação	[RFC2930]
TSIG	250	assinatura de transação	[RFC2845]
IXFR	251	transferência incremental	[RFC1995]
AXFR	252	transferência de uma zona inteira	[RFC1035]
MAILB	253	registros relacionados à e-mail	[RFC1035]
MAILA	254	agente de e-mail (Obsoleto - ver MX)	[RFC1035]
*	255	uma requisição para todos os registros	[RFC1035]

Nome do Campo: DNS OpCodes

Referência: [RFC5395] [RFC1035]

OpCode	Nome	Referência
0	Consulta	[RFC1035]
1	Consulta inversa (Obsoleto)	[RFC3425]
2	Status	[RFC1035]
3	Não-atribuído	
4	Notificação	[RFC1996]
5	Atualização	[RFC2136]
6-15	Não-atribuído	

Nome do Campo: DNS RCODEs

Referência: [RFC5395] [RFC1035]

RCODE	Nome	Descrição	Referência
0	NoError	Nenhum erro	[RFC1035]
1	FormErr	Erro de formato	[RFC1035]
2	ServFail	Falha no servidor	[RFC1035]
3	NXDomain	Domínio não existente	[RFC1035]
4	NotImp	Não implementado	[RFC1035]
5	Refused	Consulta recusada	[RFC1035]
6	YXDomain	Nome existe quando não deveria	[RFC2136]
7	YXRRSet	Registro existe quando não deveria	[RFC2136]
8	NXRRSet	Registro que deveria existir não existe	[RFC2136]
9	NotAuth	Server Not Authoritative for zone	[RFC2136]
10	NotZone	Nome não contido na zona	[RFC2136]
11-15	Unassigned		
16	BADVERS	versão de OPT errada	[RFC2671]
16	BADSIG	Falha na assinatura TSIG	[RFC2845]
17	BADKEY	Chave não reconhecida	[RFC2845]

18	BADTIME	Assinatura fora da janela de tempo	[RFC2845]
19	BADMODE	Modo TKEY errado	[RFC2930]
20	BADNAME	chave do nome duplicada	[RFC2930]
21	BADALG	Algoritmo não suportado	[RFC2930]
22	BADTRUNC	Truncado	[RFC4635]
23-3840	Não-atribuído		
3841-4095	Reservado para uso privado		[RFC5395]
4096-65534	Não-atribuído		
65535	Reservado		[RFC5395]

Nome do Campo: Flags do Cabeçalho DNS

Referência: [RFC1035]

Bit	Flag	Descrição	Referência
bit 5	AA	Resposta Autoritária	[RFC1035]
bit 6	TC	Resposta Truncada	[RFC1035]
bit 7	RD	Recursão Desejada	[RFC1035]
bit 8	RA	Recursão Permitida	[RFC1035]
bit 9		Reservado	
bit 10	AD	Dado Autêntico	[RFC4035]
bit 11	CD	Checação Desabilitada	[RFC4035]