UNIVERSIDADE DE BRASÍLIA
Engenharia de Redes de Comunicação
Pós-graduação Lato Sensu
Curso de Especialização em Gestão de segurança da Informação

Descoberta de DNS Tunneling utilizando ferramentas de Big Data

Bruce William Percílio Azevedo

Prof. Dr. Laerte Peotta

Brasília 2017

UNIVERSIDADE DE BRASÍLIA
Engenharia de Redes de Comunicação
Pós-graduação Lato Sensu
Curso de Especialização em Gestão de Segurança da Informação

Descoberta de DNS Tunneling utilizando ferramentas de Big Data

Bruce William Percílio Azevedo

Prof. Dr. Laerte Peotta

Monografia apresentada ao departamento de engenharia de redes de comunicação, da Universidade de Brasília – UNB, como requisito parcial à obtenção do grau especialista em Gestão de Segurança da Informação - GSI

Brasília 2017

UNIVERSIDADE DE BRASÍLIA

Engenharia de Redes de Comunicação

Pós-graduação Lato Sensu

Curso de Especialização em Gestão de Segurança da Informação

Monografia apresentada ao departamento de engenharia de redes de comunicação, da Universidade de Brasília – UNB, como requisito parcial à obtenção do grau especialista em Gestão de Segurança da Informação – GSI

Descoberta de DNS Tunneling utilizando ferramentas de Bigdata

Bruce William Percílio Azevedo

Aprovado por:

Professor Orientador: Dr. Laerte Peotta

Professor:

Professor:

Brasília, data a definir (dia da banca) 2017

Dedicatória

Dedico esse trabalho a todos aqueles que ousaram imaginar o que havia além do que seus olhos podiam enxergar.

Agradecimentos

Agradeço primeiramente a Deus pelo maravilhoso presente que é a vida.

Agradeço aos meus pais pelo apoio incontestável que me deram durante todo o meu percurso na pós-graduação.

Agradeço a universidade pela oportunidade que me forneceu e a meus professores por tudo o que se propuseram a me ensinar.

E a todos que direta ou indiretamente fizeram parte da minha formação, deixo os meus sinceros agradecimentos.

Epígrafe

Se você conhece o inimigo e conhece a si mesmo, não precisa temer o resultado de cem batalhas. Se você se conhece, mas não conhece o inimigo, para cada vitória ganha sofrerá também uma derrota. Se você não conhece nem o inimigo nem a si mesmo, perderá todas as batalhas. – Sun Tzu

Resumo

O protocolo DNS é uma das bases da rede mundial de computadores. Devido a sua estrutura simples e a necessidade de sua utilização, ele vem sido alvo de hackers. Uma das técnicas mais utilizadas por hackers (visando o protocolo DNS) é esconder comunicação maliciosa dentro do tráfego DNS. Utilizando ferramentas de Big Data com foco em análise de dados de máquina, é possível extrair pacotes DNS, normaliza-los, armazena-los e analisa-los, com o objetivo de buscar por anomalias em sua estrutura. Para conseguir distinguir pacotes autênticos dos alterados será utilizado uma técnica de quantificação de entropia nas queries dentro dos pacotes DNS. O objetivo principal é verificar se existe coligação da taxa de entropia e a frequência de queries, com o uso de DNS tunneling.

Palavras chave: Entropia, frequência, DNS Tunneling.

Abstract

The DNS protocol is one of the foundations of the global computer network. Due his simple structure and his utilization need, it has been hackers target. One of the most used techniques by hackers (aiming at the DNS protocol) is to hide malicious communication within the DNS traffic. Using Big Data tools with focus on machine data analysis, we can extract DNS packets, parsing them, load them and analyze them, in order to search for anomalies in their structure. With objective to distinguish between authenticated and modified packets, a technique of quantifying entropy in queries within DNS packets will be used. The main objective is to verify if there is an association of the entropy rate and the frequency of queries, with the use of DNS tunneling.

Key words: Entropy, frequency, DNS Tunneling.

Lista de Ilustrações

Lista de tabelas

Lista de Abreviaturas

DNS – Domain Name System

ETL – Extract Transform Load

Sumário

1 Introdução

A utilização de Big Data para extrair conhecimento útil e estratégico de dados provenientes de maquinas é uma proposta que vem sido muito aceita, adotada e dissimulada. Segundo o artigo "New Busisess Trends Created by Big Data utilization" (Makoto Yahada, Yasuhada Namba, Jun Yoshida, 2014), dados produzidos por máquinas estão entre os 5 tipos com maior valor. Será utilizado uma ferramenta de Big Data com foco em dados de máquina, com o objetivo descobrir anomalias no trafego DNS.

A utilização de DNS em qualquer organização é algo essencial hoje em dia, pois esse protocolo é uma das bases da rede mundial de computadores. Segundo a SANS em seu artigo "Security Issues with DNS " (2003), uma das principais técnicas utilizadas por atacantes é alterar o conteúdo dos pacotes DNS, fazendo com que os eles se desvirtuem do seu objetivo inicial e sirvam para permitir comunicação mascarada dentro da infraestrutura de rede.

Tomando como base o artigo da SANS "Using Splunk to Detect DNS Tuneling" (Rick Wanner, 2016), o objetivo desse trabalho é utilizar uma ferramenta de Big Data juntamente com técnicas de entropia para auxiliar na descoberta de anomalias no trafego DNS. Visando encontrar características que liguem o conteúdo de pacotes DNS que fujam do padrão com a utilização de DNS Tunneling. Para cumprir com esse objetivo algumas perguntas devem ser respondidas: Como funciona o protocolo DNS? Como pessoas mal-intencionadas usam o protocolo DNS para esconder a comunicação? Qual a diferença de um pacote DNS legitimo para um pacote adulterado? Como reconhecer um pacote DNS adulterado de forma automatizada?

1.1 Motivação

O protocolo DNS é importante para a localização de ativos dentro de uma infraestrutura complexa (como a internet). Sua utilização é necessária para todos os usuários. Fazer uso de seus recursos para mascarar qualquer comunicação IP é um grande risco a infraestrutura vigente.

Ferramentas de Big Data foram concebidas para analisar uma grande quantidade de dados simultaneamente. A elaboração de uma pesquisa que busque abordar de forma prática técnicas de verificação de DNS tunneling utilizando ferramentas de Big Data, é

viável e pode trazer bons resultados na busca por mais segurança de forma não intrusiva na infraestrutura monitorada.

1.2 Objetivo do trabalho

O objetivo desse trabalho é apresentar um estudo sobre como detectar de forma passiva ataques DNS Tunneling. Serão analisados os pacotes provenientes do trafego DNS, com o intuito de buscar anomalias em seu conteúdo que apontem o desvio de padrão causado durante a efetivação desse tipo de ataque.

1.3 Metodologia de pesquisa

A metodologia de pesquisa foi dividida em 3 fases, que buscam facilitar o entendimento do tema e explana-lo da melhor forma possível, sendo elas:

- **Fase 1**: Realizar pesquisas bibliográficas baseadas no assunto, buscando mecanismos dinâmicos que permitam melhor identificação, leitura e análise de artigos e livros relevantes para melhor compreender e explanar sobre o tema.
- **Fase 2**: Obter dados de uma comunicação DNS, para categorizar o comportamento normal e utilizar técnicas para identificar um possível desvio desse comportamento.
- **Fase 3**: Simular e desenvolver o modelo proposto, analisando os resultados. Nessa fase serão efetuadas as conclusões e identificação das contribuições.

1.4 Contribuição do trabalho

Busca-se com esse trabalho as seguintes contribuições:

- Utilização de Big Data para normalizar e absorver eventos gerados diretamente por máquinas
- Apresentação de técnicas para analisar desvio de comportamento do protocolo DNS.
- Apresentação de proposta para identificação de comportamento suspeito através da quantificação de sua entropia.

- Simulação de modelo proposto, aplicando a solução proposta em um cenário controlado.

1.5 Organização do trabalho

Para um melhor entendimento deste trabalho, a sua organização é descrita a seguir:

O capítulo 2 oferece uma fundamentação dos conceitos de Big Data, sua utilização focada em dados de máquina e sua utilização na descoberta de ataques DNS, e quais são os principais ataques DNS empregados hoje em dia.

O capitulo 3 é apresentado uma explicação sobre o que é DNS tunneling, pesquisas que mostram como essa técnica funciona, como vem sendo implementada e como utilizar ferramentas de Big Data junto a técnicas de cálculo de entropia para detecta-las.

O capitulo 4 apresenta um estudo prático utilizando ferramentas reais e mercadológicas.

O capitulo 5 encerra o trabalho, apresentando resultados obtidos e caminhos futuros que podem ser seguidos para a sequência desse trabalho.

2 Fundamentos e conceitos

2.1 Big Data

O termo "Big Data" vem sendo repetido em diversos contextos e utilizado em diversas áreas do conhecimento, se tornando ambíguo. Segundo o artigo "A very short history of Big Data" escrito por Gil Press e no site da Forbes, a historia do termo "Big Data" começou a mais de 70 anos atrás, quando se iniciou a pesquisas cientificas que buscavam quantificar a taxa de crescimento no volume de dados, que foi popularmente conhecido como "information explosion" (um termo primeiramente utilizado em 1941, pelo dicionário de inglês Oxford). Com o passar do tempo mais pesquisas foram sido realizadas sobre o assunto e o termo "Big Data" foi se popularizando. Atualmente o termo é assemelhado com técnicas e ferramentas que visam o armazenamento, normalização e manipulação de grandes quantidades de dados (estruturados ou não), objetivando extrair valores e informações de seu conteúdo. As pesquisas mais relevantes são:

- **1944**, Fremont Rider, "The scholar and the future of the research library".
- **1961**, Derek PRice, "Science Since Babylon".
- **1967**, B.A. Marron e P.A.D. de Maine, "Automatic Data Compression".
- **1971**, Arthur Miller, "The assult on provacy"
- **1980**, I.A Tjomslad, "Where do we go from here"
- **1983**, Ithiel de Sola Pool, "Tracking the flow of information".
- **1990**, Peter J. Denning, "Saving all bits".
- **1997**, Michel Cox e David Ellsworth, "Application controlled demand paging for out-of-core visualization".
- **1997**, Michael Lesk, "How much information is there in the world".
- **1999**, Steve Bryson, David Kenwright, Michael Cox, David Ellsworth, e Robert Haimes, "Visually exploring gigabyte data sets in real time".
- **1999**, Bryson, Kenwright and Haimes join David Banks, Robert van Liere, e Sam Uselton, "Automation or interaction: what's best for big data?".
- **2001**, Doug Laney, "3D Data Management: Controlling Data Volume, Velocity, and Variety.".

- **2011**, Martin Hilbert e Priscila Lopez, “The World’s Technological Capacity to Store, Communicate, and Compute Information”.
- **2011**, James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh e Angela Hung Byers, “Big data: The next frontier for innovation, competition, and productivity.”.

Todas essas pesquisas tiveram muita relevância no desenvolvimento do Big Data que é conhecida hoje. Tendo sua relevância nos aspectos que cobriram, a que melhor cobriu a definição da estrutura do que Big Data foi a de Doug Laney, “3D Data Management: Controlling Data Volume, Velocity, and Variety.”. Ele trouxe uma fundamentação desse conceito, relatando que as técnicas de Big Data são estruturadas sobre 3 “Vs”:

- **Volume**: Retrata a quantidade de dados que são produzidas pelas mais diversas fontes, sendo normal o trabalho com Giga e Terabytes.
- **Velocidade**: Esse alicerce mostra a velocidade que os dados são gerados, estima-se que 2,5 quinquilhões de dados são gerados diariamente.
- **Variedade**: São os formatos que os dados são gerados, podendo ser estruturados ou não, alguns exemplos famosos são arquivos texto claro (csv, txt) e arquivos binários (.evt, .bin).

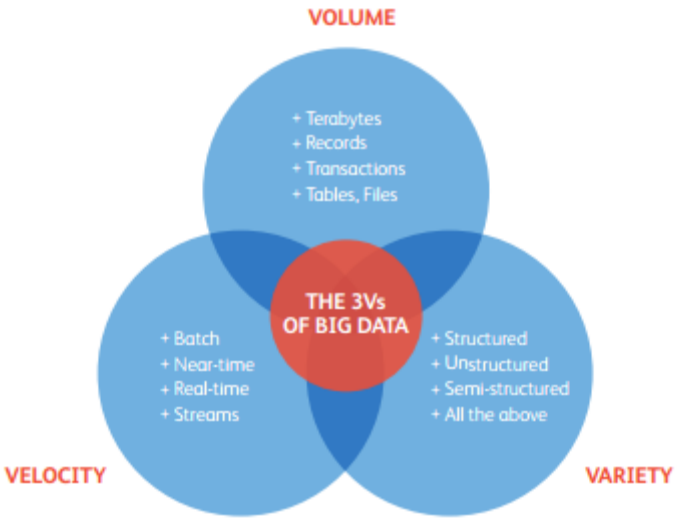

Figura 1: Volume x Velocity x Variety

Qualquer ferramenta de Big Data é implementada buscando lidar com esses 3 pilares. Após o conceito fundamentado, as pesquisas que foram realizadas buscaram

principalmente focar em sua aplicabilidade visando áreas especificas. O artigo "New Business Trends Created by Big Data Utilization" escrito por Makoto Yasuda, Yasuhura Namba e Jun Yoshida, em 2014 explana bem essa questão. Trazendo as principais fontes de dados utilizadas por empresas e pesquisadores no uso da Big data. Elas são:

- **Humam-oriented (Orientação Humana)**: Dados produzidos por intermédio humano, como por exemplo vídeos, áudios e fotos (encontrados principalmente em redes sociais).
- **Machine (Máquina)**: Dados produzidos por dispositivos inteligentes no desempenhar de sua tarefa ou durante sua operação.
- **Location (Localização)**: Dados baseados em localização geográfica, como por exemplo GPS, localização de chamadas de telefone, navegação de carros, etc.
- **Market (Negócios)**: São dados produzidos por empresas cotidianamente, como documentos, dados de funcionários, transações financeiras, etc.
- **Smart Infrastructure (Infraestrutura inteligente)**: Dados produzidos a partir de sensores inteligentes que monitoram condições de determinada região, como clima, trafego, temperatura, etc.

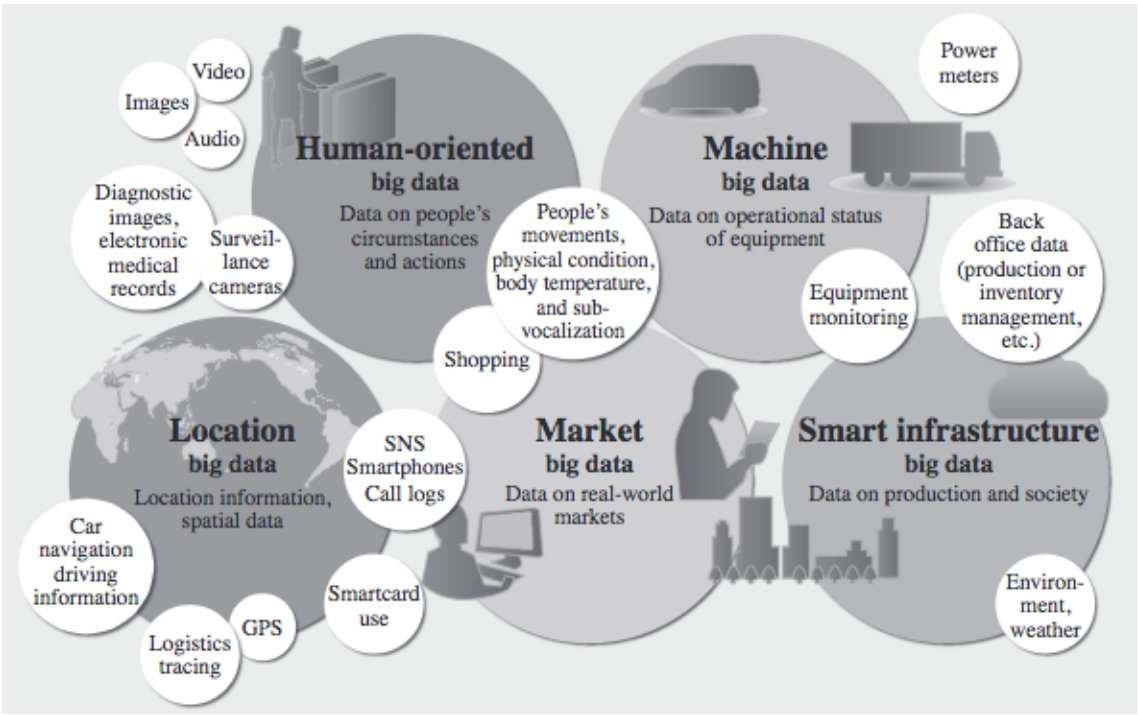

Figura 2: Tipos de dados de máquina

A presente pesquisa foca estritamente em dados produzidos por máquinas.

Para realizar a construção de conhecimento sobre essa fonte são utilizados 2 processos repetidos inúmeras vezes, o processo ETL e a construção de conhecimento e visibilidade de dados com ferramentas.

A metodologia que se inicia os trabalhos em qualquer base dados (independente do tipo) utilizando ferramentas de Big Data é o processo ETL (Extract, Transform, Load). O processo ETL foi originalmente desenhado para trabalhar dados originados de negócios, porém seu conceito se aplica a todas as fontes de dados apresentadas anteriormente. O artigo "Conceptual Modeling for ETL Process" escrito por Panos Vassiliadis, Alkis Simitsis e Spiros Skiadopoulos, publicado na National Techinical University of Athens, fala que o processo ETL pode diminuir o esforço necessário para gerar conhecimento sobre dados brutos em até 80%. O processo ETL é um baseado em 3 etapas:

1. **Extract (Extrair)**: Nessa primeira parte é identificado o tipo de dado, sua localização e o procedimento necessário para realizar sua coleta.
2. **Transform (Transformar)**: Nessa etapa é realizada a normalização dos dados. Os dados podem estar disponíveis em qualquer tipo e formato, aqui uma padronização deve ser definida e adotada.
3. **Load (Carregar)**: Depois de extraída e normalizada, os dados devem ser inseridos em um local para sua análise.

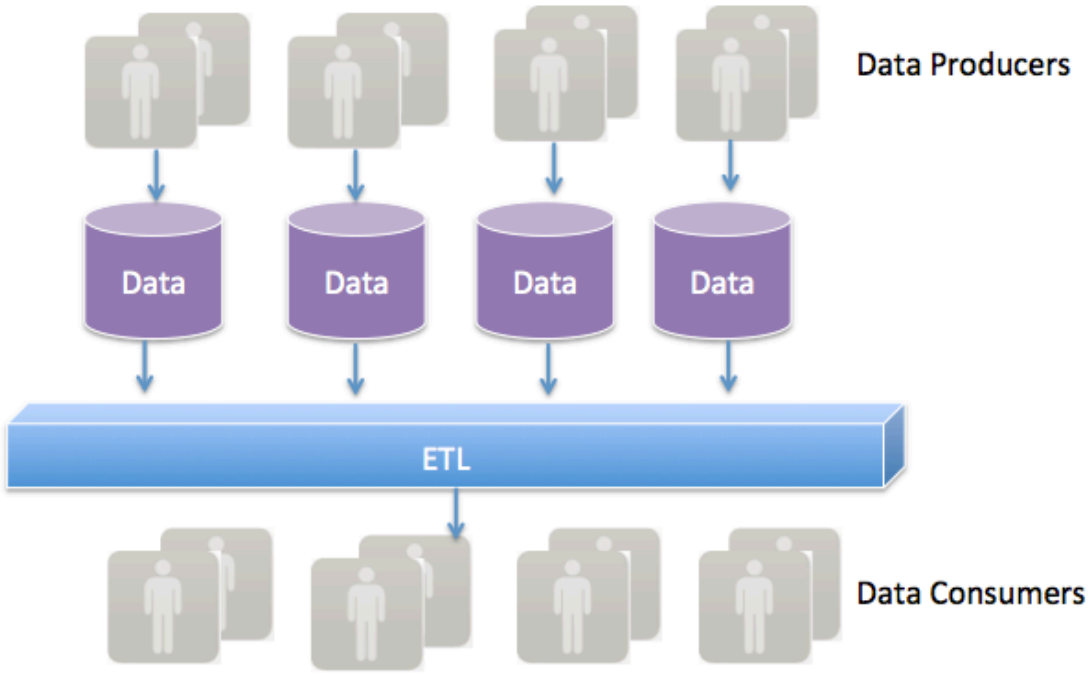

Figura 3: Processo ETL

Os dados que vão ser submetidos ao processo de análise buscando a construção de conhecimento passam inicialmente por algum tipo de processo ETL, para que depois possam passar pelo processo de Data Mining ou Data Science. Esse processo é singular para cada tipo de dado trabalhado, sendo o processo ou ferramenta utilizada útil para determinada situação. Nesse processo a estrutura dos dados é entendida, seus objetos extraídos, manipulados e uma visibilidade baseado nesse tratamento é criada.

2.1 Big Data e tipos de ataques DNS

A utilização de ferramentas de Big Data com foco na analise de dados vem sendo utilizadas para auxílio na descoberta de ataque DNS. Um grande exemplo é o artigo "Using Splunk to Detect DNS Tuneling", publicado na SANS e escrito por Rick Wanner em 2016. Ele utiliza uma ferramenta de big data para extrair, transformar, armazenar e gerar visibilidade sobre dados provenientes do trafego DNS, apontando superficialmente um tipo de ataque que é efetivado sobre o protocolo DNS.

A analise do conteúdo de cada pacote DNS por ferramentas de Big Data (com foco na descoberta de ataques DNS) são feitas objetivando identificar características resultantes da aplicação de técnicas maliciosas por usuários mal-intencionados. Segundo o artigo "DNS and DNS attacks", escrito por Lior Rozen, publicado no blog Radware em 7 de setembro de 2016, as principais técnicas utilizadas para atacar servidores DNS são:

- **Reflection Attacks**: Esse ataque é utilizado para atacar uma vitima "3° party", mesmo que ele não execute requisições ao servidor DNS. Este é um dos vetores de ataque mais comuns no mundo DDoS. Sua popularidade veio do fato de que ser completamente falsificado (sendo muito difícil de identificar o atacante), e pode amplificar a largura de banda do ataque de forma que alguns pacotes causem a saturação de uma larga banda de internet.
- **DNS Tunnels**: Essa por si só não é uma técnica para atacar servidores DNS, mas sim uma maneira de utilizar a infraestrutura e o protocolo DNS para trafegar dados maliciosos a baixo do radar. A técnica usa o protocolo DNS como túnel, escondendo comunicação maliciosa sobre o trafego DNS. Essa técnica é usada para ignorar regras de firewall, mecanismos de monetização WI-FI, sistemas de prevenção de perda de dados e qualquer outra tecnologia usada para inspecionar ou limitar dados através da rede. Malwares muitas vezes usam essa técnica com o

objetivo de se comunicar com o mundo exterior e evitar a segurança da infraestrutura de rede.

- **Server Attacks**: Esse tipo de ataque é direcionado a um servidor DNS especifico e pode ter diversos objetivos, sendo o mais comum buscar causar negação de serviço. Outro objetivo comum é obter todos os dados armazenados em um servidor DNS, afim de estudar a infra-estrutura de rede do alvo. Tais estudos são usados mais tarde para encontrar vetores mais eficientes de ataque. Outro objetivo desse ataque é tomar controle do servidor através de vulnerabilidades e anomalias do protocolo. Embora servidores autoritários e recursivos sofram desse tipo de ataque, são utilizados técnicas diferentes, buscando se aproveitar dos modos diferentes de operação, buscando maximizar o impacto do ataque. Provedores de hospedagem são frequentemente alvos desse tipo de ataque.
- **Spoofing Results**: Esse tipo de ataque visa alterar uma resposta DNS válida por uma maliciosa. Por mais que o ataque seja realizado no servidor DNS, seu objetivo é afetar os clientes. O seu objetivo é enganar o usuário, fazendo ele acessar um site malicioso ao invés de acessar o verdadeiro. A técnica é utilizada principalmente como parte de uma técnica de phishing em sites de dados pessoais ou financeiros. Uma vez que o servidor DNS foi tomado e configurado para responder com dados falsos é muito difícil detectar o ataque. Isso acontece por que para o usuário tudo parece legítimo, porém um ataque “man-in-the-middle” está ocorrendo.

A técnica que essa pesquisa objetiva identificar, é a de **Dns Tunnel(ing)**.

3 DNS Tunneling

DNS Tunneling é uma técnica utilizada para esconder determinada comunicação que utiliza protocolos baseados em IP sobre o protocolo DNS. Para melhor explana-la vou começar descrevendo o protocolo DNS.

3.1 Protocolo DNS

O protocolo DNS (Domain Named System) está descrito na RFC 1035. Segundo ela, seu objetivo é prover um mecanismo para denominar recursos, de tal forma que possam ser usados para identifica-los a partir de diferentes ativos, redes, famílias de protocolos, internets e organizações administrativas.

Hoje em dia o DNS é um dos protocolos mais utilizados no mundo, sendo uma das bases fundamentais da internet. É mais conveniente para um usuário gravar o nome de um domínio do que seu endereço IP. Um servidor DNS serve para fazer justamente essa tradução. O cliente solicita o endereço de determinado domínio e o servidor DNS responde essa solicitação com o IP desse domínio. A **figura 4** representa bem esse serviço:

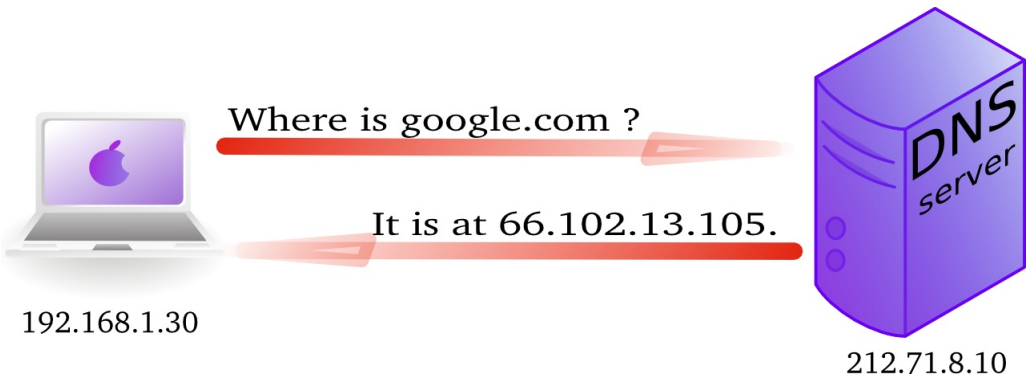

Figura 4: Requisição DNS

O protocolo DNS é baseado em UDP/IP e seu modelo de funcionamento é hierárquico. Quando é solicitado o endereço de um domínio que determinado servidor não possui o IP em sua base, ele envia essa solicitação para outro servidor. Essa hierarquia pode ser vista nitidamente na rede mundial de computadores. Veja uma ilustração dessa hierarquia na **figura 5**:

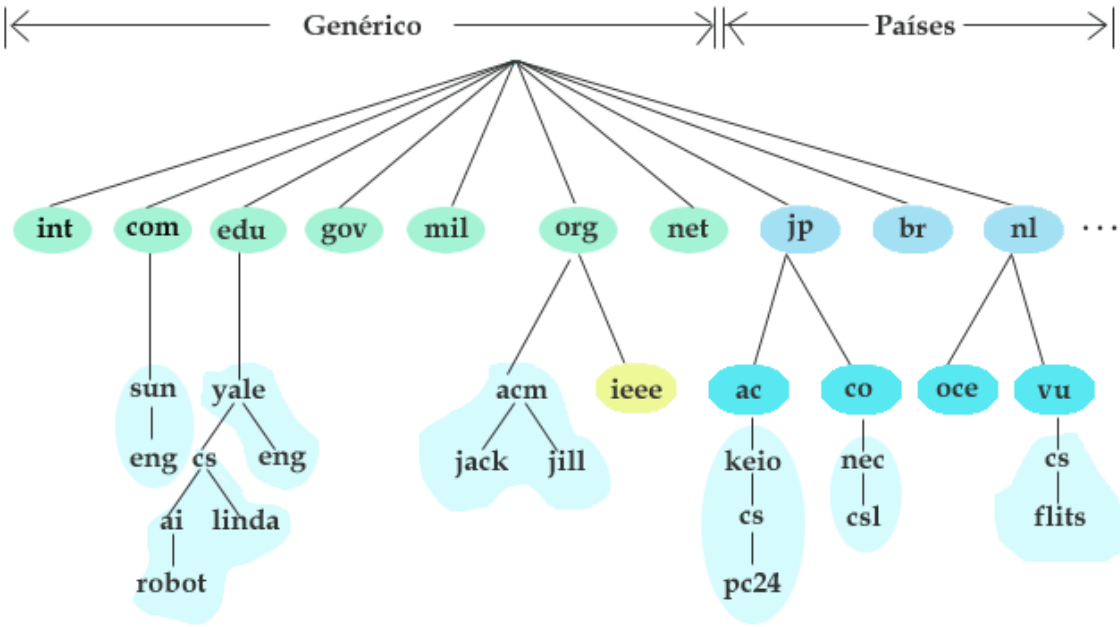

Figura 5: Hierarquia de servidores DNS na internet

3.2 Estrutura de pacote DNS

A estrutura de um pacote DNS é simples, ela segue o padrão do modelo Internet e foi desenhada exclusivamente para agregar informações necessárias para a tradução dos nomes de domínio em endereços IP. Veja essa estrutura na **figura 6**:

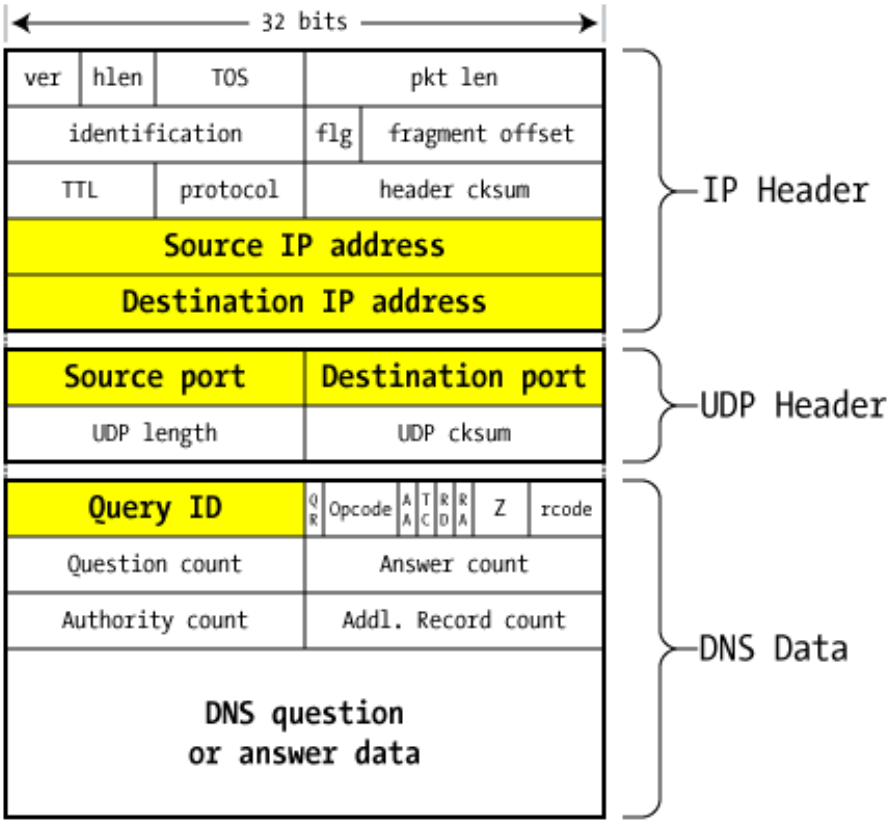

Figura 6: Estrutura de pacote DNS

Principais campos:

- **Source/Destination IP (Ip de Origem/Destino)**: Contém o endereço da máquina que enviou e deve receber o pacote (contendo a requisição)
- **Source/Destination Port Number (Numero da Porta de Origem/Destino)**: Por padrão o servidor DNS recebe requisições na porta UDP 53. O cliente abre uma porta randômica para realizar uma solicitação e recebe uma resposta na mesma porta.
- **Query ID (Identificado da Query)**: É o único campo que permanece intacto após o servidor DNS processar a solicitação de tradução de domínio para endereço. Justamente para gerenciar a sessão.
- **Question Record Count**: É o tipo de dado que controla o pacote (sendo usado para diferencia-los dos demais)
- **DNS Question/Answer data (Pergunta DNS/ dados da resposta)**: Esse é o local onde fica as requisições feitas ao servidor DNS e suas respostas ao cliente. Dentro da Query também é encontrado o seu tipo. Veja os mais utilizados na **tabela 1**:

TYPE	value and meaning
A	1 a host address
NS	2 an authoritative name server
CNAME	5 the canonical name for an alias
SOA	6 marks the start of a zone of authority
MX	15 mail exchange
TXT	16 text strings

Tabela 1: Tipos de Queries DNS

3.3 Escondendo a comunicação

Segundo o artigo "Detecting DNS Tunnneling" escrito por Greg Farnham e publicado pela Sans institute no dia 25 de fevereiro de 2013, o motivo do protocolo DNS ser alvo de ataques é a sua grande significância para com a rede global de comunicações (internet). Utilizando DNS Tunneling, outro protocolo pode ser escondido através do protocolo DNS. Essa técnica pode ser utilizada para servir de "Comando e Controle", vazamento de dados ou esconder qualquer trafego baseado em IP. Na apresentação de 2012 da conferência da RSA, Ed Skoudis declarou que "Comando e Controle" baseado em DNS é um dos 6 mais perigosos novos tipos de ataque. Ed disse ainda "Atacantes tem usado recentemente essa técnica em um caso de roubo de milhões de contas" (Skoudis, 2012).

A primeira discussão a respeito de DNS Tunneling foi feita por Oskar Pearson em Bugtrack mailing em 1998, desde então o número de ferramentas que utilizam DNS tunneling vem crescendo. As ferramentas desenvolvidas possuem técnicas semelhantes, cujo as suas maiores diferenças estão na codificação utilizada e nos detalhes de comunicação.

Um ataque de DNS Tunneling é constituído por 2 ativos, um servidor com um domínio DNS previamente preparado para receber as requisições de um cliente e preparar uma comunicação codificada entre eles, e um cliente com um agente compatível com a codificação utilizada pelo domínio malicioso. Quando o cliente disparar uma requisição ao servidor DNS malicioso, será criado um túnel pelo qual o cliente poderá acessar os serviços disponíveis pelo DNS malicioso se escondendo sobre o protocolo DNS. Sendo

assim dispositivos de proteção que permitam trafego DNS não impedem essa comunicação.

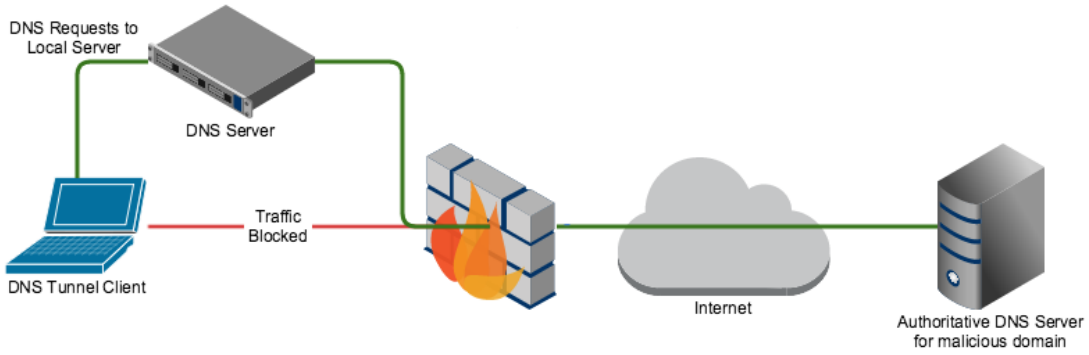

Figura 7: Estrutura de ataque DNS Tunneling

Exemplo de funcionamento:

1. Cliente faz a requisição sobre determinado conteúdo
2. Firewall bloqueia acesso a determinado conteúdo
3. Cliente utiliza agente DNS Tunneling para se comunicar com DNS malicioso
4. Firewall permite trafego DNS
5. DNS malicioso fecha túnel com agente
6. Cliente faz a requisição sobre determinado conteúdo
7. Conteúdo é escondido sobre protocolo DNS e é permitido pelo firewall

Toda comunicação feita com entre o cliente e o servidor malicioso é codificada. As ferramentas existentes fazem uso de várias técnicas para realizar essa codificação. Sendo as principais:

- **Base32 Encoding**: Codificação Base32 ou 5-bit é facilmente encontrada em DNS tunneling, principalmente para requisição de clientes. Enquanto o DNS faz diferenciação entre letras maiúsculas e minúsculas, essa codificação os ignora, deixando 37 caracteres validos (com números e caracteres especiais validos) para troca de informação.
- **Base64 Encoding**: Codificação Base64 ou 6-bit pode ser usado como respostas do tipo 'TXT' por parte do DNS malicioso. Como essa codificação faz diferenciação entre letras maiúsculas e minúsculas, ela possui o total de 64

caracteres validos (com números e caracteres especiais validos) para troca de informação.

- **Binary (8 bit) Encoding**: Codificação Binary 8-bit pode ser usada. Os autores de Heyoka descobriam que embora ele não funcione com todos os servidores DNS, eles podem utilizar 8 bits por caractere para codificar a troca de comunicação sobre o protocolo DNS utilizando redes com grande bandwidth (Reveli, 2009).
- **NetBios Encoding**: Esse método pode é utilizado pela ferramenta DNScat-B. Cada bit é o que equivale a 2 caracteres, cada bit é separado em 4 pedaços, um decimal de base 65 é adicionado para cada pedaço.
- **Hex Encoding**: Esse método pode é utilizado pela ferramenta DNScat-B. Para codificação Hex, dois caracteres hexadecimais são utilizados para representar cada Bit.

As ferramentas mais conhecidas para realizar DNS tunneling são:

- **DeNiSe**: DeNiSe é uma prova de conceito de tunelamento IP sobre DNS escrito em Python. DeNiSe tem um Github que possui seus scripts desenvolvidos de 2002 a 2008 (mdornseif, 2002).
- **Dns2tcp**: Dns2tcp foi desenvolvido por Olivier Dembour e Nicolas Collignon, Este Software é escrito em C e roda no sistema operacional Linux (com cliente com suporte a Windows). Ele suporta requisições DNS do tipo KEY e TXT (Dembour, 2008).
- **DNScapy**: DNScapy foi desenvolvido por Pierre Bienaime. O DNScapy suporta tunelamento SSH sobre o protocolo DNS. Ele pode ser configurado para usar requisições DNS do tipo CNAME e TXT, ou ambos de forma randômica (Bienaime, 2011).
- **DNSCat (DNScat-P)**: Originalmente lançado em 2004 com uma atualização em 2005. Este software foi escrito por Tadeusz Pietraszek. DNSCat é conhecido por ser um “canivete suíço” por envolver comunicação bidirecional sobre o protocolo DNS. DNScat foi escrito em Java e roda sobre o sistema operacional Linux, suportando requisições DNS do tipo CNAME (Pietraszek, 2004).
- **DNSCat (DNScat-B)**: Este software foi escrito por Ron Bowes. A sua versão publica mais conhecida foi lançada em 2010 para os sistemas operacionais Linux,

Mac Os e Windows. Ele utiliza codificação NetBios e Hex, suportando requisições DNS do tipo CNAME, NS, TXT e MX. (Bowes, 2010).

- **Heyoka**: Heyoka é uma prova de conceito que cria um túnel bidirecional para infiltração de dados. Esta ferramenta foi escrita em C e foi testada no sistema operacional Windows. Ela utilizada codificação Base32 e base64 e utiliza EDNS para suportar requisições DNS (Revelli, 2009).
- **Iodine**: É um programa de tunelamento DNS lançado primeiramente em 2006 com atualização em 2010. Ele foi escrito por Bjorn Andersson e Erick Ekman em C, sendo suportado pelos sistemas operacionais Linux, Mac Os e Windows. Atualmente ele foi portado para Android (Andersson, 2010).
- **Squeeza**: Squeeza é uma ferramenta de SQL Injection, ela é dividida em canal de controle e canal de infiltração. O canada de comando pode ser usado para criar dados em banco de dados e executar comando de controle. Ela suporta infiltração de canais: http erros, timing e DNS (Haroon, 2007).

3.4 Análise de DNS Tunneling

Segundo o artigo publicado pela Sans em 25 de fevereiro de 2013 escrito por Greg Farnham cujo título é "Detecting DNS Tunneling" uma das maneiras mais eficientes de se detectar DNS Tunneling é através da verificação de comportamento. A análise de comportamento busca mapear comportamento padrão e encontrar anomalias, ou seja, determinadas características que fogem o comportamento considerado "normal". Como o DNS foi criado para facilitar a localização de ativos, nomes simples e/ou hierarquicamente padronizados de domínios são mais comuns, logo um domínio com nome aleatório foge a esse conceito. Para determinar o nível de aleatoriedade do nome dos domínios consultados, será utilizado o calculo de entropia de Shannon. Usar Shannon para detectar anomalias no trefego é considera uma técnica recente, apresentadas por Lakhina em 2005 no seu artigo "Detecting Distributed Attacks using Network-Wide Flow Traffic" e por MacKey em 2003 no seu livro "Information Theory, Inference, and Learning Algorithms".

3.4.1 Entropia de Shannon

O calculo de entropia proposto por Claude Elwood Shannon foi apresentada no artigo "A Mathematical Theory of Communication" escrito por em julho de 1948, uma versão desse artigo foi publicada em The Bell System Technical Journal. Veja a formula na **figura 8**:

$$H = -\sum_{i=1}^{S} (Pi)(Log2Pi)$$

Figura 8: Formula de Shannon

H - Entropia
Pi - Corresponde a quantidade de repetições de um objeto em cadeia de objetos, dividido pelo total de objetos dessa cadeia
S - Total de objetos

O cálculo de Shannon é utilizado para quantificar a incerteza sobre uma variável aleatória. Esse calculo foi implementado em forma de algoritmo e será utilizado para quantificar a aleatoriedade em uma frase. Veja o algoritmo na imagem **figura 9**:

```
def shannon(word):
    entropy = 0.0
    length = len(word)

    occ = {}
    for c in word :
        if not c in occ:
            occ[ c ] = 0
        occ[c] += 1

    for (k,v) in occ.iteritems():
        p = float( v ) / float(length)
        entropy -= p * math.log(p, 2) # Log base 2
    return entropy
```

Figura 9: Algoritmo de Shannon

Segue um passo a passo do funcionamento do algoritmo:

1. Recebe a frase que será calculada a entropia.
2. Inicializa a variável "entropy" com 0, a variável "length" calcula o tamanho da frase recebida e a variável "occ" como um dicionário.

3. Cria uma um laço de repetição selecionando cada letra (fazendo diferenciação de maiúsculo e minúsculo) dentro da frase.
4. Se for a primeira aparição da letra ela é adicionada como chave no dicionário e é inicializado com o valor 0.
5. A chave do dicionário é identificada e seu valor é acrescentado em 1.
6. Cria um laço de repetição que recupera a chave e o valor para todos os itens do dicionário.
7. A variável "p" recebe o valor da chave do dicionário (a quantidade de vezes que a letra aparece na frase) dividido pelo tamanho da frase.
8. A variável "entropy" soma -(p * log (p) na base 2)) ao seu valor
9. Retorna o valor da variável "entropy"

3.4.2 Obtenção do fluxo de pacotes DNS

A obtenção do fluxo de pacotes DNS para a quantificação de sua aleatoriedade utilizando Shannon é feita através do monitoramento do canal de comunicação em que se deseja identificar anomalias. Existem diversas maneiras de se monitorar uma rede, sempre dependendo de seu arquitetura e nível de detalhes que se espera obter. Uma das maneiras mais comuns é a utilização Sniffers.

Segundo o artigo "Analysis of Various packet sniffer tools" escrito por Pallavi Asrodia e Hemlata Patel em 15 de maio de 2012 e publicado em International Journal of Electrical, Electronics and Computer Engineering, um Sniffer é uma técnica utilizada para monitorar cada pacote que cruza a infraestrutura de rede. O Sniffer é um pedaço de hardware ou software que monitora todo o trafego da rede ou apenas um complemento que monitora o trafego que é enviado para ele ou o ativo que o hospeda (sendo importante salientar que ele monitora a entrada e saída de pacotes). Ainda segundo o artigo, um Sniffer funciona da seguinte maneira:

1. O Sniffer coleta dados binários brutos a partir da rede monitorada.
2. O dado capturado é convertido em um formato que pode ser lido/manipulado.
3. Separação dos dados capturados por protocolos e análise de seus metadados dependo de sua estrutura.

3.4.3 Análise do fluxo de pacotes DNS

Utilizando o Sniffer em conjunto com uma ferramenta de Big Data é possível Extrair, Normalizar, Armazenar e Analisar cada pacote DNS que trafega pela rede que esta sendo monitorada. Cada pacote DNS possui uma query com o nome do domínio que o cliente quer saber seu IP. Esses domínios possuem nomes que refletem características da marca ou da empresa que o disponibiliza, ou seja, nome que possuem planejamento por trás de sua origem. Utilizando o calculo de Shannon em um desses nomes de domínio padronizados, é notório que o seu valor é baixo. Veja na imagem **figura 9**:

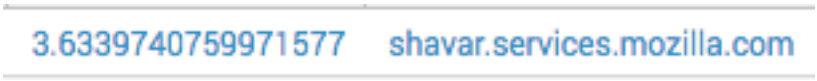

Figura 9: Valor de Shannon para query normal

Agora quando é realizado o mesmo procedimento em uma query que faça parte de uma troca de comunicação indesejada utilizando DNS Tunneling, foi identificado que o valor do calculo de Shannon é sempre acima dos demais. Veja na imagem **figura 10**.

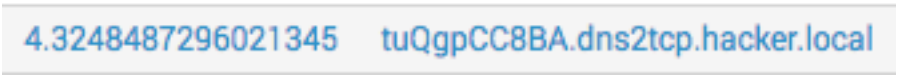

Figura 10: Valor de Shannon para query maliciosa

Isso acontece por que a informação é passada de forma criptografada dentro da parte de dados do pacote DNS, fazendo a entropia da query fique mais alta. Logo, será utilizado o calculo de Shannon e quantificado a entropia de queries DNS para identificar frequências significantes de valores pacotes trafegado com valores altos (resultantes desse calculo).

4 Estudo de caso

4.1 Construção de ambiente

Todo ambiente de teste foi construído em cima do VMWARE fusion. Para o estudo de caso foram criadas 5 máquinas virtual: um Firewall, um servidor DNS (Bind9), um servidor DNS Malicioso, uma máquina para ser usada como usuário e um servidor Splunk. Veja a estrutura completa na imagem **figura 11**:

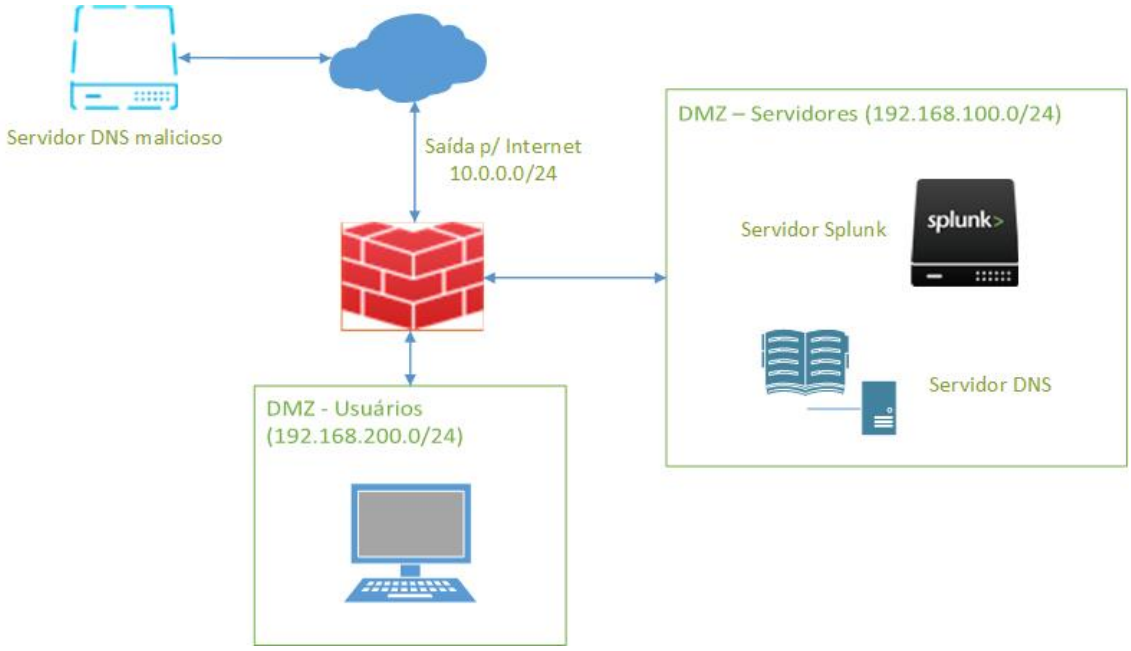

Figura 11: Estrutura de laboratório

Segue o procedimento de configuração para cada dispositivo desse laboratório:

- **Configuração do Firewall**

A estrutura da rede foi desenhada para centralizar todo o tráfego no Firewall, fazendo ele também o papel de roteador, permitindo ou não a comunicação da DMZ com a rede de usuários, ou a saída da rede de usuários, da DMZ para a internet e cuidando das restrições de acesso.

O firewall foi configurado possuindo 3 interfaces, eth0 na rede 10.0.0.0/24 com IP dhcp que permite saída para a internet, eth1 na rede 192.168.200.0/24 com IP 192.168.200.5 que não permite acesso direto a internet e eth2 na rede 192.168.100.0/24 com o IP 192.168.100.5 que não permite acesso direto a internet. A **figura 12** traz a tabela de roteamento:

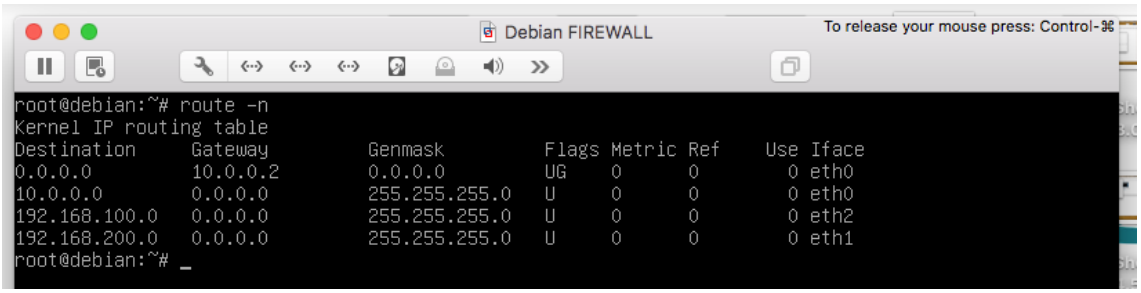

Figura 12: Tabela de roteamento do Firewall

As regras de acesso do Firewall foram feitas para permitir que qualquer pacote que entre pelas interfaces eth1 e eth2 com destino que não sejam as redes 10.0.0.0/24, 192.168.100.0/24 e 192.168.200.0/24, sejam nateadas para um IP da rede 10.0.0.0/24 e tenham acesso a internet.

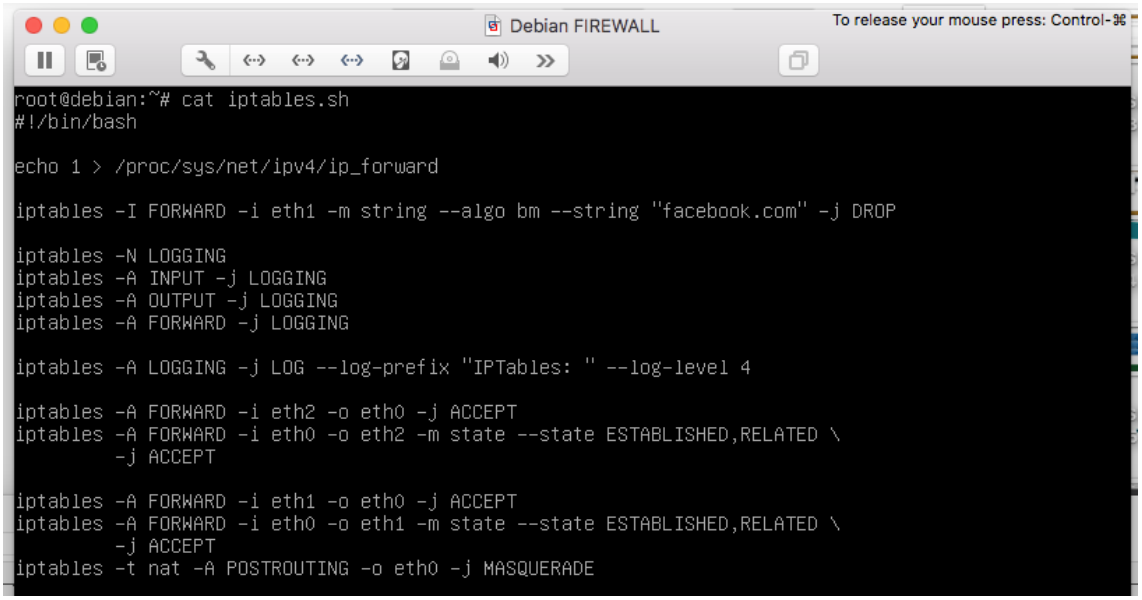

Figura 13: Regras do Firewall

Foi instalado um agente do Splunk dentro do Firewall. Esse agente contém um Sniffer que captura todos os pacotes DNS que passam pelo Firewall e o agente os envia para o servidor do Splunk.

```
root@debian:/opt/splunkforwarder/bin# ./splunk status
splunkd is running (PID: 862).
splunk helpers are running (PIDs: 863 912).
root@debian:/opt/splunkforwarder/bin#
```

Figura 14: Status do agente Splunk no Firewall

- **Configuração do DNS (Bind9)**

Um servidor DNS foi configurado para responder as requisições DNS que são feitas pelos ativos que estão dentro da DMZ de usuários. Ele opera na porta 53 e quando não conhece o domínio especificado envia a requisição para o servidor que está no endereço IP 8.8.8.8.

```
options {
        listen-on port 53 { any; };
        #listen-on-v6 port 53 { ::1; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query { any; };

        recursion yes;

        forwarders {
                192.168.200.205;
                8.8.8.8;
        };

        forward only;
```

Figura 15: Configuração gerado do DNS

O nome do domínio do servidor DNS é “tccunb.local”. Veja na **figura 16**:

```
zone "tccunb.local" IN {
        type master;
        file "tccunb.local";
        allow-update { none; };
};
```

Figura 16: Configuração do DNS para "tccunb.local"

- **Configuração do servidor DNS Malicioso**

O servidor utilizado para realizar DNS Tunneling foi configurado na rede 10.0.0.0/24 que permite acesso a internet. Qualquer servidor nas redes 192.168.100.0/24 e 192.168.200.0/24 que queiram se comunicar com esse servidor, passará pelo Firewall e ele realizara um NAT para que essa comunicação seja possível. Como é apresentado na imagem a seguir, o IP 10.0.0.170 foi configurado de forma estática. Veja na **figura 17**:

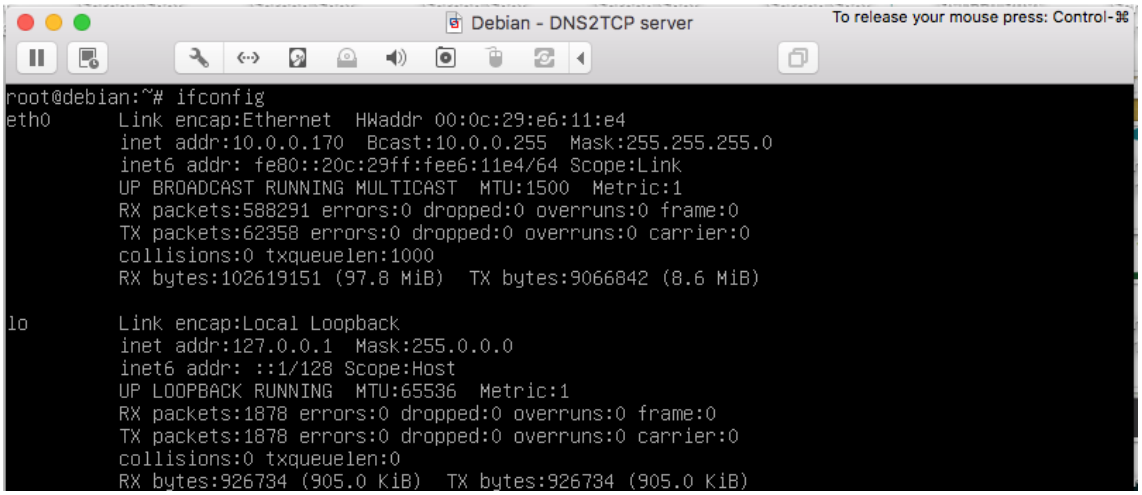

Figura 17: Interfaces do servidor DNS malicioso

- **Configuração do Usuário**

Uma máquina com o sistema operacional Kali Linux foi instalada dentro da DMZ de usuários. Ela fará o papel do cliente e a partir dela será feito a conexão utilizando a técnica de DNS tunneling. Ela foi configurada com o IP 192.160.200.200/24. Veja a **figura 18**:

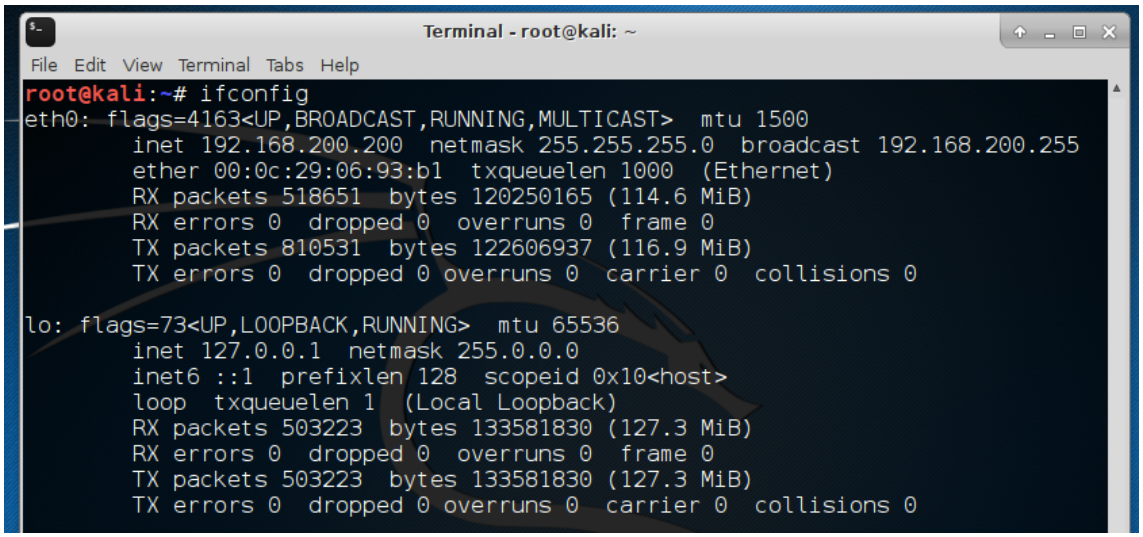

Figura 18: Interfaces do cliente

A rota default dessa máquina é o Firewall e foi configurado como servidor DNS o ativo com o IP 192.168.100.10 (Ip do DNS na DMZ de servidores). Veja a **figura 19**:

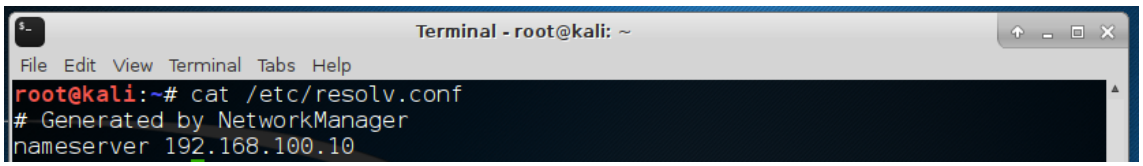

Figura 19: Servidor DNS configurado no cliente

- **Configuração do Servidor Splunk**

Um servidor para suportar o Splunk foi instalado, ele possui o sistema operacional CentoOS e foi configurado na DMZ de servidores. Foi configurado com o IP estático 192.168.100.250. Veja a **figura 20**:

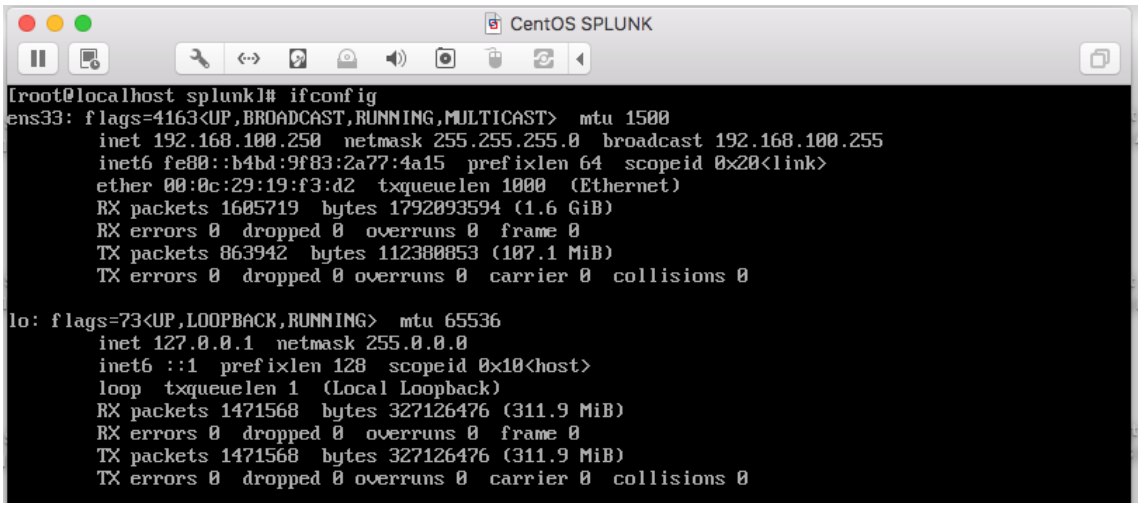

Figura 20: Interfaces do servidor Splunk

A instalação do Splunk foi feita com o procedimento padrão definido em sua documentação, baixado diretamente do site e instalado no /opt. Veja a **figura 21**:

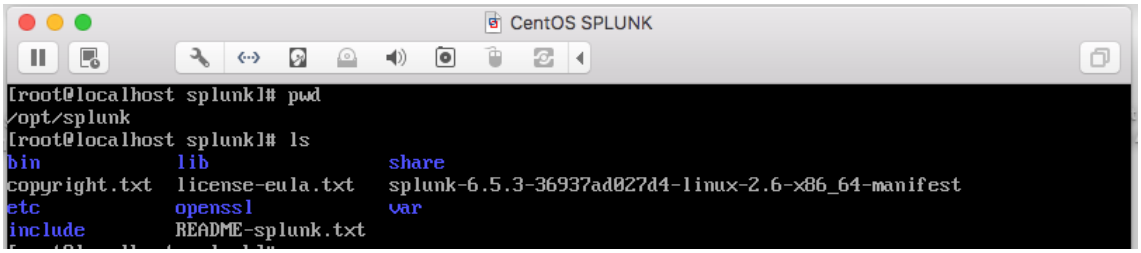

Figura 21: Splunk instalado

4.2 Burlando Firewall e capturando pacotes

Esse procedimento foi todo realizado para fins didáticos. Foi criado uma regra no Firewall para negar o acesso ao facebook os ativos que pertencem a DMZ de usuários. Veja a regra na **figura 22**:

```
iptables -I FORWARD -i eth1 -m string --algo bm --string "facebook.com" -j DROP
```

Figura 22: Regra de bloqueio para o Facebook

Os Clientes que tentarem acessar qualquer site que tenha em sua url "facebook.com" terá a sua requisição "Dropada". Sendo assim o browser vai enviar a requisição, porém não vai obter resposta.

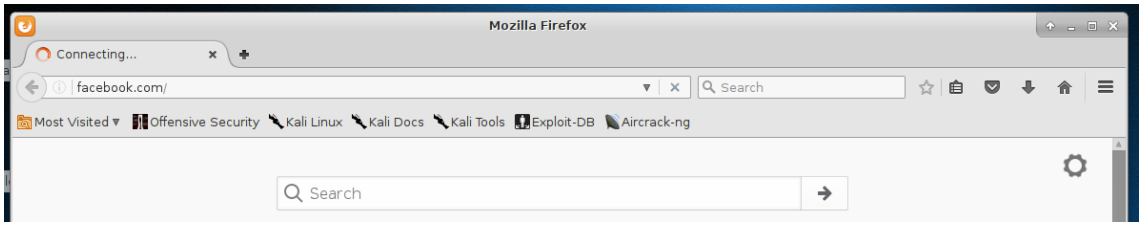

Figura 23: Tentativa fracassada de acesso ao Facebook

Para burlar esse Firewall, será utilizar a técnica de DNS tunneling. Primeiramente foi configurado um servidor fora da infraestrutura que o Cliente está e que possui acesso total a internet. Foi utilizado a ferramenta dns2tcp para criar um servidor de DNS tunneling. Veja a **figura 24**:

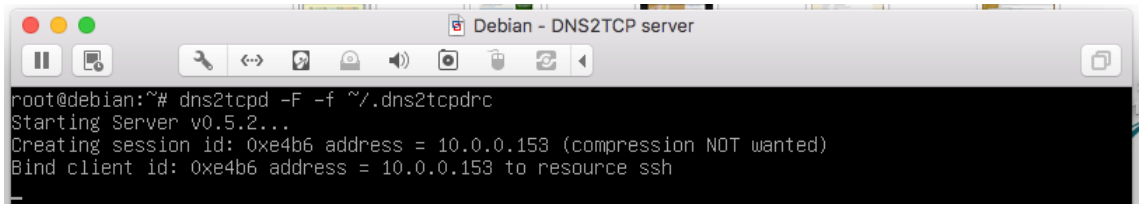

Figura 24: Inicialização de Servidor Malicioso com ferramenta DNS2TCP

O cliente vai utilizar a ferramenta dns2tcp para se comunicar com o servidor. O servidor estava aguardando essa comunicação, ele vai fechar um DNS tunneling e vai esconder tráfego SSH (que funciona baseado em protocolo IP). Uma conexão direta com o servidor malicioso vai estar disponível via SSH na porta 4430.

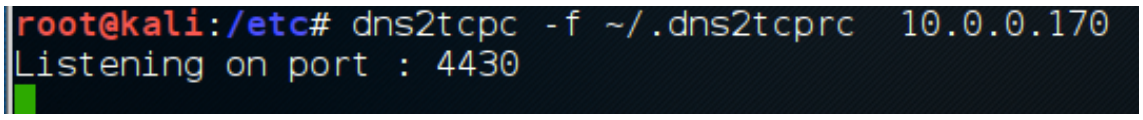

Figura 25: Inicialização de agente com ferramenta DNS2TCP

Após fechado o DNS tunneling é preciso realizar uma conexão SSH local na porta 4430, que vai dar acesso ao túnel DNS criado. Assim qualquer navegação feita por esse túnel será na verdade feita pelo servidor malicioso e enviada por dentro do túnel para o cliente.

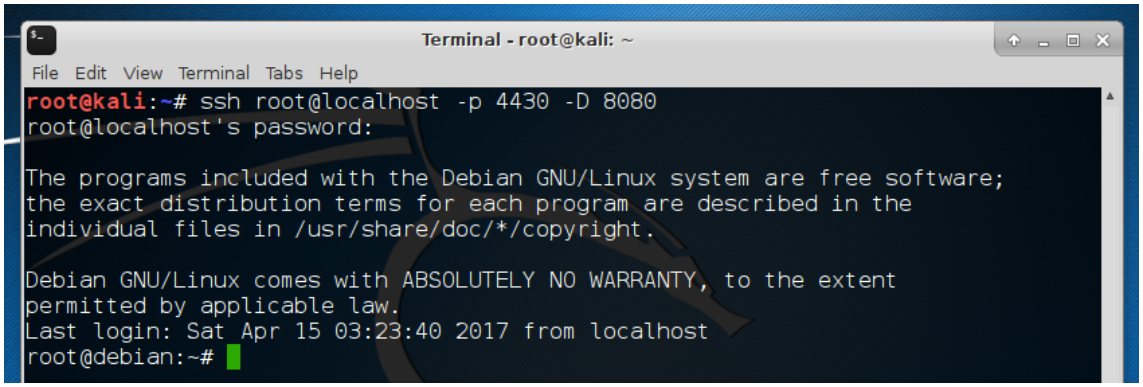

Figura 26: Acesso SSH local e abertura de porta 8080

Usando o ssh é possível criar uma conexão e abrir uma porta para trafegar dados http/https por essa conexão. Para usar um browser para utilizar essa porta para sair para internet é necessário configurar o proxy especificando o IP local e a porta aberta.

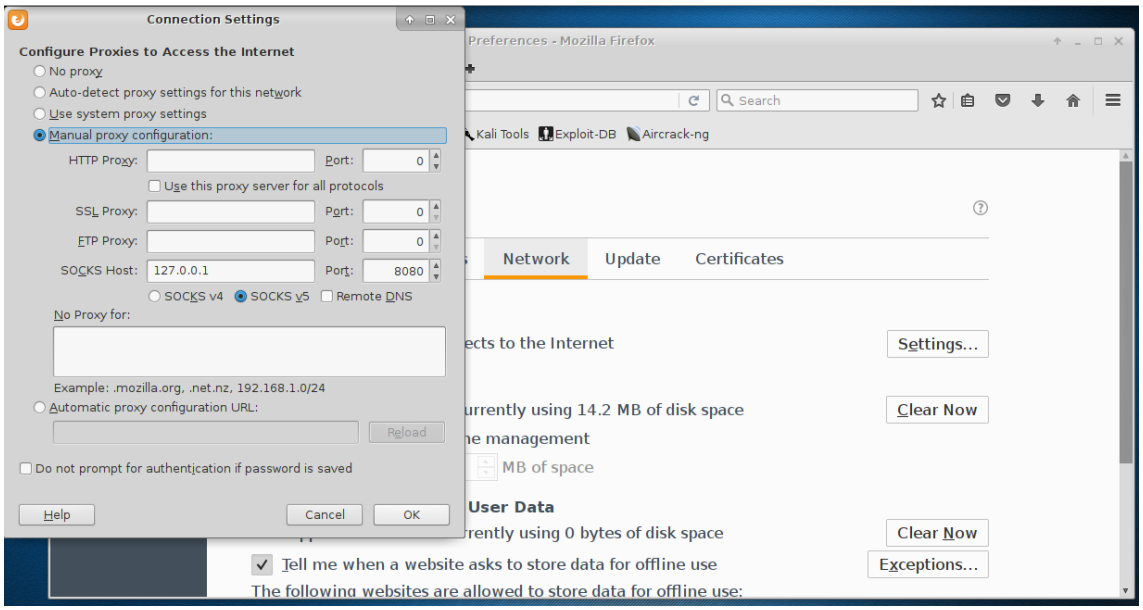

Figura 27: Configuração de Proxy para porta 8080

Após realizar essa configuração o facebook vai poder ser acesso normalmente. Burlando assim as regras de segurança da infraestrutura.

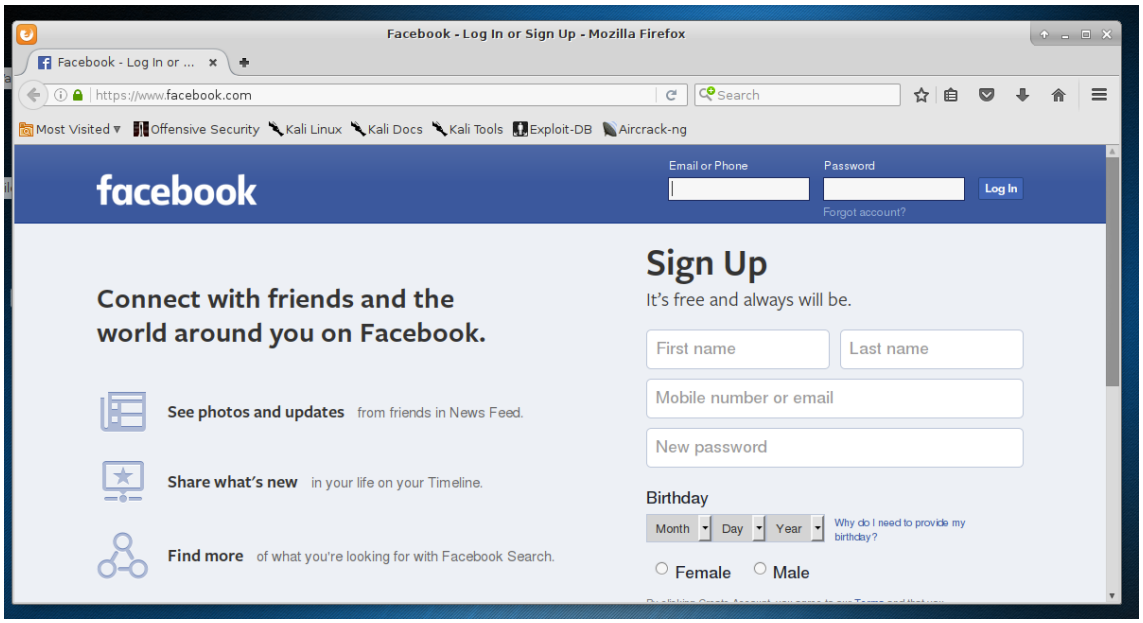

Figura 28: Tentativa de acesso bem sucedida ao Facebook

Todo o trafego que sai dos clientes da DMZ de usuários passa pelo Firewall. Dentro do Firewall existem um agente Splunk com um Sniffer que foi configurado para capturar todos os pacotes do tipo DNS e enviar para o servidor do Splunk.

```
root@debian:/opt/splunkforwarder/bin# ./splunk list forward-server
Active forwards:
        192.168.100.250:9997
Configured but inactive forwards:
        None
```

Figura 29: Agente de Splunk no Firewall enviando dados para Servidor do Splunk

4.3 Detecção de DNS Tunneling

Um pacote capturado pelo Sniffer dentro do Firewall é transformado em Json e apresentado de forma que possibilite entendimento humano. Veja um exemplo na **figura 30**:

```
{ [-]
   bytes: 40
   bytes_in: 40
   bytes_out: 0
   capture_hostname: debian
   dest_ip: 224.0.0.251
   dest_mac: 01:00:5E:00:00:FB
   dest_port: 5353
   endtime: 2017-05-12T11:43:37.181558Z
   flow_id: 666c5254-a2d8-4d83-9519-740bd630c8d9
   message_type: [ [+]
   ]
   network_interface: eth0
   packets_in: 1
   packets_out: 0
   protocol_stack: ip:udp:dns
   qdcount: 1
   query: [ [-]
     _googlecast._tcp.local
   ]
   query_type: [ [-]
     PTR
   ]
   src_ip: 10.0.0.1
   src_mac: 00:50:56:C0:00:03
   src_port: 5353
   time_taken: 0
   timestamp: 2017-05-12T11:43:37.181558Z
   transaction_id: 0
   transport: udp
}
```


Figura 30: Pacote DNS dentro do Splunk

Foi realizado um tratamento para extrair as informações e liga-las a determinada chave, criando um dicionário. Exemplo “src_ip” contém o valor “10.0.0.1”. Foi utilizado a linguagem de pesquisa da ferramenta baseada nos campos extraídos e cruiado 1 Dashboard contendo 6 painéis baseados no conteúdo dos pacotes DNS. Para cada painel é foi necessário a cronstrução de uma pesquisa que visa apresentar informações estatísticas dos dados explorados.

Dentro do Splunk foi criado uma App para englobar as pesquisas realizadas:

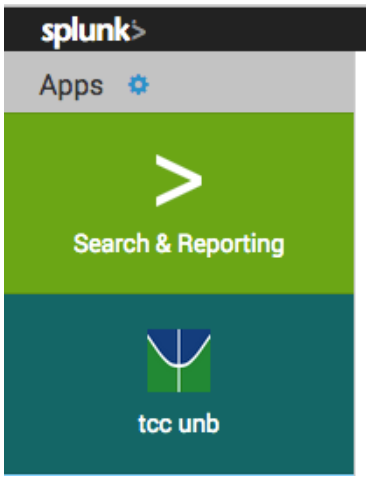

Figura 31: App "TCC UNB"

Dentro dessa App está o Dashboard com todos os painéis criados. Veja a seguir a descrição de cada painel e sua respectiva pesquisa:

- **Pesquisa 1**

Essa pesquisa traz a quantidade de máquinas que firam DNS Tunneling no tempo pesquisado. Foi realizado o calculo de entropia para cada query contida nos pacotes DNS, aqueles que tiveram o valor acima de 4 por mais de 10 vezes (constatando trafego e não um caso isolado), tem seu IP contabilizado.

Dashboard:

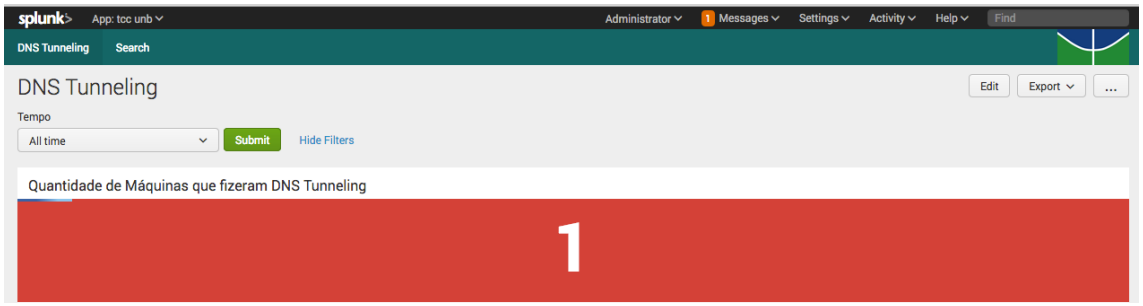

Figura 32: Quantidade de máquinas fizeram DNS Tunneling

Pesquisa Realizada:

*host=debian sourcetype="stream:dns" source="stream:dns" src_ip=192.168.200.**
| rename "query{}" as query
| `ut_shannon(query)`
| search ut_shannon > 4
| stats count(query) as quantidade by src_ip
| search quantidade > 10
| stats count(src_ip)

- **Pesquisa 2**

Esse Painel em forma de Pizza mostra a porcentagem dos IPs que receberam pacotes DNS com query que receberam nota maior que 4 após calculo de Shannon.

Dashboard:

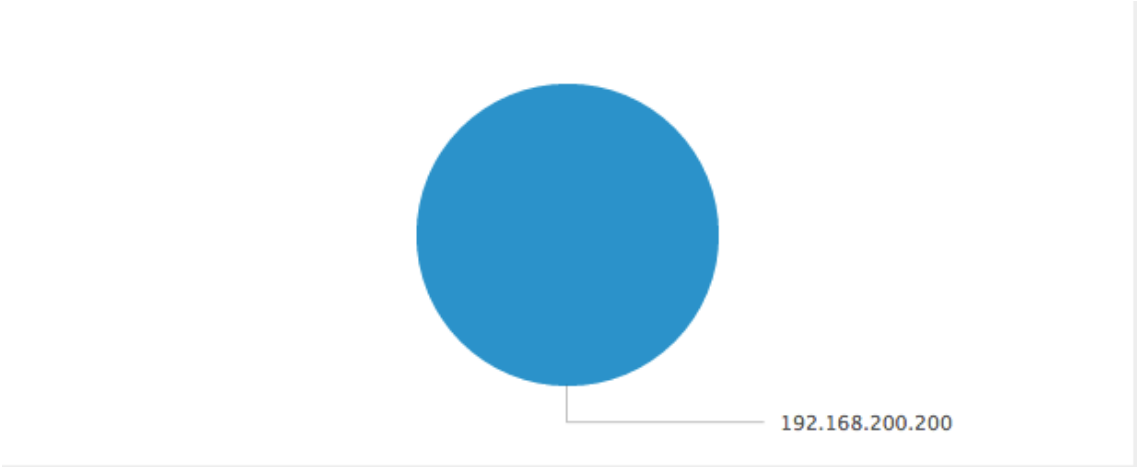

Figura 33: Top Ip Origem

Pesquisa Realizada:

*host=debian sourcetype="stream:dns" source="stream:dns" src_ip=192.168.200.**

| rename "query{}" as query

| `ut_shannon(query)`

| search ut_shannon > 4

| top src_ip

- **Pesquisa 3**

Esse Painel em forma de Pizza mostra a porcentagem dos IPs que enviaram pacotes DNS com query que receberam nota maior que 4 após calculo de Shannon.

Dashboard:

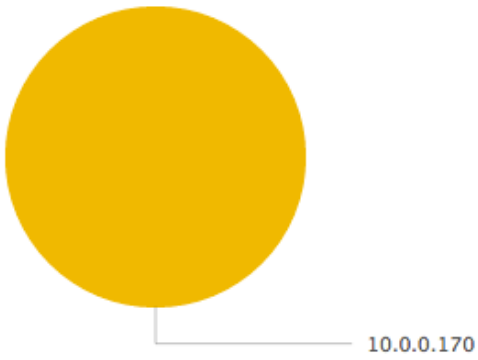

Figura 34: Top Ip Destino

Pesquisa Realizada:

*host=debian sourcetype="stream:dns" source="stream:dns" src_ip=192.168.200.**
| rename "query{}" as query
| `ut_shannon(query)`
| search ut_shannon > 4
| top dest_ip

- **Pesquisa 4**

Esse Painel em forma de linha temporal mostra a media de entropia (por minuto) entre a transação de pacotes de DNS. A linha vermelha marca até qual nível a média dos valores das queries (feitas utilizando calculo de Shannon) é considerada calculo saudável.

Dashboard:

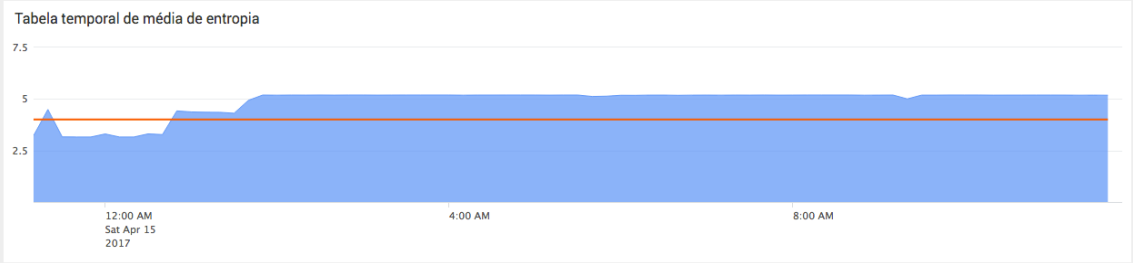

Figura 35: Linha temporal com medido de DNS Tunneling

Pesquisa Realizada:

*host=debian sourcetype="stream:dns" source="stream:dns" src_ip=192.168.200.**
| table _time, src_ip, dest_ip, dest_port, "query{}"
| rename "query{}" as query
| `ut_shannon(query)`
| timechart avg(ut_shannon) as count
| eval average=4

- **Pesquisa 5**

Esse Painel traz uma tabela de rastreabilidade dos pacotes DNS armazenados. Ele contém os seguintes campo: “Horário da transação do pacote”, “Ip de Origem”, “Ip de Destino”, “Porta de destino”, “Valor do calculo de Shannon” e “Query DNS”.

Dashboard:

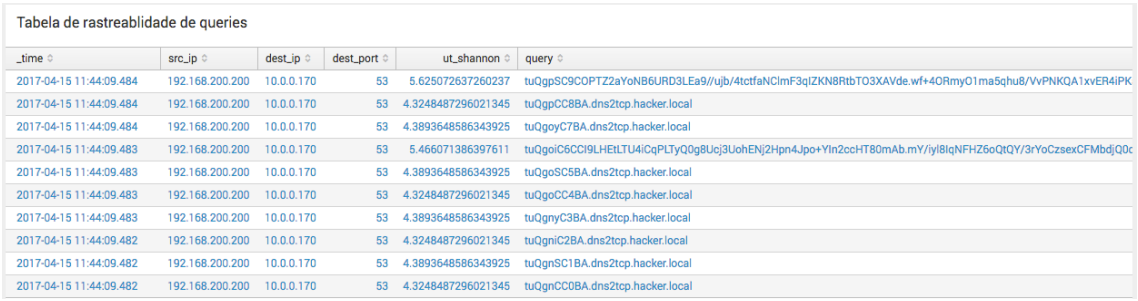

Figura 36: Tabela de rastreabilidade de Queries

Pesquisa Realizada:

*host=debian sourcetype="stream:dns" source="stream:dns" src_ip=192.168.200.**

| rename "query{}" as query

| `ut_shannon(query)`

| search ut_shannon > 4

| dedup query

| table _time, src_ip, dest_ip, dest_port, , ut_shannon, query

- **Pesquisa 6**

O seguinte painel traz informações geográficas sobre IPs dos servidores que são respondidos durante as requisições DNS. Quando algum cliente solicita o endereço de um domínio, o servidor DNS responde o IP. Esse painel é baseado nesse IP respondido.

Dashboard:

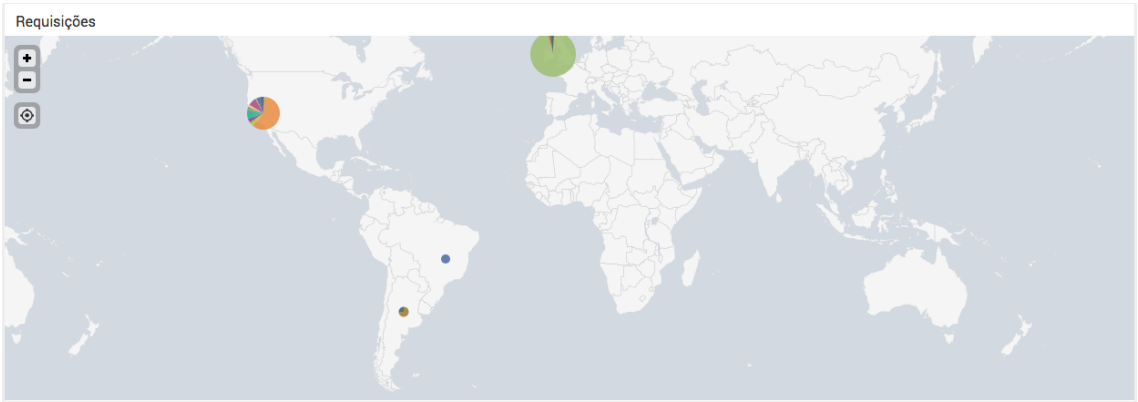

Figura 37: Geolocalização de Ips respondidos

Pesquisa Realizada:

host=debian sourcetype="stream:dns" source="stream:dns"

| rename "host_addr{}" as ip

| iplocation ip

| geostats count by ip locallimit=99999

4.4 Considerações Finais

Baseado no processo descrito, foi constatado que quanto maior o nível de entropia mais provável que os pacotes DNS estejam corrompidos e/ou queiram esconder alguma comunicação sobre o protocolo DNS.

Após observação do comportamento dos pacotes em ambiente controlado, foi comprovado que uma sequencia de pacotes com queries cujo valor do calculo do shannon seja maior que 4 e que sejam provenientes do mesmo servidor DNS e enviados para o mesmo cliente, se trata de um caso de DNS tunneling.

5 Conclusão

Dados provenientes de máquinas são uma das 5 fontes de informação com maior valor. A utilização de ferramentas de Big Data para analisar esse tipo de dado é amplamente adotada e dissimulada. Foi utilizado uma ferramenta com essa metodologia para extrair, normalizar, armazenar e extrair valores provenientes do trafego DNS. Tendo como objetivo principal verificar a estrutura dos pacotes DNS, buscando por anomalias que possam servir como base para identificar DNS Tunneling.

Após a realização dos procedimentos necessários para a iniciar a construção e manipulação de objetos, foi constatado que as queries contidas nos pacotes DNS normais são padronizadas e as que fazem parte de DNS Tunneling (devido a criptografia da técnica maliciosa) possui um nível de aleatoriedade maior. Sendo assim, foi utilizado a técnica de Shannon para quantificar o nível de aleatoriedade dessas queries. Após observar os valores provenientes do cálculo de Shannon sobre as queries, foi constatado que pacotes com o valor acima de 4 e que fazem parte de sequencia considerável de transações entre dois ativos, se trata (na maioria os casos) de DNS Tunneling.

6 Apêndice

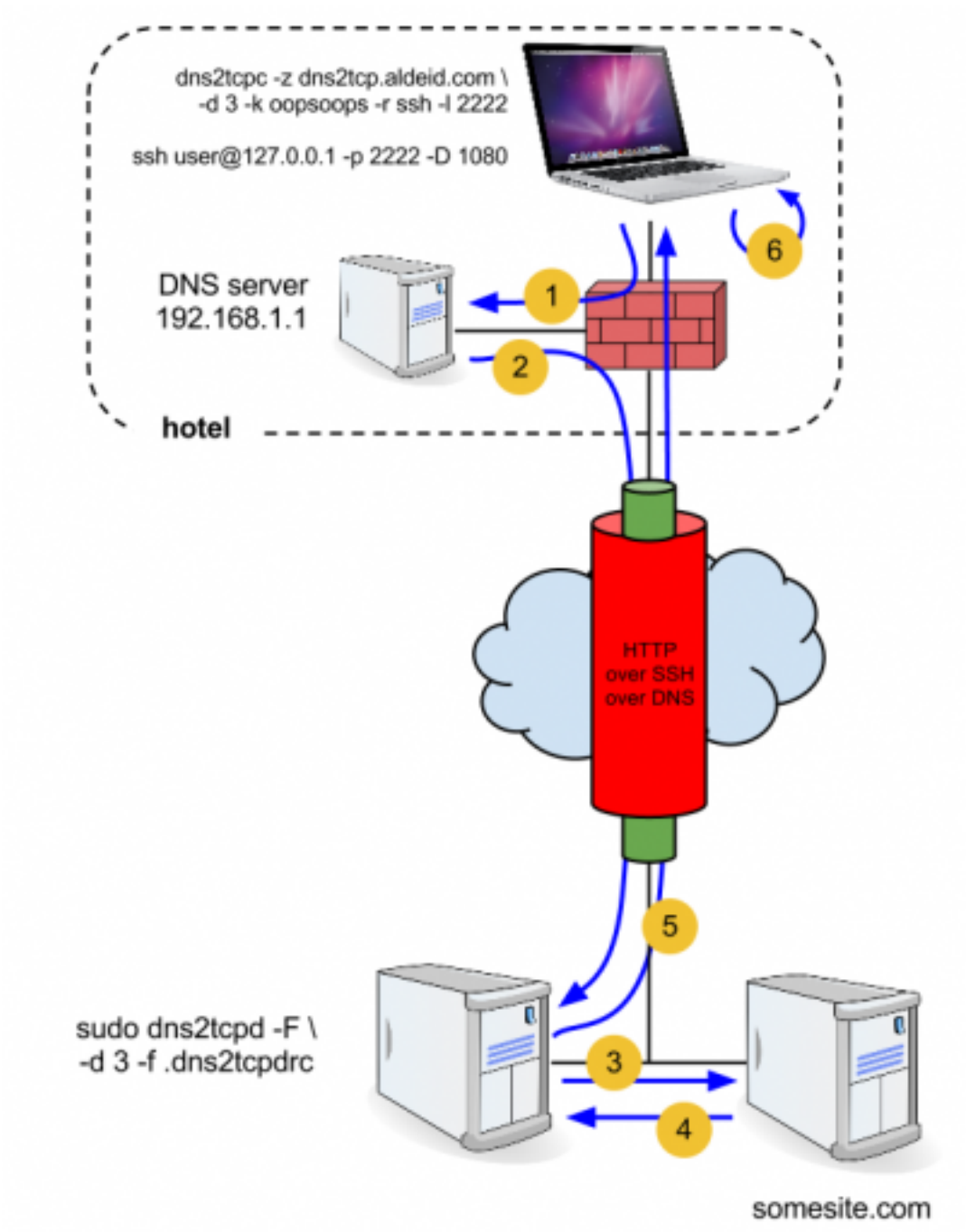

Figura 38: Modelo de DNS Tunneling usando Dns2tcp

7 Referências Bibliográficas

Ben Walker. “Every data Big Data Statics”. Vouchercloud, 2015. Disponível em: <http://www.vcloudnews.com/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/> - Acesso em: 17 de setembro de 2016.

Suzanne Wu. “How much information is the in the world”. Unversity of Southern Carlifornia, 2011. Disponível em: <http://news.usc.edu/29360/how-much-information-is-there-in-the-world/> - Acesso em: 19 de setembro de 2016.

P. Mockapetris. “Domain Names – Implementation and Specification”. Network Working Group, 1987. Disponível em: <https://www.ietf.org/rfc/rfc1035.txt> - Acesso em: 17 de março de 2017.

Greg Farnham. “Decting DNS Tunneling”. Sans Institute, 2013. Disponível em: <https://www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling-34152> - Acesso em: 17 de março de 2017.

Sebastien Tricaud. “When entropy meets Shannon”. Splunk Blog, 2016. Disponível em: <https://www.splunk.com/blog/2016/04/21/when-entropy-meets-shannon/> - Acesso em: 19 de março de 2017.

Rick Wanner. “Using Splunk to Detect DNS Tunneling”. Sans Institute, 2016. Disponível em: <https://www.sans.org/reading-room/whitepapers/dns/splunk-detect-dns-tunneling-37022> - Acesso em: 19 de abril de 2017.

Pallavi Arodia e Hemlata Patel. “Analysis of Various Packet Sniffing Tools for Network Monitoring and Analysis”. Internacional Journal of Electrical, 2012. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.429.567&rep=rep1&type=pdf> - Acesso em: 21 de abril de 2017.

Gil Press. “A very Short History of Big Data”. Forbs, 2013. Disponível em: <https://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/#3dd6591365a1> - Acesso em: 01 de maio de 2017.

Makoto Yasuda, Yasuharu Namba e Jun Yoshida. “New Business Trends Created by Big Data Utilization”. Hitachi Review, 2016. Disponível em: <http://www.hitachi.com/rev/pdf/2014/r2014_01_101.pdf> - Acesso em: 04 de maio de 2017.

Splunk,.“Instalation Manual”. Splunk Documentation. Disponível em: <http://docs.splunk.com/Documentation/Splunk/6.5.3/Installation/InstallonLinux> - Acesso em: 05 de maio de 2017.

Anant Jhingran. “Landscape for Enterprise Data Integration”. Apigee blog, 2012. Disponível em: <https://apigee.com/about/blog/technology/etl-api-%E2%80%93-changed-landscape-enterprise-data-integration> - Acesso em: 10 de maio de 2017.

Panos Vassiliadis, Alkis Simitsis e Spiros Skiadopoulos. “Conceptual Modeling for ETL Processes”. National Technical University of Athens, 2007. Disponível em: <https://pdfs.semanticscholar.org/03af/306bcb882da089453fa57539f62aa7b5289e.pdf> - Acesso em: 11 de maio de 2017.

Stiphen J. Friedl. “An Illustration Guide to the Kaminsky DNS Vulnerability”. Stive Friedl, 2008. Disponível em: <http://unixwiz.net/techtips/iguide-kaminsky-dns-vuln.html> - Acesso em: 11 de maio de 2017.

Florent Carli. “Security Issues with DNS”. Sans Institute, 2003. Disponível em: <https://www.sans.org/reading-room/whitepapers/dns/security-issues-dns-1069> - Acesso em: 12 de maio de 2017.

Lior Rozen. “DNS and DNS attacks”. Radware blog, 2016. Disponível em: < https://blog.radware.com/security/2016/09/dns-and-dns-attacks/> - Acesso em: 22 de maio 2017.