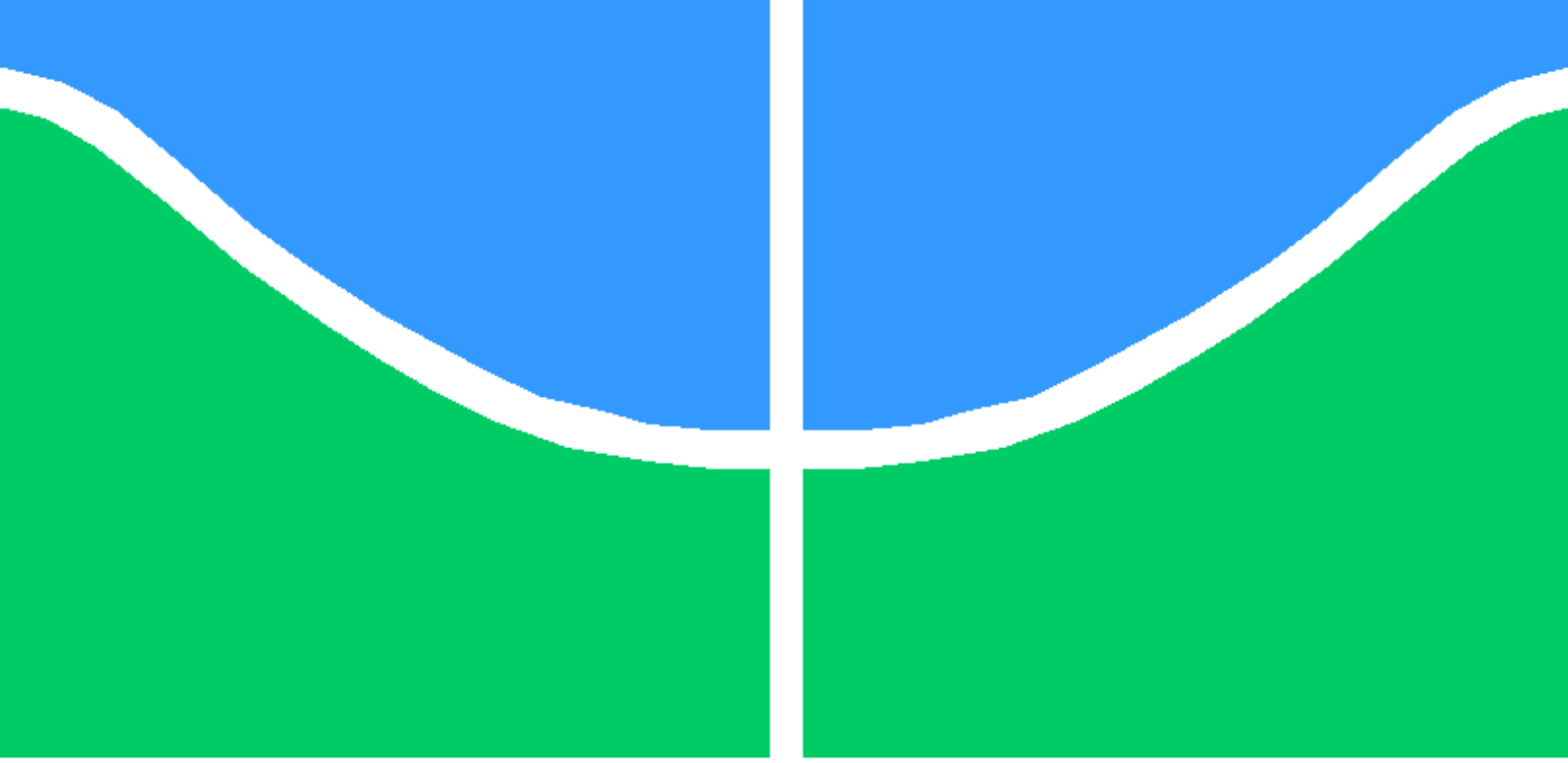

TRABALHO DE GRADUAÇÃO

ANÁLISE DE TRÁFEGO MALICIOSO DIRECIONADO A UMA HONEYNET COM INSPEÇÃO PROFUNDA DE PACOTES

Gabriel Arquelau Pimenta Rodrigues

Brasília, Novembro de 2017

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASILIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

ANÁLISE DE TRÁFEGO MALICIOSO DIRECIONADO A UMA HONEYNET COM INSPEÇÃO PROFUNDA DE PACOTES

Gabriel Arquelau Pimenta Rodrigues

Relatório submetido ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Engenheiro de Redes de Comunicação

Banca Examinadora

Prof. Flávio Elias Gomes de Deus, ENE/UnB _____
Orientador

Prof. Robson de Oliveira Albuquerque _____
Co-Orientador e Examinador Externo

Prof. Georges Daniel Amvame Nze, ENE/UnB _____
Examinador Interno

Porque nada há encoberto que não haja de ser manifesto; e nada se faz para ficar oculto, mas para ser descoberto.

Marcos 4:22

Agradecimentos

Agradeço a Deus pelas oportunidades e desafios a que me designou, sendo eles um intermédio para meu aprendizado; aos meus pais, Carlos Gomes e Lucileide Pimenta, pelos ensinamentos e apoio incondicional; aos meus professores orientadores, Flávio Elias e Robson Albuquerque, pelo conhecimento transmitido, suporte no desenvolvimento deste trabalho e esforços investidos no meu crescimento no mundo da segurança; ao professor Rafael Timóteo, que, mesmo não tendo sido meu orientador, foi também responsável pelo meu progresso academial; ao Laboratório para Análise Forense de Dispositivos Computacionais, que me propiciou um ambiente favorável à pesquisa e condução deste projeto. Agradeço ainda à belíssima letrista Paloma Medeiros, pelas revisões, e aos bons amigos que fiz durante o curso, pelo suporte.

Gabriel Arquelau Pimenta Rodrigues

RESUMO

Qualquer rede conectada à Internet é sujeita a ataques cibernéticos. Fortes medidas de segurança, ferramentas e investigadores forenses juntos contribuem na detecção e mitigação desses ataques, reduzindo os danos, possibilitando o reestabelecimento da rede a suas operações normais, e aumentando a segurança da rede. Este trabalho foca numa abordagem forense com Inspeção Profunda de Pacotes para detectar anomalias no tráfego de rede. Como ataques cibernéticos podem ocorrer em qualquer camada do modelo de rede TCP/IP, Inspeção Profunda de Pacotes é uma técnica efetiva para revelar conteúdo suspeito no cabeçalho ou na carga útil de qualquer pacote, exceto casos em que se faz uso de criptografia. Embora eficiente, essa técnica ainda encara desafios. As contribuições deste estudo se dão na associação de Inspeção Profunda de Pacotes com análise forense para avaliar diferentes ataques direcionados à Honeynet operando no laboratório LATITUDE da Universidade de Brasília. Nessa perspectiva, este trabalho pôde identificar e mapear o conteúdo e comportamento de ataques como a botnet Mirai e força-bruta, alvejando diferentes serviços. Os resultados obtidos demonstram o comportamento de ataques automatizados (como worms e bots) e não automatizados (força-bruta conduzida com diferentes ferramentas). Os dados coletados e analisados são, então, usados para gerar estatísticas de nomes de usuários e senhas, distribuição de IP e serviços e outros. Este trabalho também discute a importância da Cadeia de Custódia na condução de uma investigação e mostra a efetividade das técnicas mencionadas em avaliar diferentes ataques de redes.

ABSTRACT

Any network connected to the Internet is subject to cyber attacks. Strong security measures, forensic tools, and investigators contribute together to detect and mitigate those attacks, reducing the damages and enabling reestablishing the network to its normal operation, thus increasing the cybersecurity of the networked environment. This work addresses the use of a forensic approach with Deep Packet Inspection to detect anomalies in the network traffic. As cyber attacks may occur on any layer of the TCP/IP networking model, Deep Packet Inspection is an effective technique to reveal suspicious content in the headers or the payloads in any packet processing layer, excepting situations where the payload is encrypted. Although being efficient, this technique still faces big challenges. The contributions of this study rely on the association of Deep Packet Inspection with forensics analysis to evaluate different attacks towards a Honeynet operating in the LATITUDE laboratory at the University of Brasilia. In this perspective, this work could identify and map the content and behavior of attacks such as the Mirai botnet and brute-force attacks targeting various different network services. Obtained results demonstrate the behavior of automated attacks (such as worms and bots) and non-automated attacks (brute-force conducted with different tools). The data collected and analyzed is then used to generate statistics of used usernames and passwords, IP and services distribution, among other elements. This work also discusses the importance of network forensics and Chain of Custody procedures to conduct investigations and shows the effectiveness of the mentioned techniques in evaluating different attacks in networks.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

Acrônimos

API	*Application Programming Interface*
C&C	Comando e Controle
BAF	*Bandwidth Amplification Factor*
DDoS	*Distributed Denial of Service*
DNS	*Domain Name System*
DNSSec	Extensão de Segurança do DNS
DPI	*Deep Packet Inspection*
EDNS0	Mecanismo de Extensão de DNS versão 0
FPGA	*Field Programmable Gate Array*
FTP	*File Transfer Protocol*
GPU	*Graphics Processing Unit*
HTTP	*Hypertext Transfer Protocol*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection Systems*
IOC	*Indicator Of Compromise*
IoT	*Internet of Things*
IPS	*Intrusion Prevention Systems*
IRC	*Internet Relay Chat*
ML	*Machine Learning*
MPI	*Medium Packet Inspection*
NetBIOS	*Network Basic Input/Output System*
NTP	*Network Time Protocol*
PDU	*Protocol Data Unit*
QoS	*Quality of Service*
RPC	*Remote Procedure Call*
RTT	*Round Trip Time*
SMB	*Server Message Block*
SPI	*Shallow Packet Inspection*
SQL	*Structured Query Language*
SSH	*Secure Shell*
TOR	*The Onion Router*
TTL	*Time To Live*

Capítulo 1

Introdução

Nos anos recentes, o tráfego da Internet apresentou um grande crescimento e, de acordo com Cisco (2017), é esperado que haja 278,108 PetaBytes de tráfego IP sendo transmitidos globalmente por mês em 2021. Como consequência, a ocorrência e diversidade de cyber-ataques, como a recente infecção global do ransomware WannaCry (COUGHLIN, 2017), também aumenta, o que promove a necessidade de estudar o comportamento dos atacantes, para, assim, alcançar uma detecção mais acurada das anomalias de rede.

Ataques de rede podem ocorrer em quaisquer ou em múltiplas camadas da pilha TCP/IP e, portanto, uma análise detalhada do conteúdo dos pacotes é necessária. Essa análise é conhecida como Inspeção Profunda de Pacotes (DPI), e consiste na inspeção de cada campo de cada camada de todos pacotes trafegando na rede, bem como sua carga útil. Essa abordagem é usada para identificar padrões que podem evidenciar tráfego malicioso e o modus operandi do suspeito.

A inspeção de pacotes pode ser feita em um tráfego conhecidamente malicioso, como o destinado e originado em uma Honeynet — uma rede intencionalmente vulnerável —, a fim de se estudar o comportamento dos atacantes, bem como suas intenções, motivações e vulnerabilidades exploradas, para, assim, obter o conhecimento necessário para impedir a intrusão em redes que devem ser protegidas. Um exemplo de estudo desse tipo foi feito por Pimenta Rodrigues et al. (2017), que gerou estatísticas e evidenciou comportamentos anômalos na Honeynet do LATITUDE-UnB, bem como métodos de detecção e mitigação dos ataques apresentados.

Como dito anteriormente, entretanto, o volume de dado transmitido pela Internet é enorme, e sua maior parte corresponde a usuários legítimos, o que dificulta a detecção de uma pequena porcentagem de tráfego suspeito. Esse mega-dado (em inglês, *Big Data*) deve ser analisado como um todo para que se obtenha conclusões válidas, uma vez que analisar alguns pacotes fora de seus respectivos contextos pode levar a uma detecção falso positiva ou falso negativa.

1.1 Definição do Problema

A expansão da Internet e advento de novas tecnologias, como Internet das coisas (IoT), permitem que companhias forneçam seus serviços online, que traz praticidade aos usuários, mas também expande a superfície de ataque. Ataques cibernéticos causam grandes prejuízos ao seu alvo, como mostrado por Anderson et al. (2013), que estima um custo de U$ 370 milhões associado ao sucesso de malwares em fraudes globais em Internet Banking em 2010, e U$ 320 milhões associado a Phishing, com o mesmo tipo de fraude, no mundo, em 2007. A divulgação dessas vulnerabilidades faz, ainda, com que as companhias atacadas percam espaço de mercado, o que resulta num prejuízo indireto somado.

Além de prejuízo financeiro, ataques de rede também ameaçam, por exemplo, a vida de dependentes de aparelhos hospitalares, como no caso do WannaCry, que deixou diversos hospitais inoperantes no mundo; ou da botnet Mirai (SYMANTEC, 2017), que infectou centenas de milhares de dispositivos IoT, alguns dos quais sendo relacionados à saúde humana.

Ainda, a ocorrência de vazamento de informações pessoais em larga escala, decorrente de ataques cibernéticos, atinge milhões de indivíduos por ano, como mostra a Figura 1.1, o que afeta suas privacidades e a segurança.

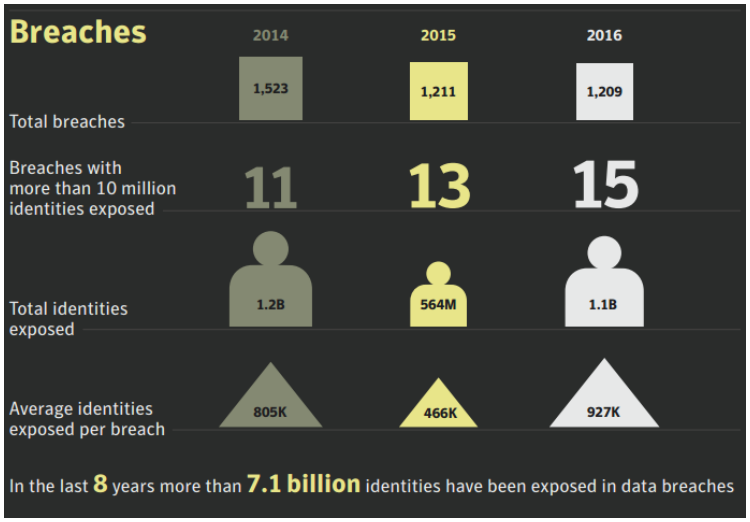

Figura 1.1: Quantidade de pessoas cujos dados foram indevidamente divulgados nos últimos anos. (SYMANTEC, 2017).

Atacantes estão constantemente evoluindo suas técnicas e, em resposta, especialistas em segurança de redes devem fazer o mesmo, a fim de melhor compreender as técnicas e vulnerabilidades exploradas, mitigando ataques e auxiliando na detecção de futuras atividades maliciosas.

Estudar o comportamento e o modus operandi de atacantes em uma Honeynet retorna informações que são úteis na detecção de atividades semelhantes em redes cujo tráfego, em sua maior parte, corresponde a usuários legítimos, reduzindo os prejuízos financeiros e outras consequências negativas advindas de ataques de redes.

1.2 Objetivos

O Objetivo Geral deste projeto é fornecer informações acerca do tráfego malicioso direcionado à Honeynet do LATITUDE, instalada por Oliveira Júnior, Sousa Júnior e Tenório (2015), possibilitando uma detecção mais acurada e precisa de tráfego semelhante em outras redes. Para alcançar este objetivo, os seguintes Objetivos Específicos são propostos:

- Estabelecimento da arquitetura para análise dos dados provenientes da Honeynet;
- Uso de DPI e softwares *open source* para visualização dos dados;
- Evidenciar características anômalas no tráfego e o modus operandi dos usuários maliciosos;
- Comparar o tráfego da Honeynet com dados obtidos de sensores externos.

1.3 Estrutura do Trabalho

O restante desse trabalho é organizado da seguinte forma: o Capítulo 2 consiste em uma discussão acerca dos conceitos básicos que compõem este trabalho.

O Capítulo 3 apresenta a arquitetura, softwares e metodologia dos estudos conduzidos.

O Capítulo 4 mostra as características maliciosas do tráfego da Honeynet estudada, e detalhes das vulnerabilidades e dos ataques detectados, bem como uma comparação desse tráfego com os dados apresentados pelo Norse.

O Capítulo 5 contém a discussão final do estudo e conclui o trabalho, apresentando possíveis trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Este Capítulo aborda os conceitos fundamentais que baseiam esse trabalho, discutindo seus desafios e pesquisas relacionadas.

2.1 O Modelo TCP/IP e Protocolos de Aplicação

Esta seção apresenta conceitos básicos acerca da arquitetura em camadas em que a Internet é baseada, que é importante para compreensão das estatísticas do tráfego malicioso e anomalias detectadas. A tabela 2.1 apresenta uma breve descrição de cada camada, junto com a Unidade de Dados de Protocolo (PDU) associada. As camadas também são comumente referenciadas com números, sendo a camada física a de número 1 e subindo até a camada de aplicação, que é a de número 5.

Tabela 2.1: Arquitetura em camadas do modelo TCP/IP (KUROSE; ROSS, 2013).

Camada	PDU	Descrição
Aplicação	Mensagem	Camada da ponta do usuário, contendo a carga útil do pacote
Transporte	Segmento	Carrega mensagens da camada de aplicação entre os lados do cliente e servidor de uma aplicação
Rede	Datagrama	Roteamento entre diferentes redes
Enlace	Quadro	Estabelece e coordena comunicação na camada física
Física	Bit	É o próprio meio de transmissão dos pacotes (e.g. cabos)

Para compreensão das anomalias presentes em cada protocolo da camada de aplicação estudado, é importante conhecer seu funcionamento legítimo. Assim, as seções 2.1.1 a 2.1.7 apresentam resumidamente esses protocolos.

2.1.1 Porta 23: Telnet

Telnet é um protocolo usado para estabelecer uma comunicação remota entre máquinas, dando ao usuário o acesso ao terminal de uma outra máquina, mesmo que elas não estejam fisicamente conectadas.

Embora essa conexão só permita a transmissão de dados em texto, ela pode ser usada para dar comandos à máquina remota e, para impedir usuários indesejados e mal intencionados, essa conexão normalmente exige uma autenticação. Entretanto, essa conexão não é criptografada, isto é, tanto o usuário e a senha quanto os dados posteriores são transmitidos em texto claro, e podem ser interceptados e lidos por terceiros.

2.1.2 Porta 22: SSH

Uma opção mais segura, e também mais comum, para estabelecer comunicações remotas é por meio do Secure Shel (SSH), que fornece mecanismos de criptografia simétrica e assimétrica para autenticar e proteger os dados transmitidos.

Para impedir a conexão de usuários indevidos, o SSH permite alguns modos diferentes de autenticação. Dentre eles, o emprego de usuário e senha, como no Telnet, mas transmitindo as credenciais em texto cifrado, o que impede a leitura por terceiros. Outro modo de autenticação, sendo este muito mais seguro, é por meio de chaves públicas, em que o cliente, tendo um par de chaves pública e privada geradas por um algoritmo apropriado, envia sua chave pública ao servidor que, se encontrar essa chave na lista de chaves permitidas, garante o acesso ao cliente.

2.1.3 Portas 139 e 445: NetBIOS e SMB

Visando a comunicação entre aplicações e dispositivos numa rede local, o Network Basic Input Output System (NetBIOS) permite, dentre outros, o compartilhamento de arquivos usando identificadores para cada dispositivo.

Usando a *Application Programming Interface* (API) do NetBIOS, o Server Message Block Protocol (SMB) fornece compartilhamento de arquivos entre impressoras, portas seriais e outros meios de comunicação e dispositivos. Ao usar a API do NetBIOS, esse serviço é vinculado à porta 139 e, quando roda diretamente sobre a pilha TCP/IP, ele é associado à porta 445.

O protocolo SMB, no entanto, é inseguro e o recente ataque do ransomware WannaCry explorou uma vulnerabilidade desse protocolo conhecida como EternalBlue para infectar centenas de milhares de dispositivos no mundo todo (BERRY; HOMAN; EITZMAN, 2017). Essa vulnerabilidade foi corrigida na atualização de segurança MS17-010 da Microsoft.

2.1.4 Porta 80: HTTP

Usado em conexões Web, o Hypertext Transfer Protocol (HTTP) é baseado em requisições, feitas pelo cliente, e respostas, pelo servidor. Usando verbos HTTP, o cliente pode operar os dados, identificáveis pelo Uniform Resource Identifier (URI), disponíveis no servidor. Esse é um protocolo sem estado, o que significa que cada requisição é independente das outras. Alguns dos verbos HTTP, e suas descrições, são:

- GET: Usado para solicitar informações de um servidor, dado um URI. Esse verbo deve retornar somente dados;
- HEAD: Semelhante ao GET, mas retorna os dados somente da linha de status e `header`, sem `body`;
- POST: Transfere dados para o servidor, como preenchimento de formulário e upload de arquivos;
- DELETE: Remove as representações do recurso identificado pelo URI fornecido.

2.1.5 Porta 21: FTP

O File Transfer Protocol (FTP) é um protocolo que, por meio de comandos e respostas, permite o armazenamento, o compartilhamento e a transferência de arquivos entre máquinas conectadas à Internet, mesmo que sejam de diferentes Sistemas Operacionais.

Para impedir que qualquer usuário se conecte a um servidor FTP, e faça operações indesejadas nos arquivos armazenados, é prática comum implementar um sistema de autenticação, solicitando do cliente um usuário e uma senha válidos, que são transmitidos na rede em texto claro. No entanto, alguns servidores FTP públicos permitem que seus clientes se conectem a ele sem nenhuma autenticação, modalidade conhecida como `Anonymous FTP`. Nesse caso, basta fornecer a string `anonymous` como usuário e uma senha qualquer ou, caso o servidor exija uma forma mínima de identificação, um endereço de e-mail válido.

2.1.6 Porta 53: DNS

Máquinas usam endereço IP para identificar e comunicar com outras máquinas. Entretanto, memorizar essas cadeias de números de cada servidor é inviável para humanos. Para tornar possível a mútua tradução entre nomes, mais facilmente memorizáveis por humanos, e endereços IP, utilizados por máquinas, utiliza-se o serviço Domain Name System (DNS).

O DNS é um serviço hierarquizado, tendo um servidor raiz que conhece os servidores inferiores a ele e os domínios a que respondem. Essencialmente, ao precisar converter de um nome para um endereço IP, o cliente faz uma requisição a um servidor DNS, que, caso conheça a tradução apropriada para esse nome, fornecerá ao cliente as informações solicitadas. Caso esse servidor não possua essas informações, ele fará a solicitação ao servidor raiz. Se esse, por sua vez, não souber,

indicará um servidor que responde por esse domínio, e assim sucessivamente, até se obter a resposta adequada, que é passada ao cliente que originou o processo de tradução. Essa conversão de nome para endereço IP é chamada de resolução de nome, enquanto que o processo contrário é chamado de resolução reversa.

Para tornar a resolução de nomes mais eficiente, o DNS usa um sistema de cache, ou seja, o dispositivo armazena temporariamente os nomes e suas traduções recentemente solicitadas, e somente fará novas requisições se ele não já conhecer a tradução.

2.1.7 Porta 123: NTP

Para gerenciar, debugar e analisar correlação entre pacotes de uma rede, é importante que o relógio das máquinas envolvidas esteja em sincronia, possibilitando associar diferentes eventos no domínio do tempo. Por padrão, as horas dos dispositivos não são acuradas, e tendem a perder mais exatidão com o tempo. Para manter dispositivos sincronizados, servidores Network Time Protocol (NTP) consultam um relógio de referência, de alta acurácia, para fornecer a hora a seus clientes.

A RFC 5905, que especifica o protocolo, descreve alguns *timestamps* importantes na comunicação NTP:

- Reference Timestamp: Hora em que o relógio do sistema foi configurado pela útlima vez;
- Origin Timestamp: Hora em que a requisição partiu do cliente;
- Receive Timestamp: Hora em que a requisição chegou no servidor;
- Transmit Timestamp: Hora em que a resposta partiu do servidor;
- Destination Timestamp: Hora em que a resposta chegou no cliente.

Esses valores são usados, por exemplo, para determinar o Round Trip Time (RTT), e configurar o relógio com maior exatidão.

2.2 Honeynet

Honeynet é uma rede propositalmente vulnerável, estabelecida com o fim de se receber ataques, por motivos variados. Cada dispositivo que pode ser comprometido dentro da Honeynet é chamado de Honeypot, e pode ser classificado de acordo com o seu uso e grau de interação com o usuário. Essas redes não costumam ser divulgadas, fazendo com que o tráfego nelas seja unicamente malicioso, já que atacantes varrem a Internet em busca de alvos desprotegidos, enquanto usuários legítimos não acessam essa rede por não a conhecerem.

De acordo com Spitzner (2003), Honeynets começaram em 1999, com um pequeno grupo de pesquisadores que logo percebeu que poucas pessoas não poderiam analisar tantos malwares e vulnerabilidades. Dessarte, se expandiram e se oficializaram como O Projeto Honeynet[1].

[1] `Siteoficial:https://www.honeynet.org/`

O uso dessas redes, no entanto, traz riscos às organizações que as implantam, sendo necessário considerar alguns aspectos antes de sua instalação.

2.2.1 Classificação

Verma (2003) classifica as Honeypots, quanto a seu uso, em de produção e de pesquisa; e, quanto ao seu grau de interação, em baixo, médio e alto. Ainda de acordo com o autor, Honeypots de produção são comumente usadas em organizações para reduzir os riscos cibernéticos, fazendo com que atacantes gastem recursos e tempo neste ambiente controlado, ao invés de no verdadeiro sistema de produção, que pode conter dados sigilosos e sensíveis.

Já os Honeypots de pesquisa têm a intenção de registrar e fornecer os dados acerca dos ataques recebidos: sua origem, vulnerabilidades exploradas, motivações e finalidades, auxiliando pesquisadores a melhor entender o comportamento de atacantes e fortalecer seus sistemas de segurança. Naturalmente, esse tipo de uso de Honeypot oferece um maior risco, uma vez que podem ser comprometidos e usados para atacar redes externas à Honeynet, enquanto os Honeypot de produção são usados justamente para desviar os esforços dos atacantes. Em ambos os usos, no entanto, devem ser implantadas medidas de controle para garantir a recuperação das máquinas em caso de serem comprometidas.

Honeypots de baixa interação fornecem sistemas simulados e, portanto, uma baixa liberdade ao usuário. Esses honeypots são passivos, logo, não podem ser usados para ataques a redes externas, diminuindo o risco de sua implementação (VERMA, 2003). Entretanto, por serem sistemas simulados, podem não fornecer informações acuradas sobre o tráfego malicioso. Um exemplo de Honeypots de baixa interação é Honeyd[2]. Honeypots de média interatividade, de acordo com o autor, fornecem mais serviços, mas ainda sem se tratar de sistemas reais.

Tabela 2.2: Características das diferentes classes de Honeypots.

	Uso		Grau de Interação		
	Produção	Pesquisa	Baixo	Médio	Alto
Risco	Baixo	Alto	Baixo	Médio	Alto
Finalidade	Desviar ataques	Estudar ataques	-	-	-
Emulação	-	-	Alta	Média	Baixa
Informação	-	-	Pouca	Moderada	Muita
Custo	-	-	Baixo	Médio	Alto

Honeypots de alta interação, por outro lado, consistem em Sistemas Operacionais reais. Isso resulta numa maior atratividade para os atacantes, maior dificuldade de detecção e informações mais completas acerca das ações dos usuários maliciosos. Contudo, quanto maior o grau de interatividade, maior será o a complexidade e o custo de sua instalação, já que os sistemas reais

[2]Site oficial: `http://www.honeyd.org/`

exigem recursos computacionais, e maior o risco de comprometimento e ataques a redes externas. Ademais, a Tabela 2.2 sumariza as particularidades de cada classificação de Honeypots.

2.2.2 Riscos e requisitos

Como os Honeypots continuamente recebem ataques, eles podem, eventualmente, ser comprometidos e usados em ataques a terceiros ou ter todos os rastros dos ataques apagados. Por esse motivo, as Honeynets devem incluir softwares que capturem e controlem o tráfego de maneira a não despertar a atenção dos atacantes.

Um desses softwares, de acordo com Provos e Holz (2007), é o Honeywall, que, segundo o autor, executa as tarefas a seguir de maneira transparente, isto é, sem endereço IP nas interfaces que conectam os Honeypots e a Internet, para dificultar a detecção.

- Captura de dados: Toda atividade dentro da Honeynet deve ser registrada sem o conhecimento do atacante;
- Controle de dados: Para impedir ataques a redes externas, tráfego de saída é limitado. Ainda, o tráfego de um Honeypot que foi infectado deve ser bloqueado;
- Análise de dados: Embora não seja indispensável na implementação, essa funcionalidade auxilia na compreensão das características do tráfego malicioso.

Ainda consoante o autor, entretanto, somente os dados de rede podem não ser suficientes para fornecer o conhecimento necessário a respeito dos ataques. Em função disso, o software Sebek viabiliza a captura de dados até mesmo em conexões criptografadas, como o pressionar de teclas em sessões SSH. Neste projeto, no entanto, somente o tráfego na rede é estudado.

Além das referidas considerações técnicas, Mokube e Adams (2007) citam os aspectos legais do uso de uma Honeynet para estudar o tráfego. Como exemplo, a alegação, pela defesa dos atacantes, de induzimento da prática de crime, que, de acordo com os autores, é uma argumentação inválida, já que os usuários não foram impelidos a transmitir os pacotes à Honeynet.

Outro aspecto legal refere-se à privacidade dos usuários, que, mesmo sendo invasores na rede, possuem esse direito. Por esse motivo, todos os dados referentes ao tráfego malicioso apresentados neste trabalho têm sua origem anonimizada.

2.2.3 Detecção de Honeynets

Independente do uso da Honeynet, é importante que ela seja de difícil detecção pelos atacantes. Se o usuário mal intencionado perceber que se trata de um ambiente controlado, ele provavelmente interromperá a execução do ataque, eliminando as evidências, e poderá mudar o foco para a rede protegida, no caso da Honeynet de produção, ou retirar as evidências referentes ao seu ato ilegítimo, no caso da Honeynet de pesquisa. Como diferem na implementação, o modo para detecção de Honeynets depende do seu grau de interação.

Para detectar Honeynets de baixa interatividade, Mukkamala et al. (2007) descrevem o uso de análise temporal de requisições Internet Control Message Protocol (ICMP). Essencialmente, sistemas simulados demoram mais para responder essas requisições, porque os pacotes precisam, primeiramente, chegar na máquina real para, então, serem direcionadas ao sistema simulado. Além disso, por serem mais limitados em relação aos sistemas reais, os simulados não implementam algumas funcionalidades do serviço emulado, que pode ser percebido pelo usuário. Provos e Holz (2007) citam outras características detectáveis de Honeypots de baixa interatividade, como o Honeyd que, em versões anteriores, tinha a mesma semente de relógio para diversos sistemas simulados, o que não é esperado em sistemas reais.

Por se tratar de sistemas reais, a detecção de Honeynets de alta interatividade é consideravelmente mais difícil, mas ainda existem métodos. Phrack, uma revista escrita por e para hackers, publicou diversos artigos em 2003 descrevendo técnicas para detecção de Honeynets. Dentre elas, a detecção pelo Sebek (COREY, 2003), por meio de traços deixados na memória. Provos e Holz (2007) relatam outros mecanismos para detectar tais Honeynets, como a verificação da existência do limite de tráfego de saída, discutido na seção 2.2.2.

2.3 Inspeção de Pacotes

Inspeção de pacotes é uma técnica assaz utilizada com diferentes finalidades, como em modelagem de tráfego, Qualidade de Serviço (QoS) e em segurança de redes. Esta seção aborda os conceitos e diferentes profundidades da inspeção de pacotes, focando em sua aplicação na cibersegurança.

2.3.1 Profundidades de Inspeção

A inspeção de pacotes pode ser classificada de acordo com as camadas que analisa de cada pacote, variando em Inspeção Rasa de Pacotes (SPI), Inspeção Mediana de Pacotes (MPI) e Inspeção Profunda de Pacotes (DPI), cada profundidade tem características que as tornam aplicáveis em diferentes cenários. A figura 2.1 mostra as camadas alcançadas por cada nível de inspeção de pacote, sendo que, até a camada de transporte, a análise é feita sob os valores dos campos dos protocolos, e na camada de aplicação, analisa-se a carga útil transmitida.

Figura 2.1: Profundidades de inspeção de pacotes e as camadas do modelo OSI e TCP/IP que abrangem (PARSONS, 2008).

De acordo com Parsons (2008), SPI é usado, por exemplo, em firewalls simples, que, por ser limitado até a camada 3, só comparam endereços de origem e destino a uma *blacklist*, bloqueando os pacotes cujos endereços estão presentes na lista.

Ainda em conformidade com o autor, MPI pode determinar, por exemplo, os formatos dos dados sendo transmitidos e, portanto, permite o bloqueio de tráfego com base no seu gênero (vídeo, áudio, texto, etc.), fornecendo QoS. Esse nível de inspeção, no entanto, não permite a leitura da carga útil do pacote.

A Inspeção Profunda de Pacotes é a única que permite a leitura completa da carga útil do pacote, por ser a única que abrange a camada de aplicação completamente, o que possibilita uma investigação mais detalhada do tráfego. No entanto, DPI gera uma maior quantidade de tráfego, exigindo maiores recursos computacionais para seu armazenamento e análise.

2.3.2 Inspeção de Pacotes em Segurança de Redes

Atacantes podem abusar de vulnerabilidades presentes em qualquer camada, usualmente deixando rastros de suas ações, que podem ser evidenciados com inspeção de pacotes. Naturalmente, quanto maior a profundidade da inspeção de pacotes, mais camadas serão analisadas e, consequentemente, maiores as chances de se detectar o ataque e mais informações serão adquiridas.

Mueller (2011) cita as capacidades primárias de DPI, que devem ser consideradas ao implementar a técnica em cibersegurança. Elas são:

- Reconhecimento: DPI pode ser usado para detectar conhecidos padrões maliciosos no tráfego. Essa capabilidade engatilha as próximas duas;
- Manipulação: Agir em resposta ao evento detectado, incluindo, por exemplo, bloqueio dos pacotes;

- Notificação: Alertar administradores e responsáveis pela rede, fornecendo informações úteis para responder ao incidente e evitar futuro tráfego similar.

A Tabela 2.3 compendia rastros maliciosos que podem ser encontrados em cada camada do tráfego. Importante notar que a camada física não é citada na tabela, posto que técnicas de inspeção de pacotes não são eficazes na detecção de padrões maliciosos nessa camada.

Tabela 2.3: Possíveis evidências de atos suspeitos encontradas em cada camada do modelo TCP/IP.

Camada	Anomalia e evidência
Aplicação	Cada protocolo dessa camada pode apresentar diferentes anomalias. Conteúdo malicioso nessa camada e sua detecção são apresentados na seção 4.4.
Transporte	Atacantes podem transmitir uma grande quantidade de pacotes TCP SYN, abrindo diversas conexões na vítima e consumindo seus recursos (US-CERT, 2014). Esse ataque pode ser detectado pelo envio de grande quantidade desse pacote em um curto período.
Rede	Endereços IP podem ser forjados para afetar suas vítimas. Templeton e Levitt (2003) citam o uso da variação do campo Time To Live (TTL) na detecção dessa anomalia.
Enlace	Endereços MAC podem ser forjados para praticar ataques. Essa prática pode ser detectada, por exemplo, por meio da variação do potência do sinal recebido (YU et al., 2016).

2.3.3 Limitações e Desafios da Inspeção

Não obstante, o uso de DPI na análise de tráfego, em especial no caso da cibersegurança, encara desafios que limitam a investigação. Um deles é a criptografia, que impede a leitura do conteúdo dos pacotes capturados. Em razão disso, a análise de tráfego de protocolos criptografados, como SSH e HTTPS, limita-se aos meta-dados, como origem, data e hora da transmissão e algoritmo de criptografia utilizado. Num ambiente de Honeypots, no entanto, o uso do software Sebek permite a captura desses dados em texto claro.

Outro desafio refere-se à capacidade de processamento de pacotes do dispositivo a que a inspeção de pacotes foi atribuída. Como visto na seção 1.1, o tráfego IP cresceu e espera-se que continue crescendo muito, resultando em uma grande quantidade de pacotes fluindo nas redes de Internet, incluindo a inspecionada. Isso exige que, para que não tenha um atraso acumulado, o dispositivo processe os pacotes a uma velocidade maior que a velocidade com que pacotes aparecem na rede. Em casos nos quais essa condição não é satisfeita, pode-se optar, por exemplo, por uma arquitetura paralela de inspeção de pacotes, aumentando o desempenho da análise. Essa prática, entretanto, pode não ser escalável quando desacompanhada de outras técnicas.

Outrossim, a expansão da superfície de ataque, também mencionada na seção 1.1, proporciona o aparecimento de novos e mais complexos ataques, exigindo dos investigadores uma inspeção mais detalhada para detectar, mitigar e responder ao ataque. Dispositivos de inspeção de pacotes

que seguem regras estáticas no reconhecimento de tráfego malicioso não vão, automaticamente, adaptar-se ao desenvolvimento dos ataques, mas exigirão que as pessoas responsáveis atualizem suas regras para incluir as referentes aos novos ataques. A fim de superar essa limitação, técnicas de Aprendizagem de Máquina (ML) possibilitam que dispositivos, que previamente conhecem características de tráfego benigno e maligno, se adéquem a novas formas de tráfego malicioso.

A última limitação aqui citada concerne à inspeção na camada 3 do tráfego, que pode ser restringida pelo uso de anonimadores de tráfego, como The Onion Router (TOR). O uso de tais tecnologias impede que a inspeção determine o endereço IP de origem de um dado pacote, consequentemente impedindo, também, seu georreferenciamento.

2.3.4 Pesquisas Recentes

Por ser uma técnica eficiente de averiguação de tráfego, inspeção de pacotes é muito utilizada por especialistas em segurança, que buscam, também, a redução das limitações descritas na seção 2.3.3. Algumas dessas pesquisas são apresentadas nessa seção.

Para tornar viável o uso de DPI em tráfego criptografado, e ao mesmo tempo garantindo a privacidade dos usuários envolvidos, Yuan e et al (2016) propuseram uma solução que permite a detecção de pacotes anômalos cifrados mesmo sem revelar seus conteúdos. A técnica consiste, fundamentalmente, numa inspeção cifrada de *tokens* aleatórios do tráfego e, por meio de algoritmos apropriados, caracteriza-se o pacote em legítimo ou não. Os autores mostraram que esse sistema é capaz de analisar os pacotes com baixa latência, uma característica importante na inspeção de pacotes, como descrito na seção 2.3.3.

Xu et al. (2016) descrevem diferentes técnicas usadas para aumentar o desempenho da inspeção de pacotes por hardware e por software. Dentre elas, aceleração de hardware com Arranjo de Portas Programáveis em Campo (FPGA) e uso de Unidade de Processamento Gráfico (GPU), comparando as vantagens e desvantagens de cada técnica e características como desempenho, custo e escalabilidade.

Para se alcançar uma detecção de novos ataques, uma técnica eficiente é ML, que automatiza e dinamiza as regras que definem o tráfego anômalo. Existem diferentes técnicas com diferentes aplicabilidades dentro do domínio de ML. Buczak e Guven (2016) explicam e comparam essas diferentes metodologias, junto com o uso e desafios de ML em cibersegurança.

Embora não seja possível determinar a origem de tráfego TOR, devido à arquitetura da tecnologia, Saputra e et al (2016) apresentaram um sistema que utiliza inspeção de pacotes que identifica características de tráfego TOR num pacote, possibilitando o bloqueio desses pacotes para evitar potenciais ataques.

2.4 Mega-dados

Como citado por Kitchin e McArdle (2016), as características principais de Mega-Dados, também chamadas de 3Vs, são:

- Volume: Consistem numa enorme quantidade de dados;
- Velocidade: Novos dados criados a uma alta taxa e em tempo real;
- Variedade: Diversos tipos de informação são armazenados em Mega-Dados.

Com a discussão feita até aqui, é possível notar que dados provenientes de redes, e que são objeto da investigação deste trabalho, são considerados Mega-Dados, já que são gerados a uma alta taxa que tende a crescer mais (CISCO, 2017), originando consequentemente um grande volume de dados. Também, a abundância de protocolos da camada de aplicação, cada um com diferentes campos e requerendo inspeções específicas, resulta numa variedade de informação contida nos dados coletados. Ainda mais se, além dos dados de tráfego de redes, a análise abranger, por exemplo, arquivos de logs de firewall, Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS).

Essas características dificultam a mineração e análise dos dados da rede, já que os eventos de interesse vão corresponder a porcentagens menores do conjunto de dado. Além disso, Talabis et al. (2014) citam a dificuldade de investigação de um evento do qual pouco se conhece. Nesse caso, os investigadores devem usar as informações que têm (data do evento, servidor atacado, etc.) para analisar o volumoso conjunto de dados e adquirir mais evidências que os ajudem a entender o ocorrido. Em oposição a esse caso, conhecer previamente que se trata de um ataque de injeção de código Structured Query Language (SQL), por exemplo, fornece mais informações que podem ser usadas para filtrar o grande conjunto de dados em um conjunto menor e fazer com que investigadores procurem por conteúdo mais específico nos logs, como códigos SQL. Isso facilita a identificação da origem do evento e consequente obtenção de mais informações sobre ele, como endereço IP do atacante, motivações, etc.

O conhecimento prévio de um evento torna o processo de investigação mais rápido e fácil, mas, para poder usá-lo para filtrar e reduzir o conjunto de dados a ser investigado, é importante que se tenha ferramentas que possibilitem a rápida pesquisa e visualização de Mega-Dados. Uma ferramenta comumente usada e especializada em Mega-Dados é o Hadoop (WHITE, 2012), aliada ao modelo de programação MapReduce (DEAN; GHEMAWAT, 2008), mas neste projeto usa-se o conjunto de ferramentas Elastic Stack, pois fornece rápida indexação e busca, além de eficazes modos de visualização de dados. O Elastic Stack é apresentado em mais detalhes na seção 3.2.1.

2.5 Forense em Redes

Focando em segurança cibernética, o principal objetivo da condução de forense em redes é de adquirir evidência para obter informação e reconstruir um incidente, crime ou ataque. Nesse caso,

as evidências podem ser obtidas do fluxo de pacotes na rede, arquivos de log de firewalls, IDS, IPS, etc.

Como apresentado na seção 2.4, os dados referentes ao fluxo de pacotes na rede, que pode ser investigado com DPI, são considerados Mega-Dados, o que resulta em desafios para a prática forense.

Khan e et al (2016) também citam a integridade de dados como desafio da forense em redes, já que evidências digitais podem ser facilmente manipuladas, acidentalmente ou não, corrompendo as informações e comprometendo sua validade. Isso pode ser evitado ao se criar cópias dessa evidência, e as analisar como se fosse a evidência original, sem correr riscos de prejudicar sua integridade.

Um aspecto positivo da investigação digital é que arquivos corrompidos podem ser facilmente identificados ao se comparar seu hash com o do arquivo original. Além disso, algumas mídias permitem a recuperação de arquivos deletados usando ferramentas apropriadas.

Uma outra ação importante para garantir a credibilidade da investigação é documentar todas pessoas que tiverem acesso às evidências e quais ações executaram. Esse documento é a Cadeia de Custódia, onde cada pessoa é uma ligação da cadeia e fornece uma trilha cronológica da evidência, desde sua descoberta até sua apresentação no tribunal (PRAYUDI; SN, 2015). Flores Armas e Jhumka (2017) citam quatro princípios com os quais a Cadeia de Custódia deve estar em acordo:

1. Nenhuma ação deve ser feita por qualquer um de forma a alterar a evidência;
2. Caso o acesso à evidência original seja necessária, deve-se explicar a relevância e implicações dos atos;
3. Deve-se registrar todas investigações feitas, de modo que um terceiro possa conduzir a investigação da mesma forma e alcançar os mesmos resultados;
4. A pessoa no comando da investigação deve garantir o cumprimento desses princípios.

Capítulo 3

Métodos Propostos e Ferramentas Utilizadas

Este capítulo apresenta as fontes dos dados analisados, isto é, Honeynet do LATITUDE e os sensores do Norse, e a arquitetura e ferramentas empregadas para análise dos dados coletados.

3.1 Fontes de Dados

Os dados analisados neste estudo são originados da Honeynet instalada no LATITUDE, no Departamento de Engenharia Elétrica da UnB, e dos sensores do Norse, que publicam os ataques detectados em tempo real. Esta seção descreve essas fontes de dados.

3.1.1 Honeynet do LATITUDE

De acordo com as classificações de Honeynet apresentadas na seção 2.2.1, a Honeynet do LATITUDE foi montada para fins de pesquisa e é de alta interatividade. A rede possui cinco Honeypots ativos, conforme mostra a Figura 3.1, que são sistemas desatualizados e inseguros, dentre os quais quatro estão rodando os serviços DNS, HTTP, NTP, FTP e um, de Sistema Operacional Windows XP SP3, não roda nenhum serviço.

Qualquer indivíduo na Internet pode conectar-se com qualquer um dos Honeypots, que não foram divulgados, o que sugere que o tráfego recebido é completamente malicioso. Para dificultar a detecção do ambiente, o Honeywall não tem endereço IP nas interfaces entre a Honeynet e a Internet, e também se conecta ao host de endereço 172.30.10.20, na rede de gerência, pelo qual os dados do tráfego são coletados. A rede de gerência e a dos Honeypots não se comunicam, para evitar que os atacantes possam adulterar ou deletar os dados coletados.

Para diminuir os riscos de uso da Honeynet em ataques a terceiros, o firewall limita o número de pacotes em comunicação externa. O host de endereço IP 172.30.20.100 é usado para testar, por exemplo, se os dados estão sendo devidamente coletados e registrados.

Os Honeypots DNS e NTP, além de serem alvos para ataques, também são usados, respectivamente, para resolução de nomes requisitados por outros Honeypots e sintonização dos relógios dos dispositivos, para registrar os pacotes em sincronia e possibilitar análise temporal do tráfego.

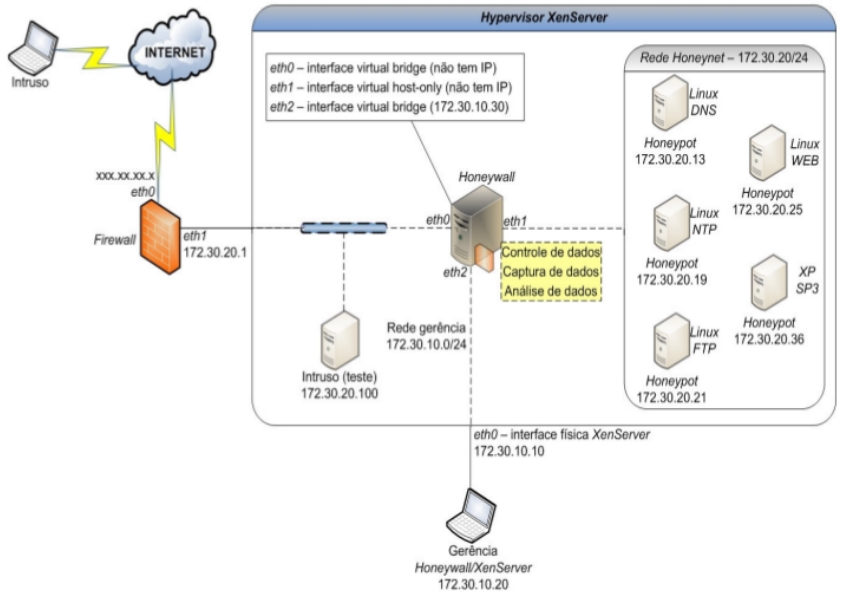

Figura 3.1: Arquitetura da Honeynet instalada no LATITUDE (OLIVEIRA JÚNIOR; SOUSA JÚNIOR; TENÓRIO, 2015).

3.1.2 Sensores do Norse

Norse, uma empresa internacional de segurança cibernética, possui uma rede de diversos Honeypots e sensores espalhados pelo mundo. Os dados de tráfego malicioso capturados por esses sensores são apresentados em tempo real no website da companhia[1].

Como mostra a Figura 3.2, esse site fornece detalhes sobre os ataques que são úteis para forense em redes, como instante de tempo do ataque, endereço IP de origem e seu georreferenciamento, alvo, porta e tipo do ataque. Além disso, o site permite aplicação de filtros regionais dos ataques, que mostram ataques destinados ou originados no Global, América Latina, Sudeste Asiático, Europa, Oeste Asiático, Estados Unidos & China.

[1]Site oficial: `http://map.norsecorp.com/#/`

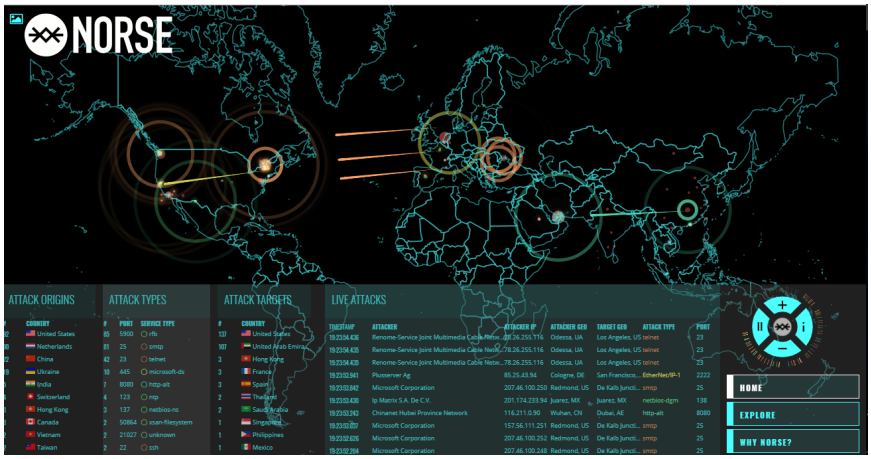

Figura 3.2: Visão em tempo real dos ataques detectados pelos sensores do Norse (NORSE, 2017).

A biblioteca BeautifulSoup, para Python, fornece uma API que possibilita a captura dos dados apresentados no navegador, ato chamado Web Scraping. Essa biblioteca é usada para coletar os dados apresentados acerca dos ataques, variando o filtro de região ciclicamente a cada hora, na sequência em que elas são apresentadas no parágrafo anterior. O código fonte em Python desse Web Scraper é dado no Anexo I.

3.2 Visualização e Análise de Dados

Em posse dos dados coletados das fontes apresentadas na seção 3.1, softwares e ferramentas são necessárias para fornecer a visualização das características do tráfego e evidenciar as anomalias presentes nele. Esta seção apresenta os dois softwares usados para isso.

3.2.1 Elastic Stack

Como discutido na seção 2.4, dados analisados em um processo de forense em redes são classificados como Mega-Dados, e exigem armazenamento, processamento e formas de análise apropriados.

Desenvolvido pela Elastic, os softwares de código aberto Logstash, Elasticsearch e Kibana viabilizam, respectivamente, a indexação, armazenamento e visualização de dados, com suporte a Mega-Dados.

Logstash é um *pipeline* de processamento de dados que recebe em sua entrada eventos, com suporte a vários formatos, os processa e estrutura em campos, para, então, os fornecer na saída. Para isso, deve-se configurá-lo com um arquivo de extensão *conf* que informa a entrada e a saída do *pipeline*, bem como suas regras de processamento. Neste projeto, pacotes do tráfego da Honeynet são tratados como eventos, e as informações contidas nas camadas desses pacotes são os campos. O arquivo *conf* do Logstash usado é dado no Anexo I.

O Logstash pode ser usado para indexar dados em qualquer ferramenta de busca, mas neste trabalho usa-se o Elasticsearch. Os dados indexados nessa ferramenta podem ser pesquisados e

filtrados de forma rápida e escalável, mesmo que os dados sejam volumosos e de diversas fontes e tipos. Isso é importante neste projeto, já que se procura anomalias e determinação de padrões em Mega-Dados.

Dados indexados no Elasticsearch podem ser visualizados com o Kibana em uma grande variedade de tipos de gráficos. Além das possibilidades padrões de visualização, uma extensão de grafos pode ser instalada para oferecer uma compreensão de como os campos se relacionam entre si. Além disso, tendo os dados indexados pelo Elasticsearch, a análise e visualização no Kibana é facilitada com uso de filtros que permitem, por exemplo, efetuar a inspeção em uma janela temporal ou limitar a investigação a um atacante ou protocolo específico. Essa possibilidade de especificar a análise facilita a identificação de comportamentos anômalos e eventos específicos em meio ao Mega-Dado. A Figura 3.3 mostra a tela inicial do Kibana com dois eventos indexados e seus respectivos campos.

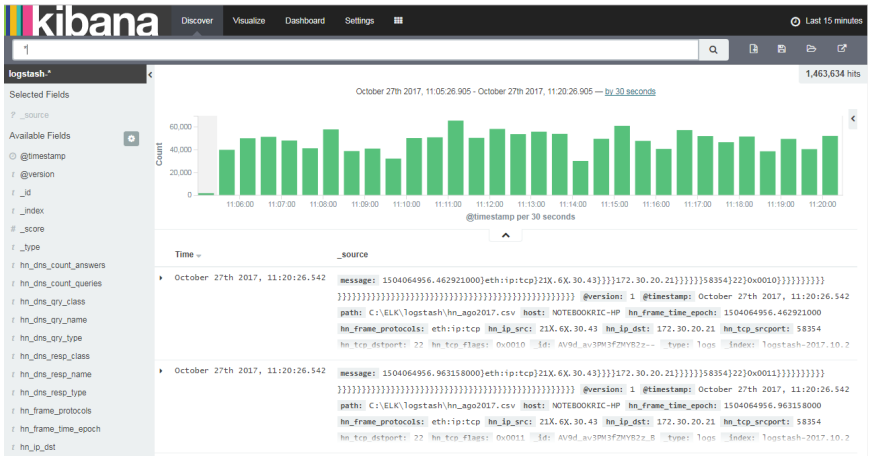

Figura 3.3: Visão inicial do Kibana, com dados sendo indexados.

3.2.2 Visualização Geográfica do Tráfego

Para adquirir conhecimento acerca da localização geográfica de origem dos ataques, desenvolveram-se duas aplicações em JavaScript que, após terem lido um arquivo de entrada contendo as informações dos ataques, os posicionam num Mapa-Múndi de acordo com sua origem, usando um círculo cujo tamanho é proporcional à quantidade de pacotes transmitidos por esse atacante e cujas cores diferenciam as portas alvejadas.

Uma dessas aplicações propicia maior interatividade com o usuário, que pode dar zoom e movimentar o mapa, enquanto a outra apresenta somente uma imagem estática. A Figura 3.4 exemplifica o uso dessas duas aplicações com um ataque direcionado à Honeynet.

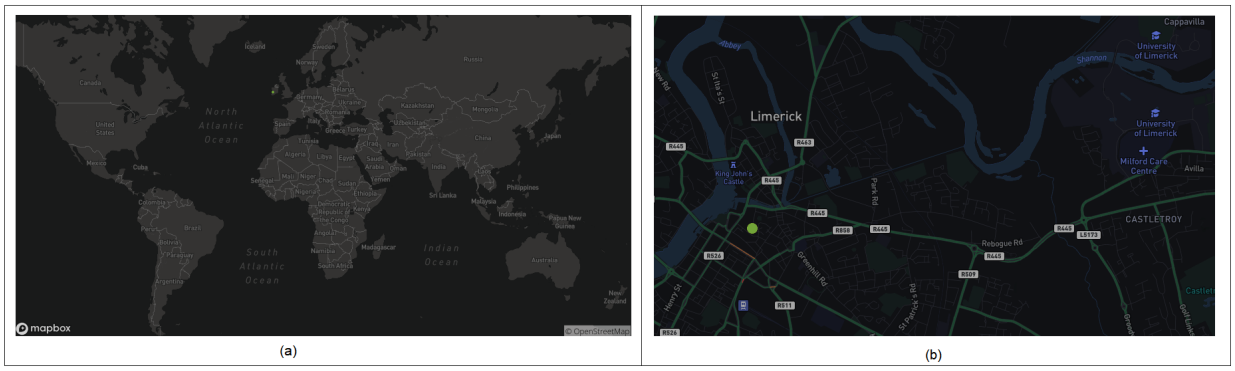

Figura 3.4: Visualização do mesmo ataque originado em Limerick, Irlanda, pelo mapa estático (a) e pelo mapa dinâmico (b).

Embora a aplicação do mapa dinâmico proporcione mais interação com o usuário, ela se torna lenta ao carregar uma grande quantidade de ataques visíveis na tela. Por isso, as duas aplicações complementam-se, sendo o mapa estático mais apropriado para análises globais sem exigir grande esforço computacional, e o mapa dinâmico para inspeções regionais, carregando uma quantidade reduzida de ataques. Os códigos fontes de ambas aplicações são disponibilizados no Anexo I.

3.3 Arquitetura para Análise dos Dados

Os arquivos pcap coletados da rede de gerência da Honeynet e analisados neste trabalho correspondem ao tráfego de agosto de 2016 até setembro de 2017. Como arquivos pcap não são legíveis por humanos, antes de sua indexação pelo Logstash é necessário convertê-lo em texto, no formato csv (exemplo de linhas na Figura 3.5a). Para isso, usa-se um comando TShark, que está disponível no Anexo I, especificando os campos que devem ser convertidos e o separador usado.

Para visualização geográfica dos ataques, os endereços IP presentes no arquivo de saída do TShark são georreferenciados, visto que, embora o arquivo csv informe país e cidade de origem do ataque, a latitude e longitude não estão presentes no dado.

Os mesmos campos definidos no comando TShark, e na mesma sequência, são configurados no *pipeline* do Logstash, para visualização no Kibana. A escolha do caractere de separação dos valores, tanto na conversão TShark quanto no Logstash, é importante e varia de acordo com o arquivo de dados a ser analisado. Por padrão, o separador é a vírgula, mas alguns campos exportados do TShark possuem esse caractere, como o país "Korea, Repubblic of"e o último campo na primeira linha da Figura 3.5a, o que prejudicaria a indexação caso esse caractere fosse o separador. Para evitar esse problema, usa-se o caractere '}' como separador, que não é comumente presente nesse conjunto de dados.

Como os dados extraídos do Norse não possuem vírgulas, esse caractere pode ser usado como separador sem gerar falhas na indexação, conforme mostra a Figura 3.5b. Os dados do Norse foram coletados entre junho e setembro de 2017.

```
1477958972}eth:ip:tcp:telnet}172.30.20.36}}}8x.21x.36.216}}}23}45058}0x0018}252}65}\x0a,\x0dlogin: }...}
1477959442}eth:ip:tcp:ssh}21x.8x.139.56}France}Paris}172.30.20.21}}}59183}22}0x0018}}}}}}}}}}}}}}}}}...}
1477959932}eth:ip:tcp}9x.4x.130.174}Belarus}Baranavichy}172.30.20.36}}}48701}23}0x0002}}}}}}}}}}}}}...}
```
(a)

```
2017-06-21, 21:18:54.252, Abts Tamilnadu, 12x.1xx.23.203, Chennai, IN, Lynnwood, US, telnet, 23, seAsia
2017-07-14, 15:08:32.111, Cantv Servicios Venezuela, 19x.xx.241.53, Caracas, VE, Roseville, US, telnet, 23, latAmer
2017-07-17, 23:56:57.471, Cgest S.A., 7x.xx.204.122, Braga, PT, Braga, PT, netbios-dgm, 138, eu
```
(b)

Figura 3.5: Exemplos de linhas do arquivo csv de saída do comando TShark (a) e resultante do Web Scraping do Norse (b). Reticiências são usadas para suprimir a repetição de campos vazios.

A Figura 3.6 esquematiza a arquitetura descrita.

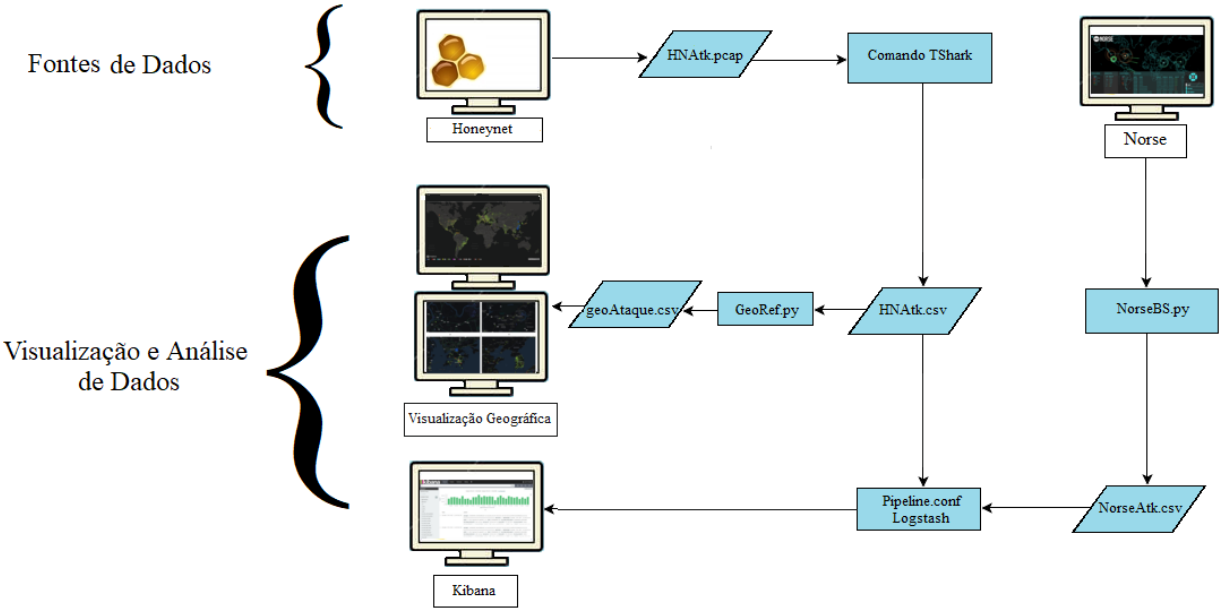

Figura 3.6: Arquitetura geral da coleta e análise dos dados.

Capítulo 4

Análise do Tráfego na Honeynet

A fim de se compreender e caracterizar o tráfego na Honeynet, não somente as anomalias propriamente ditas devem ser estudadas, mas também estatísticas e correlações entre campos dos pacotes fornecem dados informativos acerca de padrões e tendências dos atacantes.

Ainda, como DPI não é uma técnica eficiente para detecção de ataques na camada física, essa camada não é abordada nesse estudo. A análise na camada de enlace também não é feita, já que, como o tráfego vem da Internet, não retornaria informações úteis. Para estudos acerca dessas camadas, referir-se a Trappe (2015) e Zou et al. (2015), para a camada 1; e Kolias et al. (2016), para a camada 2.

Diante disso, este Capítulo apresenta estatísticas referentes às características presentes nas camadas 3 e 4 do tráfego malicioso, a princípio isoladamente, e, então, correlaciona-se as características das duas camadas, evidenciando o modo como comportam-se determinados usuários. Por fim, analisa-se e as propriedades anômalas da camada 5 de alguns protocolos explorados pelos atacantes.

Nos grafos apresentados neste Capítulo, a seguinte convenção é adotada: nós laranjas representam os Honeypots alvejados; nós violetas são o IP dos usuários maliciosos; os verdes representam os países a que es IPs maliciosos são georreferenciados; e os amarelos são os serviços explorados. Em todos os grafos, a largura da aresta indica o grau de correlação entre os nós conectados. Nos grafos e no restante deste trabalho, os endereços IP públicos são anonimizados, substituindo-se os últimos dígitos dos dois primeiros octetos por um X, para impedir a identificação exata dos dispositivos pelo público e garantir a privacidade dos usuários. Caso informação de georreferenciamento seja fornecida sobre esse endereço, mais dígitos são anonimizados, já que informação extra sobre o IP é fornecida e pode ser usada para revelar o endereço.

4.1 Camada de Rede

Uma inspeção nessa camada informa os endereços IP responsáveis pelos atos maliciosos e, com o georreferenciamento, é possível determinar os países onde estão esses dispositivos. Esse

estudo indicou que 164409 endereços IP únicos, associados a 169 países, transmitiram pacotes à Honeynet. No entanto, uma quantidade não conhecida desses endereços foi forjada (*spoofed*) para assumir os valores dos endereços das vítimas, como visto na seção 4.4.6, fazendo com que o valor real de endereços IP maliciosos seja menor. Esses endereços forjados, entretanto, não interferem na análise deste Capítulo. Os IPs e países mais frequentes no tráfego são mostrados na Figura 4.1.

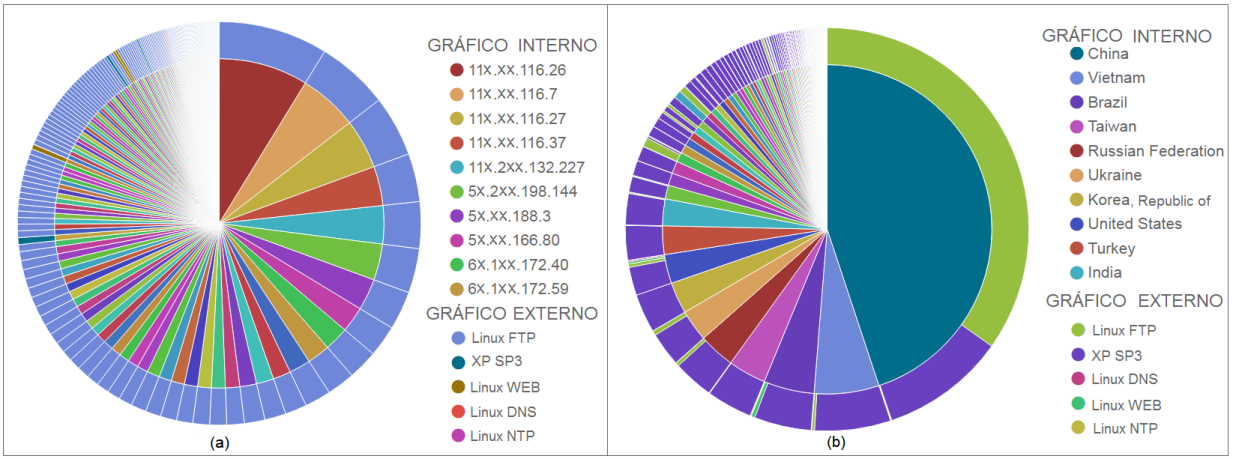

Figura 4.1: Endereços IP (a) e países (b) mais frequentes na Honeynet e os respectivos Honeypots alvejados.

A Figura 4.1a sugere uma preferência dos atacantes pelo Honeypot FTP. Todavia, essa figura refere-se somente aos IPs mais frequentes que são dominantemente georreferenciados à China, que, conforme a Figura 4.1b, focou seus ataques principalmente nesse Honeypot. É notável que ao incluir outros países na análise, como na Figura 4.1b, a preferência se dá pelo Honeypot XP.

Na Figura 4.1a, os endereços IP no intervalo 11x.xx.116.0/24 são georreferenciados à cidade de Shenzhen, na China, acusando a existência de uma sub-rede infectada ou montada para fins maliciosos. Essa constatação também é válida para os demais endereços IP maliciosos apresentados na figura.

A Figura 4.2 mostra que a preferência pelo Honeypot FTP se dá somente pelos dispositivos mais frequentes. Analisando as origens gerais do tráfego, a figura evidencia a preferência pelo Honeypot XP. Além disso, nota-se que os nós referentes aos Honeypots não são interconectados, o que significa que os atacantes transmitiram pacotes principalmente a um único Honeypot.

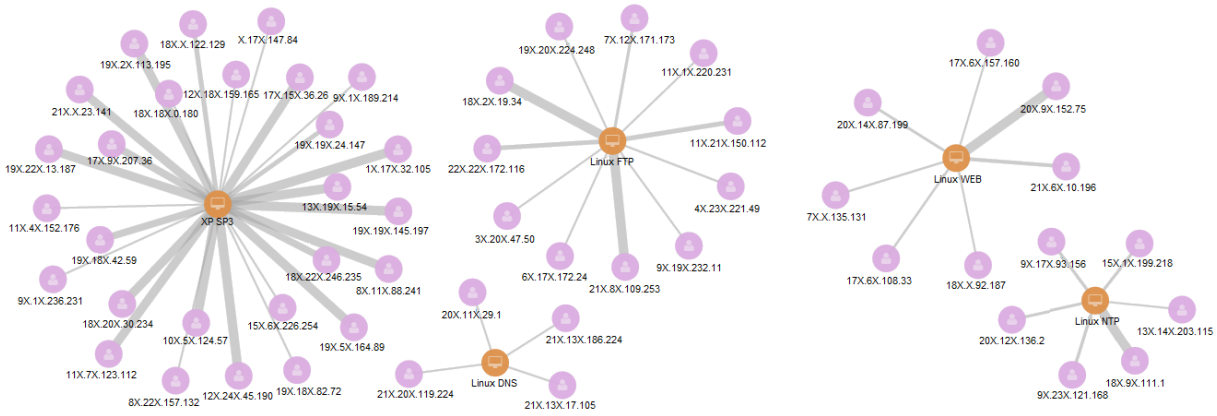

Figura 4.2: Relação entre endereços IP gerais e o Honeypot atacado.

Ao estender a análise de endereços IP aos países, depreende-se que a propensão para atacar o Honeypot XP também é válida, como mostra a Figura 4.3.

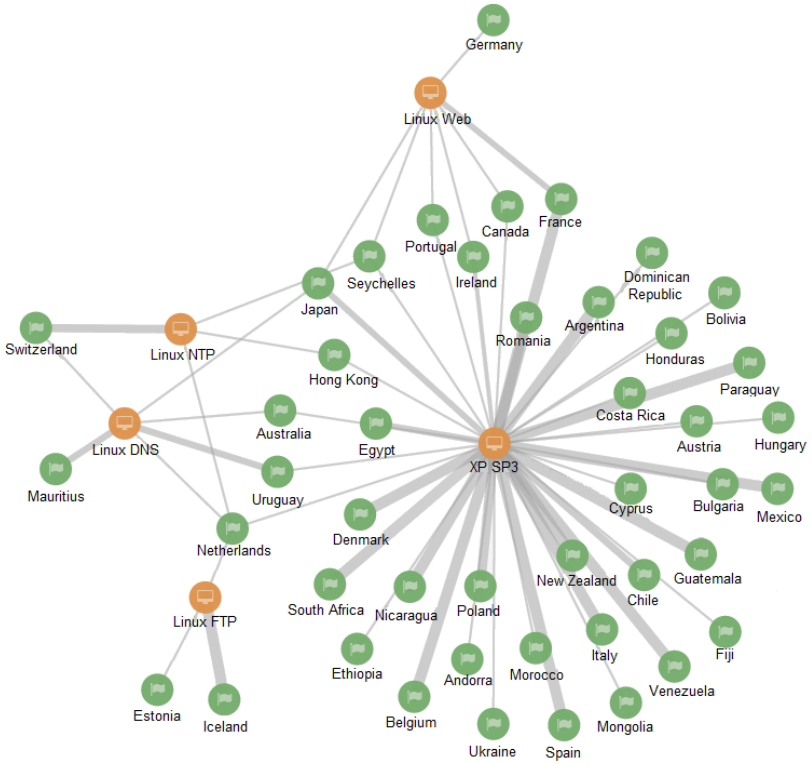

Figura 4.3: Relação entre países de origem gerais e o Honeypot atacado.

4.1.1 Visualização Geográfica

Embora a análise anterior forneça informações interessantes acerca dos países de origem dos ataques, ela não informa as cidades a que esses endereços IP são georreferenciados, que poderiam esclarecer padrões regionais, além de maior precisão de localidade.

Já que para desempenhar um ataque cibernético é necessário que haja uma infraestrutura básica, isto é, ao menos um dispositivo conectado à Internet, é esperado que esses ataques sejam originados de regiões mais urbanizadas.

A Figura 4.4 é uma imagem de satélite que retrata a noite no Planeta Terra. Naturalmente, regiões mais urbanizadas são caracterizadas pela maior concentração de luzes acesas. Nesse sentido, destacam-se a costa leste dos Estados Unidos, sudeste e litoral brasileiro, Europa como um todo, Índia, sudeste chinês, Japão e leste da Austrália. Na África, a concentração de luz aparece nos extremos Sul e Norte do continente. E no Egito, a luz delineia a margem do Rio Nilo, que é a base da civilização egípcia.

Figura 4.4: Terra de noite, indicando regiões mais urbanizadas pela iluminação.

A fim de se estudar padrões geográficos com maior clareza, os dados referentes ao tráfego malicioso na Honeynet é dado na entrada da aplicação de visualização geográfica estática. O mapa apresentado na Figura 4.5, referente a essa aplicação, mostra, com precisão de cidade, as origens do tráfego destinado à Honeynet. Nessa figura, diferentes cores de círculo representam diferentes portas alvejadas dos Honeypots, conforme a legenda, e o tamanho indica a quantidade de pacotes enviados pelo mesmo endereço IP — quanto maior o círculo, maior a quantidade de pacotes.

É possível notar que as regiões destacadas na Figura 4.4, que apresentam uma maior urbanização, são também destacadas por serem as responsáveis pelo maior tráfego malicioso à rede estudada, conforme a Figura 4.5.

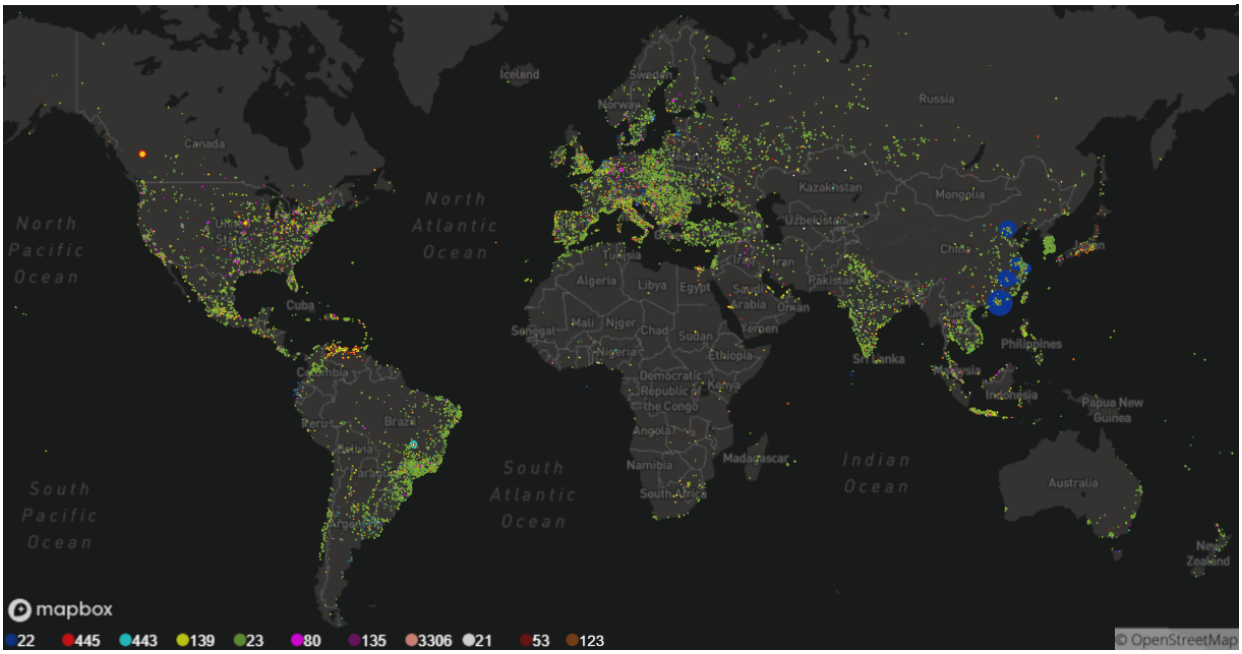

Figura 4.5: Distribuição geográfica das origens dos ataques direcionados à Honeynet.

A Figura 4.5 mostra uma tendência da maioria dos países de explorar a porta 23 (Telnet), conforme sugerido pela grande quantidade de círculos verdes no mapa. Explicita, também, uma grande quantidade de pacotes explorando a porta 22 (SSH) de usuários georreferenciados ao sudeste chinês, evidenciados pelos grandes círculos azuis nessa região. Nota-se, ainda, uma preferência da Venezuela em conduzir ataques à porta 139 (NetBIOS-SS), com a significativa quantidade de círculos amarelos nesse país, e provavelmente tendo o Honeypot XP SP3 como alvo, já que esse serviço é exclusivo de máquinas Windows. Outro detalhe notado nesse mapa é que pontos vermelhos e amarelos estão em geral acompanhados, como no oeste do Canadá, indicando que as portas 139 e 445 são de alguma forma correlacionadas.

Embora esse mapa forneça uma boa visualização macroscópica da origem dos ataques, ele falha em retornar informações específicas de regiões ou cidades, já que é uma imagem estática, impossibilitando o uso de zoom.

Essa outra aplicação permite o zoom e movimentação no mapa, possibilitando uma análise detalhada de países e cidades, conforme apresentado na Figura 4.6. A legenda das cores e tamanho dos círculos da Figura 4.5 também é válida na Figura 4.6.

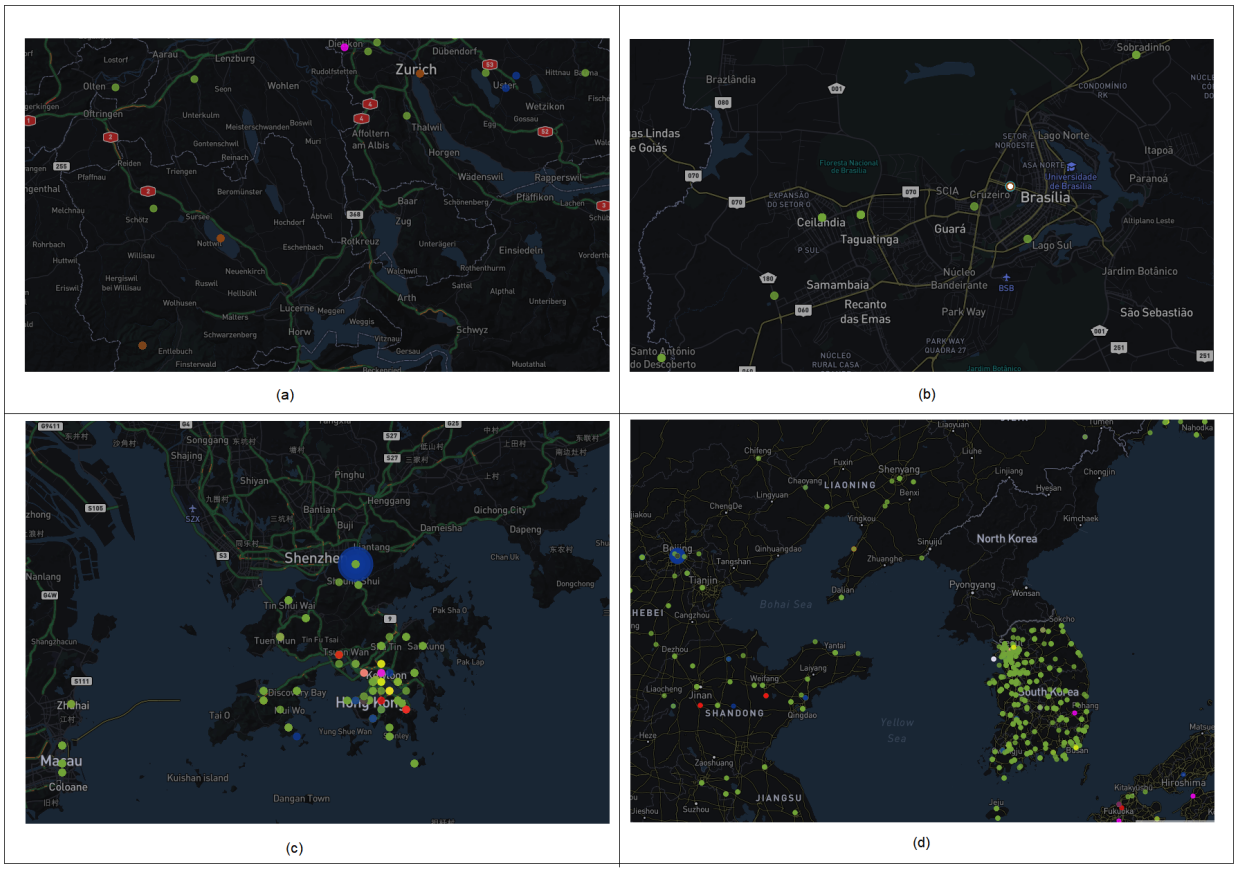

Figura 4.6: Visualização geográfica detalhada da origem dos ataques, focando em Suíça (a), Brasília - Brasil (b), Shenzhen - China (c) e Coreias do Sul e do Norte (d).

Com a análise mais detalhada, é possível focar em determinados países para se obter informações acerca de pacotes georreferenciados a eles. Um exemplo disso é a Figura 4.6a, que foca a análise na Suíça. Considerando a pequena quantidade de pacotes recebidos pelo Honeypot NTP, como mostrado na seção 4.3.3, a quantidade de círculos marrons (referentes a tráfego NTP) nesse país é significativa, o que indica que esse país é um relevante responsável pelos pacotes destinados à porta 123.

Esse mapa também evidencia ataques originados em cidades específicas, conforme a Figura 4.6b. Esse estudo minucioso permite ver, por exemplo, que uma única localidade executou três tipos de ataques em diferentes proporções, resultando na sobreposição dos círculos ao centro da cidade de Brasília: HTTPS, evidenciado pelo círculo azul claro maior; NTP, pelo círculo marrom mediano; e FTP, pelo círculo branco menor. Entretanto, esses ataques não necessariamente são feitos pelo mesmo endereço IP ou por máquinas localizadas exatamente nesses pontos, já que o georreferenciamento não é tão acurado.

Também é possível, com esse mapa, analisar em específico pontos que se destaquem na análise macroscópica. Como exemplo disso, a Figura 4.6c foca na região do sudeste chinês, que foi uma fonte significativa de ataques, explorando o protocolo SSH. A investigação detalhada dessa região informa que esses ataques são originados na cidade de Shenzhen (China), corroborando os resulta-

dos apresentados na seção 4.1, que sugere que esses pacotes foram destinados ao Honeypot Linux FTP. Essa figura mostra também uma grande quantidade de ataques alvejando diversas portas originados em Hong Kong.

A Figura 4.6d enfatiza a discrepância entre as Coreias: enquanto a do Sul, um país avançado em tecnologia, transmitiu uma grande quantidade de pacotes (majoritariamente Telnet e concentrados em Seul); a do Norte, devido ao regime ditatorial que proíbe o uso de computadores à maioria da população, não enviou um pacote sequer à Honeynet. Embora a Coreia do Norte conduza ataques cibernéticos, eles são usualmente feitos por agentes do governo e focados nos Estados Unidos e Coreia do Sul, buscando roubo de dados ou causando prejuízo a esses países (WARF, 2015). Portanto, uma desconhecida rede brasileira não lhes seria de interesse. Essa figura também mostra que Pequim (China) corresponde a uma das mais significativas origens de ataque SSH no sudeste da China, embora não tão expressiva quanto Shenzhen.

Essa análise reforça a importância das duas aplicações geográficas, já que o mapa dinâmico se torna lento ao carregar as 164409 origens únicas mundiais dos ataques, enquanto o mapa estático não fornece uma análise detalhada por região.

4.2 Camada de Transporte

Uma investigação nessa camada retorna, dentre outras informações, a porta a que cada pacote malicioso é destinado. Associando-se portas notórias ao seu serviço vinculado, é possível inferir quais protocolos da camada de aplicação foram explorados. A Figura 4.7 mostra a distribuição dos pacotes recebidos em relação a sua porta destino.

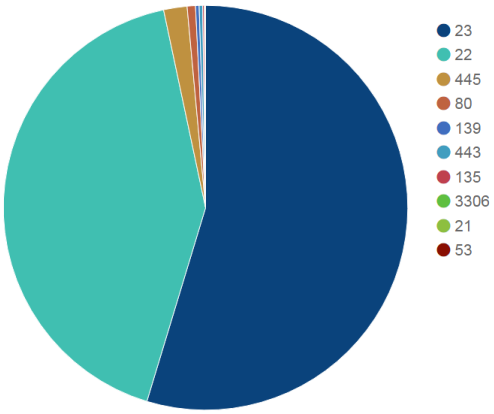

Figura 4.7: Serviços da Honeynet mais explorados.

Como os protocolos nas portas 22 (SSH) e e 443 (HTTPS) são criptografados, DPI não é uma técnica viável para avaliar o conteúdo dos pacotes. Uma opção é analisar os arquivos de log dessas conexões. A análise nesses arquivos sugere que os ataques SSH correspondem predominantemente a força bruta.

Já os pacotes destinados às demais portas são transmitidos em texto claro, permitindo o uso de

DPI. A análise do conteúdo malicioso presente em pacotes de alguns desses protocolos é apresentada na seção 4.4.

4.3 Correlação entre camadas de Rede e Transporte

A análise das seções 4.1 e 4.2 esclarece de onde o tráfego malicioso vem em sua maior parte, além de serviços e Honeypots mais explorados, mas malogra em fornecer uma visualização de como esses dados se relacionam. A correlação entre os campos de endereço IP de origem e seu georreferenciamento, da camada 3, e a porta destino, da camada 4, indica as preferências por tipos de ataques de determinados atacantes e países de origem.

4.3.1 País de origem e porta alvejada

A correlação entre esses campos retorna informações que auxiliam na compreensão dos tipos de ataque de preferência de cada país. Como visto na Figura 4.8, há uma maior inclinação dos países de origem a explorar o serviço Telnet, na porta 23. Constata-se também um alto grau de correlação entre Irlanda e a porta 80 (HTTP); Uzbequistão e a porta 21 (FTP); China e a porta 22 (SSH); e Suíça e a porta 123 (NTP), o que entra em acordo com a análise relativa à Figura 4.6.

Devido ao número limitado de nós e arestas, essa análise não necessariamente indica que a China só efetuou ataques na porta 22 ou que ataques nessa porta foram feitos somente por esse país, mas que a relação entre os dois foi mais significativa que entre os demais.

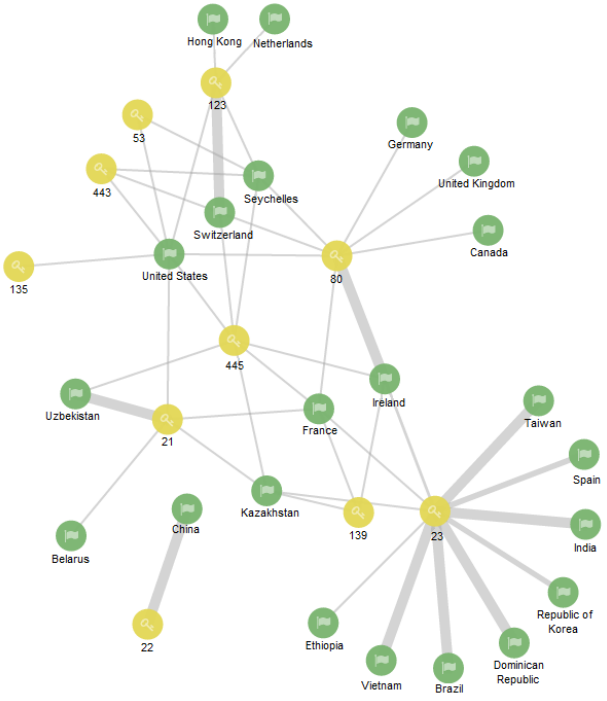

Figura 4.8: Relação entre país de origem dos ataques e serviço explorado.

4.3.2 Endereço IP e porta alvejada

A correlação entre esses campos das duas camadas é mostrada na Figura 4.9, que reforça a preferência dos atacantes à porta 23, já que uma quantidade significativamente maior de nós de IP está conectada ao nó desse serviço.

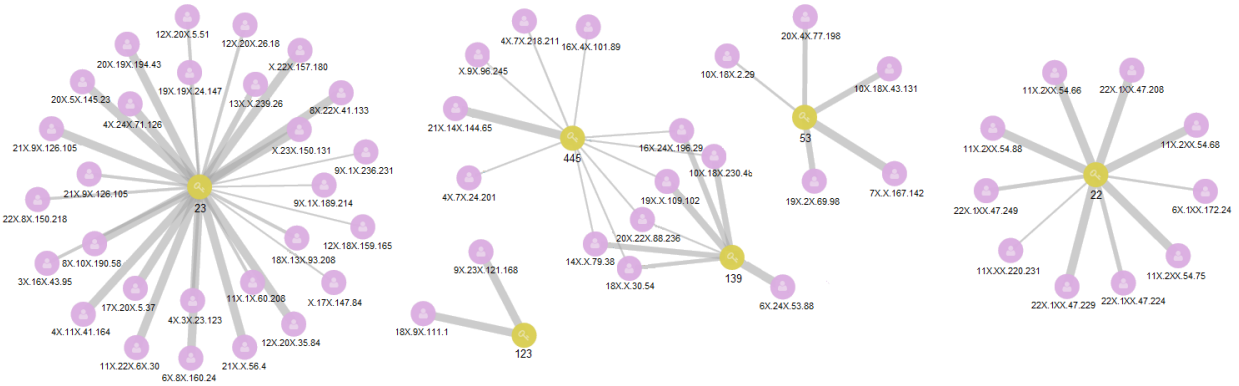

Figura 4.9: Relação entre endereço IP dos atacantes e serviço explorado.

Além disso, todos os nós conectados ao nó da porta 22 são georreferenciados à China, o que reforça a preferência dos usuários desse país a realizar ataques SSH, conforme indicado nas seções 4.1 e 4.3.1.

Ainda na Figura 4.9, é possível notar que os nós de serviço não são interconectados, sugerindo que os dispositivos foram usados em ataques majoritariamente em uma única porta, indicando as preferências dos atacantes. A única interconexão se dá entre os nós das portas 139 (NetBIOS-SS) e 445 (Microsoft-DS), o que pode ser explicado pelo comportamento dos protocolos sob essas portas, explicado na seção 2.1.3. Muitos malwares são conhecidos por explorar vulnerabilidades nessas portas, como o W32IRC (GU, 2008), que alvejou a Honeynet (PIMENTA RODRIGUES et al., 2017) e, portanto, é o provável causador dessa interconexão. Esse malware é estudado na seção 4.4.2 e explica o comportamento notado entre os pontos vermelhos e amarelos da Figura 4.5.

4.3.3 Honeypot e porta alvejados

É esperado que Honeypots rodando um dado serviço tenham esse serviço como o mais explorado. Como mostrado na Figura 4.10, esse comportamento é notado nos Honeypots, excetuando-se o Linux FTP. Ao invés de ter sido alvo frequente de ataques FTP, como força bruta, 99,92% dos pacotes recebidos por esse Honeypot são SSH. A análise nas seções 4.1, 4.3.1 e 4.3.2 indicam que esses ataques são procedentes majoritariamente da China.

A Figura 4.10 também revela que o Honeypot XP SP3 foi o mais alvejado, provavelmente por ser um sistema desatualizado e vulnerável, como provou o recente ataque do ransomware WannaCry (COUGHLIN, 2017). Dentre os Honeypots Linux, a preferência foi pelos que rodam o serviço FTP, WEB, DNS e NTP, nesta ordem. É evidente também que os Honeypots XP e Linux FTP juntos receberam quase o tráfego total.

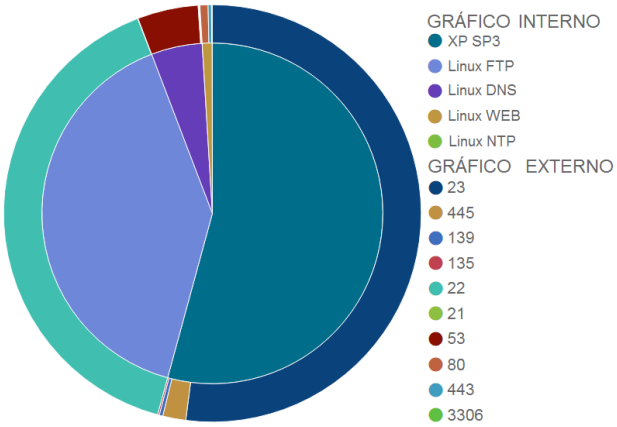

Figura 4.10: Honeypots atacados e seus respectivos serviços explorados.

Ademais, a alta relação entre os gráficos interno e externo da Figura 4.10 e a similaridade entre as Figuras 4.10 e 4.7 indicam uma alta correlação entre o Honeypot alvejado e o serviço explorado, isto é, cada Honeypot recebeu quase exclusivamente ataques explorando um único protocolo da camada de aplicação.

4.4 Camada de Aplicação

A camada de aplicação abrange diversos protocolos, usados com diferentes finalidades e, portanto, com características distintas, exigindo uma análise específica para cada. Essa seção estuda alguns dos protocolos dessa camada que foram abusados por usuários maliciosos, descrevendo o ataque, o modus operandi dos atacantes, e formas de prevenir e mitigar tais atividades.

4.4.1 Anomalia do tráfego para porta 23

Recentemente, o código da botnet Mirai, um malware que infecta dispositivos IoT rodando BusyBox, foi tornado público. Uma das principais características dessa botnet é a força bruta que executa, sob Telnet, dispositivos com credenciais fracas. Uma vez adquirido o controle do dispositivo, o malware reporta a infecção ao servidor de Comando e Controle (C&C), aderindo à botnet. Como muitos donos de dispositivos conectados à Internet nunca mudam os usuários e senhas padrões — ou mudam para outras credenciais fracas —, Mirai infectou mais de 380 milhões de dispositivos, que foram então usados em um dos maiores ataques Ataques de Negação de Serviço Distribuídos (DDoS) registrados até então, gerando pelo menos 1.1 Tbps na companhia francesa de computação na nuvem OVH, sediada em Roubaix, França (RADWARE, 2016).

Os comandos do ataque usualmente seguem a sequência a seguir:

`{usuario}` e `{senha}`, e então `enable` ou `system` ou `shell`, ou `sh`, e depois

`/bin/busybox MIRAI`,

onde `{usuario}` e `{senha}` são as tentativas do ataque de força bruta e são definidas no dicionário da Mirai. Os comandos seguintes são usados para detectar se o alvo é um roteador ou uma Honeypot de baixa interatividade comum.

Se a autenticação é bem sucedida, os seguintes comandos são executados para baixar a carga útil do malware e usar o recém infectado dispositivo para escanear outros alvos vulneráveis:

```
'busybox tftp' -r [MalwareFileName] -g [IPsource]
'busybox tftp' -g -l 'dvrHelper' -r [MalwareFileName] [IPsource].
```

Depois da execução, o malware deleta a si próprio, para evitar a breve detecção, e deixa o processo executando na memória.

Para checar o modus operandi do ataque da botnet Mirai na Honeynet, usa-se o grafo apresentado na Figura 4.11, onde os nós laranjas são o campo `Telnet data`; e os endereços IP responsáveis pelo ataque, os nós verdes.

Embora não sejam mostrados todos endereços IP nem os campos `Telnet data`, devido ao número limitado de nós e relacionamentos, pode ser inferido que diversos usuários transmitiram muitas vezes os comandos `enable`, `system`, `shell` e `sh`, que correspondem ao comportamento da Mirai, evidenciando que esse malware é massivamente presente no mundo.

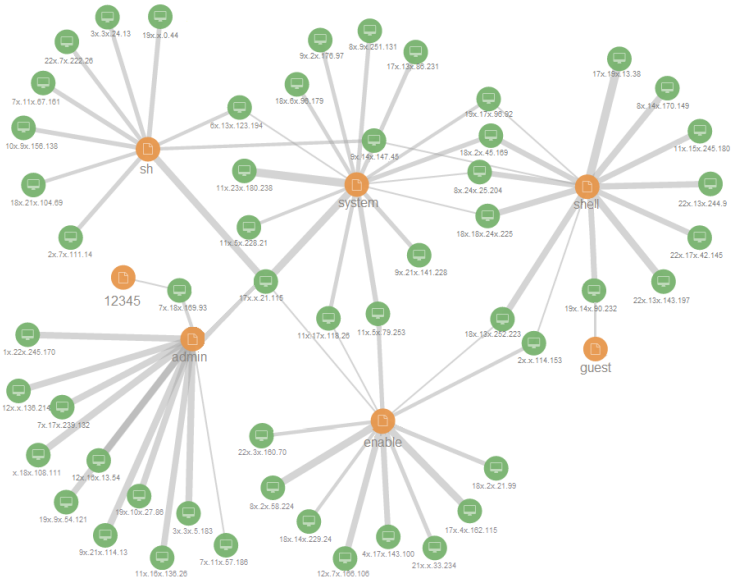

Figura 4.11: Honeypots atacados e seus respectivos serviços explorados.

Ao se explorar a carga útil do protocolo Telnet, foi possível ler o campo `Telnet data` e determinar quais foram as credenciais mais tentadas durante a força bruta da Mirai, mostradas na Figura 4.12, que evidencia a preferência pela *string root* tanto no usuário como na senha. É notado também pela figura que a quantidade única de usuários tentados é menor, concentrando-se em poucas *strings*, enquanto as senhas são mais distribuídas em uma maior quantidade de valores únicos. Isso

é demonstrado por se ter 98,86% dos pacotes contendo um dos 14 usuários apresentados, enquanto 15 senhas correspondem a 76,06%. As senhas mostradas na Figura 4.12 correspondem a senhas comuns de manufatureiros de dispositivos IoT chineses, como `xc3511` e `xmhdipc`, ou senhas fracas comumente usadas, como `root` e `123456`.

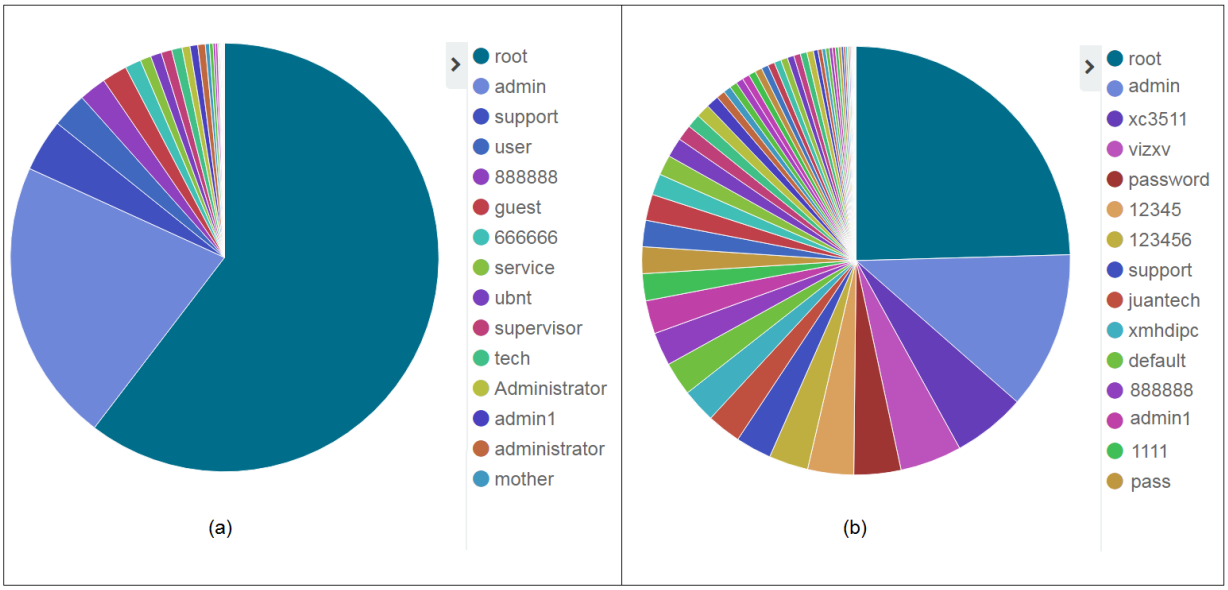

Figura 4.12: Honeypots atacados e seus respectivos serviços explorados.

Como esse malware transmite comandos específicos e identificáveis na carga útil do Telnet, sua detecção e prevenção não é dificultada, podendo configuras dispositivos DPI para bloquear tráfego com essas *strings* na carga útil. Ainda, as credenciais hardcoded tentadas pelo malware reforçam a importância de alterar usuários e senhas padrões dos dispositivos e usar credenciais fortes, que não podem ser facilmente adivinhadas.

4.4.2 Anomalia do tráfego para porta 139 e 445

W32.IRCBot-TO é um malware que cria outro tipo de botnet que abre um backdoor explorando uma vulnerabilidade de buffer overflow em sistemas Windows (CVE-2006-3439), e permite que atacantes executem códigos remotamente via uma mensagem forjada de Remote Procedure Call (RPC). A maior parte do tráfego dos dispositivos infectados por esse malware direcionaram pacotes às portas 139 e 445. O worm se espalha usando o protocolo Internet Relay Chat (IRC) como backdoor, e garante ao mestre da botnet acesso às máquinas infectadas, que podem ter informações roubadas e usadas para executar DDoS (NAZARYAN, 2017).

Seu modus operandi é baixar e executar arquivos binários chamados *netadp.exe* e, então, escanear por outras vítimas vulneráveis. Os campos analisados nesse protocolo são `path` e `file`, aos quais, na presença do malware, são atribuídos os respectivos valores de `\\<target-ip>\IPC\$` e `\\browser` ou `\\srvsvc`.

A sequência de pacotes (SEQ1), transmitidos por um atacante de endereço IP 9x.xx.53.209,

georreferenciado à cidade de Dublin, Irlanda, corresponde a esse malware atacando o Honeypot XP. O primeiro número de cada linha é o número do pacote transmitido, e a seta representa sua direção.

```
SEQ 1

523379 <-> 9x.xx.53.209 TCP 1963 - 172.30.20.36 TCP 139 [SYN, SYN,ACK]
523380 -> SMB Negotiate Protocol Request
523385 <- SMB Negotiate Protocol Response
523387 -> SMB Session Setup AndX Request, NTLMSSP_NEGOTIATE
523388 <- SMB Session Setup AndX Response, NTLMSSP_CHALLENGE, Error:  STATUS_MORE_PROCESSING_REQUIRED
523389 -> SMB Session Setup AndX Request, NTLMSSP_AUTH, User:  \
523390 <- SMB Session Setup AndX Response
523391 -> SMB Tree Connect AndX Request, Path:  \\<honeypot-public-Ip>\IPC$
523392 <- SMB Tree Connect AndX Response
523393 -> SMB NT Create AndX Request, Path:  \srvsvc
523394 <- SMB NT Create AndX Response, FID: 0x0000, Error:  STATUS_ACCESS_DENIED
523394 <- SMB NT Create AndX Response, FID: 0x0000, Error:  STATUS_ACCESS_DENIED
523397 <- SMB NT Create AndX Response, FID: 0x800e
523398 -> DCERPC Bind:  call_id:  1, UUID: SRVSVC
523399 <- SMB Write AndX Response, FID: 0x800e, 116 bytes
523401 -> SMB Read AndX Request, FID: 0x800e, 1024 bytes at offset 0
523402 <- SMB Bind_ack:  call_id:  1, result:  Provider rejection
```

Depois do handshake TCP, o atacante tentou conectar-se à vítima e fazer o download do arquivo *netadp.exe*, usando o compartilhamento IPC para conectar-se ao pipe SRVSVC. Entretanto, ele não foi bem sucedido nesse ataque por ter acesso negado pelo Honeypot. Como referência, a sequência de pacotes (SEQ2), obtida do trabalho de Gu (2008), que corresponde a um ataque bem sucedido do W32.IRCBot-TO.

```
SEQ 2

6 <-> <infector-ip> TCP 2971 - <honey-ip> 445 [SYN, SYN,ACK]
13 -> SMB Negotiate Protocol Request
14 <- SMB Negotiate Protocol Response
17 -> SMB Session Setup AndX Request, NTLMSSP_AUTH, User:  \
18 <- SMB Session Setup AndX Response
19 -> SMB Tree Connect AndX Request, Path:  \\<honey-ip>\IPC\$
20 <- SMB Tree Connect AndX Response
21 -> SMB NT Create AndX Request, Path:  \browser
22 <- SMB NT Create AndX Response, FID: 0x4000
23 -> DCERPC Bind:  call_id:  0 UUID: SRVSVC
```

```
24 <- SMB Write AndX Response, FID: 0x4000, 72 bytes
25 -> SMB Read AndX Request, FID: 0x4000, 4292 bytes at offset 0
26 <- DCERPC Bind_ack
27 -> SRVSVC NetrpPathCanonicalize request
28 <- SMB Write AndX Response, FID: 0x4000, 1152 bytes
29 -> SMB Read AndX Request, FID: 0x4000, 4292 bytes at offset 0

Initiating Egg download
30 <-> <honey-ip> TCP 1028 - <infector-ip> 8295 [SYN, SYNACK]
34-170 114572 byte egg download ...

Connecting to IRC server on port 8080
174 <-> <honey-ip> TCP 1030 - 66.25.XXX.XXX 8080 [SYN, SYNACK]
176 <- NICK [2K|USA|P|00|eOpOgkIc]\r\nUSER 2K-USA
177 -> :server016.z3nnet.net NOTICE AUTH
 :*** Looking up your hostname...\r\n'' ...
179 -> ...  PING :B203CFB7
180 <- PONG :B203CFB7
182 -> Welcome to the z3net IRC network ...

Joining channels and setting mode to hidden
183 -> MODE [2K|USA|P|00|eOpOgkIc] +x\r\nJOIN ##RWN irt3hrwn\r\n

Start scanning 203.0.0.0/8
185 -> ....scan.stop -s; .scan.start NETAPI 40 -b -s;
.scan.start NETAPI 203.x.x.x 20 -s;
.scan.start NETAPI 20 -a -s;.scan.start SYM 40 -b -s;
.scan.start MSSQL 40 -b -s\r\n...
191 -> 203.7.223.231 TCP 1072 > 139 [SYN]
192 -> 203.199.174.117 TCP 1073 > 139 [SYN] scan,scan...
```

Embora na (SEQ1) de pacotes o ataque não tenha sido executado com êxito, a similaridade nos dois conjuntos de pacotes é evidente e corresponde ao modus operandi (SEQ2) de W32.IRCBot. Se a infecção for sucedida, depois do malware instalado, ele se conecta ao servidor IRC e escaneia por outros alvos, criando a entrada de registro a seguir, para executar *netadp.exe* ao iniciar a máquina infectada:

```
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Network Bridge

<System>\netadp.exe
```

A presença dessa entrada é um *Indicator Of Compromise* (IOC) e implica na infecção da

máquina por W32.IRCBot-TO.

4.4.3 Anomalia do tráfego para porta 80

Para inspeção da carga útil do HTTP, analisou-se os campos `HTTP request method`, que retorna o verbo HTTP usado na requisição, e `HTTP request URI`, que contém o URI a que o cliente tenta se conectar.

Nessa investigação notou-se que requisições maliciosas e automáticas foram enviadas ao Honeypot Web, com a intenção de coletar informação útil para futuras explorações. A Tabela 4.1 mostra algumas dessas requisições e as ferramentas a que elas estão associadas. A maior parte dessas ferramentas também possibilita o escaneamento na porta 443 (HTTPS) e, portanto, parte do tráfego cifrado para essa porta pode corresponder a elas.

Além das requisições de escaneamento, um worm conhecido como *The Moon* também foi pego no tráfego à Honeynet. Esse worm infecta roteadores Linksys explorando uma vulnerabilidade que garante o acesso a esses dispositivos sem autenticação.

The Moon enviou uma requisição `GET /HNAP1 /HTTP/1.1` para o Honeypot, que é usado para identificar o modelo e versão do firmware do roteador. Em seguida, transmitiu os scripts `tmUnblock.cgi` e `hndUnblock.cgi` que permitem execução de comandos sem a necessidade de autenticação. Após infectar seu alvo, o worm usa o roteador para se espalhar para outros dispositivos vulneráveis.

Tabela 4.1: Requisições maliciosas que foram direcionadas ao Honeypot Web.

Verbo	URI	Ferramenta associada
`HEAD`	`http://18x.16x.113.82/check_proxy HTTP/1.1`	Escaneando tipo de proxy em uso
`HEAD`	`/robots.txt HTTP/1.0`	Escaneando quais bots são permitidos no servidor
`GET`	`/muieblackcat HTTP/1.1`	*muieblackcat* é um bot
`GET`	`//pma/scripts/setup.php HTTP/1.1`	que escaneia
`GET`	`/w00tw00t.at.blackhats.romanian.anti-sec:) HTTP/1.1`	vulnerabilidades PHP
`GET`	`/phpmyadmin/scripts/setup.php HTTP/1.0`	*ZmEu* é um bot que
`GET`	`/dbadmin/scripts/setup.php HTTP/1.1`	escaneia
`GET`	`/mysqladmin/scripts/setup.php HTTP/1.1`	vulnerabilidades
`GET`	`/admin/phpmyadmin/scripts/setup.php HTTP/1.1`	phpMyAdmin. Também
`GET`	`/admin/pma/scripts/setup.php HTTP/1.10`	executa força bruta SSH
`GET`	`/MyAdmin/scripts/setup.php HTTP/1.1934`	
`GET`	`/nmaplowercheck1487075443 HTTP/1.1`	*Nmap* sonda
`GET`	`/nice%20ports%2C/Tri%6Eity.txt%2ebak HTTP/1.0`	informações do servidor

As requisições mostradas na Tabela 4.1 e as associadas ao *The Moon* não são associáveis a tráfego legítimo, e a existência delas indica atividade suspeita. Com isso, o dispositivo DPI deve bloquear tráfego semelhante. Referente a outras explorações nessa porta, DPI também pode ser usado, por exemplo, em firewalls de camada de aplicação, para detectar e responder a pacotes maliciosos, como *strings* que evidenciam injeção de código.

4.4.4 Anomalia do tráfego para porta 21

Embora a análise da seção 4.3.3 mostre que a quase totalidade do tráfego direcionado ao Honeypot FTP tenha sido pelo protocolo SSH, alguns pacotes correspondem ao FTP. Esse Honeypot é protegido por uma forte senha, para impedir que seja invadido por atacantes que poderiam, consequentemente, armazenar malwares e arquivos ilegais nesse servidor.

Tentando adquirir acesso ao Honeypot, usuários executaram um ataque de força bruta, adivinhando usuários e senhas fracas e tipicamente usadas. A Figura 4.13 mostra a distribuição dos usuários e senhas tentados pelos atacantes na porta 21 desse Honeypot. A figura evidencia que uma parte significativa dos atacantes tentou conectar-se ao serviço anônimo do FTP, fornecendo a *string* `anonymous` como usuário. A exigência comum de uma identificação mínima do FTP anônimo explica a existência de senhas que se assemelham a um endereço de e-mail, com o caractere '@'.

A figura também mostra que a distribuição de usuários é mais concentrada em poucas *strings*, cada uma com grande representação do tráfego total. Por outro lado, a distribuição das senhas se dá numa maior quantidade de *strings* menos significativas no tráfego total. De fato, os atacantes tentaram usar apenas 27 nomes de usuário únicos, com um total de 115 senhas diferentes.

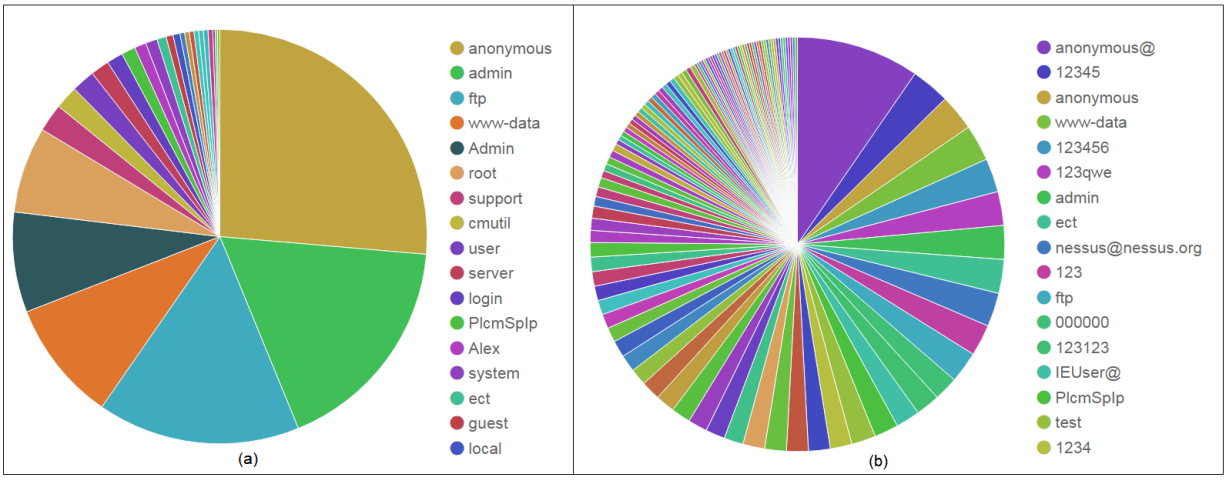

Figura 4.13: Distribuição dos usuários (a) e senhas (b) tentadas durante o ataque de força bruta na porta 21.

A análise do tráfego nesse ataque de força bruta mostrou que dois endereços IP georreferenciados à Brasília, 17x.1xx.231.151 e 17x.xx.151.232, usaram senhas que evidenciam que possuíam conhecimento interno da rede do LATITUDE, onde a Honeynet é instalada. Na tarde do dia 14 e

na manhã do dia 15 de dezembro de 2016, ambos os endereços tentaram conectar-se ao FTP anônimo usando e-mails no domínio local do Departamento de Engenharia Elétrica da UnB. Embora muitos desses e-mails sejam inexistentes, 17x.1xx.231.151 destacou-se por fornecer como senha os verdadeiros endereços de e-mail pessoal de um professor do Departamento, e pessoal e corporativo do administrador da rede local. Os ataques de força bruta FTP originados desses endereços IP foram mostrados no mapa da cidade de Brasília da Figura 4.6.

4.4.5 Anomalia do tráfego para porta 53

Usuários mal intencionados podem explorar o funcionamento do DNS para executar DDoS reflexivos. Neste ataque, um usuário faz diversas requisições de resolução de nome a um servidor DNS com o endereço IP forjado para o de sua vítima. Essas requisições normalmente são do tipo `ANY`, que retorna todas informações conhecidas sobre a zona DNS. Desta forma, as respostas são muito maiores do que as requisições, e são enviadas à vítima.

Este ataque é portanto classificado como de amplificação, por aumentar o tráfego do atacante e direcioná-lo à vítima. Uma métrica importante desse tipo de ataque é o fator de amplificação, que é dado pela razão entre a banda consumida na rede da vítima e a banda consumida na rede do usuário malicioso. O ataque é também classificado como reflexivo, por usar um terceiro como intermédio.

Para detecção dessa atividade maliciosa, analisou-se os endereços IP de destino e origem, da camada 3, e o tempo de transmissão do pacote, da camada 2, além do `DNS query name`, da camada 5, que contém o nome que o cliente deseja resolver. A Figura 4.14 mostra dados interessantes a respeito do comportamento malicioso na porta 53.

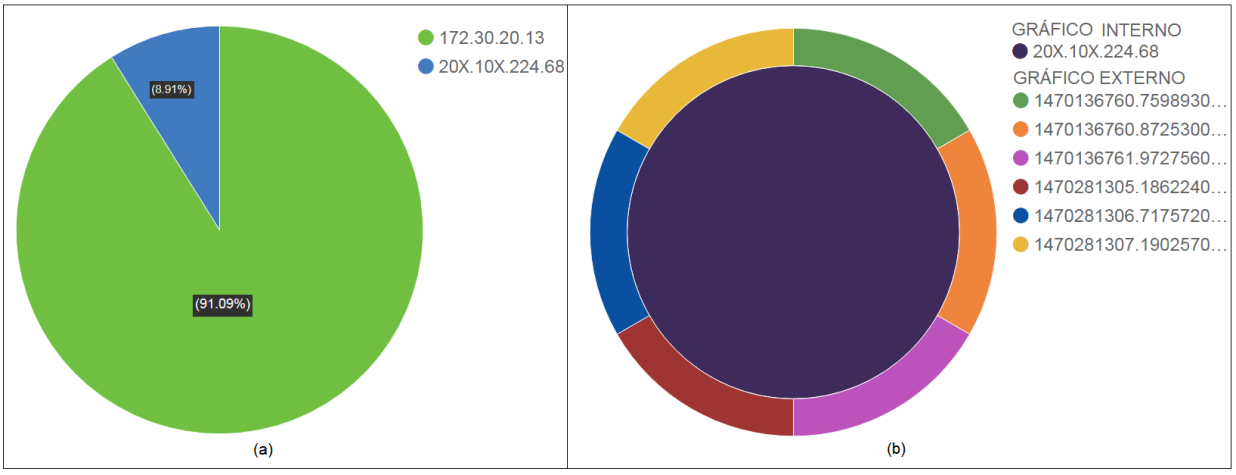

Figura 4.14: Distribuição das origens dos pacotes na conversa entre Honeypot DNS e um de seus clientes (a) e quantidade de requisições feita por tempo por esse cliente (b).

Tipicamente, o protocolo DNS usa o protocolo UDP da camada de transporte, com carga útil de 512 bytes. Contudo, para se enviar pacotes maiores, pode-se optar por usar TCP, mas isso aumenta a dificuldade do ataque, por ser muito mais difícil forjar endereços IP por esse protocolo.

Uma quantidade considerável de tráfego DNS usando TCP foi gerada na honeynet evidenciando as intenções dos atacantes de se alcançar grandes pacotes de resposta. Também para aumentar esse limite do tamanho de pacotes UDP, servidores DNS podem adotar uma extensão DNS (EDNS0) que permite respostas UDP de até 4096 bytes (ROSSOW, 2014), aumentando substancialmente o fator de amplificação do ataque.

Com essa extensão, as requisições e respostas do Honeypot DNS mostradas na Figura 4.14a têm tamanho de 4096 bytes. As requisições, em azul na figura, são do tipo `ANY`, o que resulta numa maior quantidade de respostas do que de requisições. O atacante, tendo o endereço IP forjado para o da vítima 20x.10x.224.68, fez 256 requisições que resultaram em 2616 respostas do Honeypot. Como os 2872 pacotes dessa conversa têm tamanho de 4096 bytes, esse ataque se deu com um fator de amplificação (em inglês, *Bandwidth Amplification Factor* - BAF) de 10,21875. De acordo com Vaughn e Evron (2006), combinando diferentes tipos de respostas, o efeito de amplificação pode alcançar valores não muito maiores que 60. Os autores também citam um valor médio do BAF entre 16 e 19 para esse ataque no conjunto de dados estudados por eles, que é um valor próximo do obtido neste trabalho.

Como mostra a Figura 4.14b, essas requisições estão distribuídas em pequenos intervalos de tempo de dois dias distintos. Os valores apresentados na figura são dados numa representação de data e hora denominada Epoch que, em Unix, corresponde aos segundos passados desde primeiro de Janeiro de 1970 às 00:00:00. Os valores mostrados equivalem a 2 de agosto de 2016 às 08:19 e 4 de Agosto de 2016 às 00:28, horário de Brasília. Como visto na figura, múltiplas requisições foram feitas quase simultaneamente, todas solicitando a resolução do mesmo nome, o que é atípico num tráfego DNS legítimo devido ao cache que existe no cliente. Esse comportamento evidencia o propósito do atacante de sobrecarregar a vítima, gerando muito tráfego em curto intervalo de tempo, e foi notado tanto com resolução quanto com resolução reversa de nomes.

A Extensão de Segurança do DNS (DNSSec), que adiciona criptografia assimétrica no tráfego desse protocolo, ao contrário do que se espera, não ajuda a prevenir ou mitigar esse ataque, mas, de acordo com Cowperthwaite e Somayaji (2010), aumenta seu fator de amplificação, já que a extensão de segurança requer o uso de EDNS0. Assim, uma forma da vítima mitigar esse ataque é bloqueando servidores DNS usados para esse fim; e para servidores DNS não serem usados maliciosamente, uma medida cabível é o bloqueio de repetidas requisições do mesmo cliente num curto período.

ICAN-SSAC (2006) recomenda técnicas que auxiliam na mitigação desse ataque, como a validação de endereços IP de origem (e.g. RFC 2827) e o uso de configurações apropriadas em servidores e desabilitar DNS recursivo, que reduz significativamente o BAF desse ataque. Outros autores, como Vaughn e Evron (2006), propuseram métodos de detecção desse ataque por meio de monitoração da rede, alcançando bons resultados.

4.4.6 Anomalia do tráfego para porta 123

Atacantes podem explorar servidores NTP para efetuar ataques semelhantes àquele descrito na seção 4.4.5, isto é, um ataque DDoS reflexivo e de amplificação, abusando do comando *Monlist*, presente em servidores NTP.

Esse comando retorna ao solicitante os endereços IP das últimas 600 máquinas com as quais o servidor interagiu, sendo limitado a 100 segmentos UDP, com 440 bytes de carga útil em cada (ROSSOW, 2014). Esse comando foi disponibilizado em servidores NTP para fornecer meios de gerenciamento e depuramento para seus administradores, mas atacantes podem fazer essa requisição com fins de reconhecimento, obtendo os endereços IP com intenções maliciosas, ou para realizar ataques DDoS.

O serviço NTP roda sobre o protocolo UDP da camada de transporte, que facilita a forja de endereços IP, se comparado ao protocolo TCP. Forjando esse endereço para o de sua vítima, usuários mal intencionados podem fazer uma requisição *Monlist*, usualmente de poucos bytes, a um servidor NTP, fazendo com que sua vítima receba as informações nas respostas do servidor, consideravelmente maiores.

Embora o fator de amplificação desse ataque usando a Honeynet não tenha sido calculado, Rossow (2014) o estudou e concluiu que ataques de amplificação NTP atingem fatores de amplificação que variam entre 556.9 e 4670.0, muito maior que o fator do ataque DNS, mostrado na seção 4.4.5. Portanto, é muito mais eficiente o uso de servidores NTP do que DNS como intermédio desse ataque, o que também foi notado por Rossow (2014).

Analisando o campo `NTP private request code`, notam-se 2202 requisições `MON_GETLIST_1` (código 42), associadas a essa vulnerabilidade, que é identificada como CVE-2013-5211, e apenas 26 requisições do tipo `REQUEST_KEY` (código 32). Essa proporção é mostrada na Figura 4.15a.

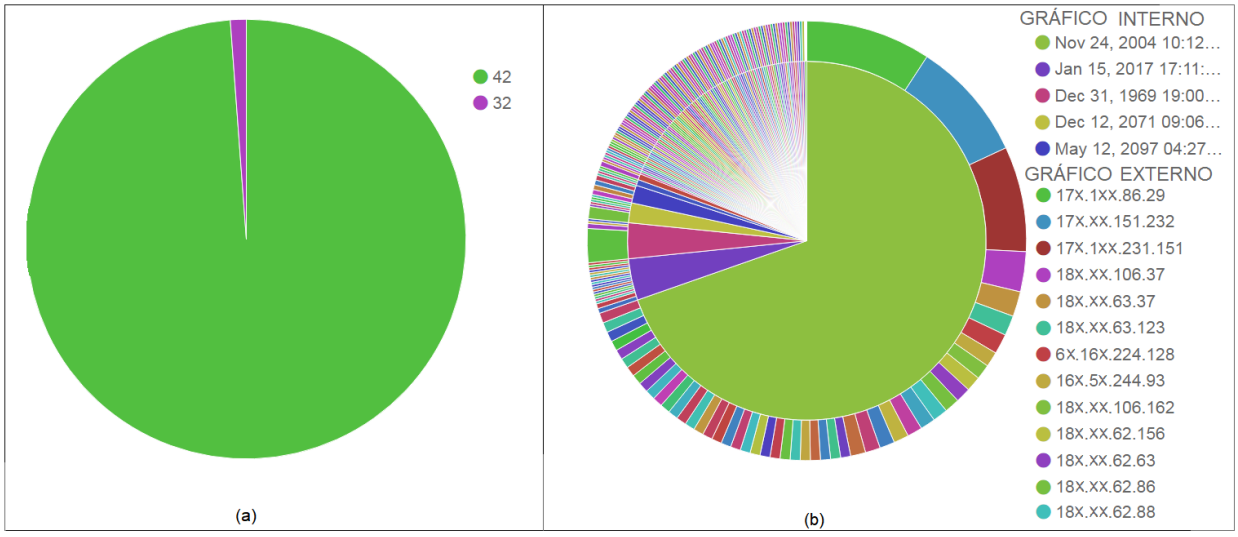

Figura 4.15: Distribuição dos tipos de requisições NTP (a) e dos valores do campo *transmit timestamp*, com seus respectivos endereços IP de origem (b).

Uma anomalia interessante é notada no campo `NTP transmit timestamp`. Já que os dados analisados são de 2016 e 2017, espera-se que esse campo tenha valores definidos nesse intervalo. Entretanto, a Figura 4.15b mostra que a maior parte dos pacotes NTP têm uma data e hora de transmissão desatualizados, enquanto outra parte significativa desse tráfego tem valores que referenciam ao futuro.

Além disso, nota-se que muitos endereços IP distintos enviaram a mesma marca temporal, com precisão de milionésimo de segundo. Isso sugere que esse pacotes foram, na verdade, transmitidos por um único dispositivo que definiu essa marca, mas forjando os endereços IP para os que aparecem na Figura 4.15b. Esses endereços são, portanto, as vítimas desse ataque, que receberão o volumoso tráfego indesejado. Nota-se também que as marcas temporais anômalas foram transmitidas por endereços IP em comum, como 24 de novembro de 2004 e 31 de dezembro de 1969, enviados pelo IP 17x.1xx.86.29, georreferenciado à cidade de Brasília.

O tráfego NTP referente aos três endereços georreferenciados a Brasília (os três primeiros da Figura 4.15b) foi mostrado no mapa dessa cidade (ver Figura 4.6b), corroborando a análise geográfica feita. Entretanto, o mapa mostra ataques NTP e FTP georreferenciados às mesmas coordenadas geográficas, possivelmente indicando que são originados do mesmo endereço IP. Mas os endereços georreferenciados à Brasília mostrados na seção 4.4.4 diferem dos mostrados nesta seção, o que anula essa indicação. Isso se dá pelo fato do georreferenciamento não ser acurado ao ponto de determinar a localização exata desses dispositivos, mas eles provavelmente se encontram próximos entre si.

Ainda, os endereços IP cujo primeiro octeto é 18x são georreferenciados à Suíça e representam uma parcela considerável das vítimas alvejadas nesse ataque NTP. Isso indica que o atacante alvejou uma subrede em específico desse país e corrobora os resultados apresentados na seção 4.3.1. Provavelmente referem-se aos círculos marrons indicados no mapa da Figura 4.6.

Para impedir essa exploração do serviço NTP, o comando `Monlist` deve ser bloqueado para endereços IP públicos e dispositivos DPI configurados para bloquear pacotes NTP com código de requisição igual a 42. Outros mecanismos de defesa incluem a atualização dos servidores NTP para a versão 4.2.7 p26 ou maior, removendo a funcionalidade `Monlist` e implementar BCP 38 (RFC 2827) na rede (ARUKONDA; SINHA, 2015). Essa análise reforça, portanto, a importância de se ter sistemas atualizados para impedir atividade maliciosa. Enfatizando a potência de amplificação explorando o serviço NTP, Kührer et al. (2014) realizaram uma campanha que reduziu o número de servidores NTP vulneráveis a esse ataque em mais de 92%, reduzindo a ocorrência da exploração dessa vulnerabilidade.

4.5 Comparação dos dados da Honeynet e do Norse

Com os quatro meses significativos de dados capturados do Norse, notaram-se 1879 endereços IP únicos tidos como origem de tráfego malicioso. Entretanto, em apenas um ciclo, isto é, com os dados coletados por somente uma hora em cada região (6 horas no total), 1864 endereços diferentes foram coletados. A correspondência de 99,20% de todos os endereços IP em um curto intervalo de

tempo em que os dados foram coletados indica que os eventos maliciosos apresentados pelo Norse são bastante repetitivos.

Desses endereços obtidos pelo Norse, 223 também estão presentes no conjunto de dados da Honeynet, que é um pequeno valor considerando a grande quantidade de endereços IP que atacaram a Honeynet, mas não tão pequeno ao considerar-se a significativamente menor quantidade de endereços IP únicos do Norse.

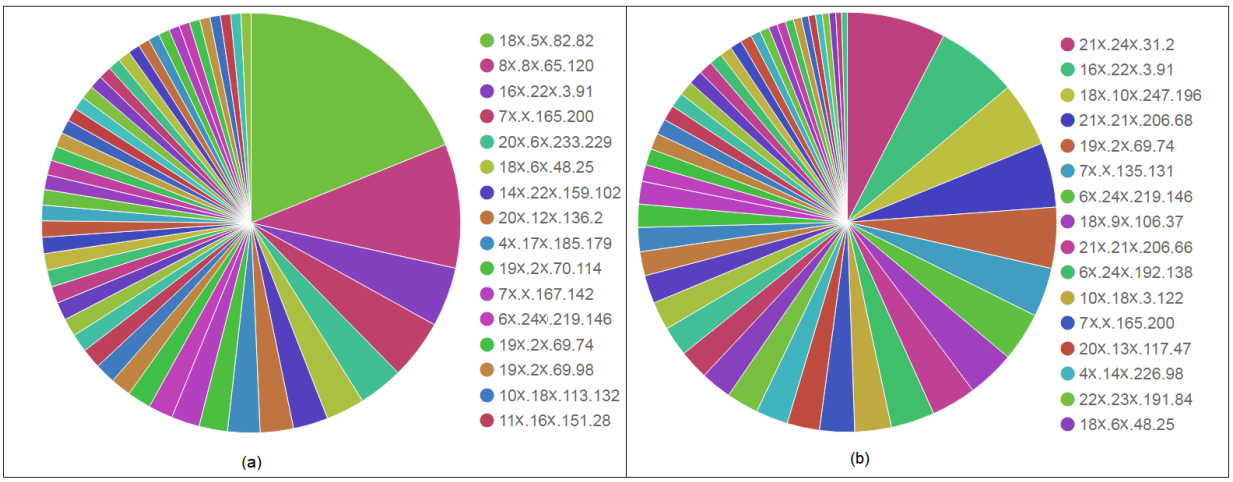

Figura 4.16: Distribuição dos endereços IP em comum no conjunto de dados do Norse (a) e da Honeynet (b).

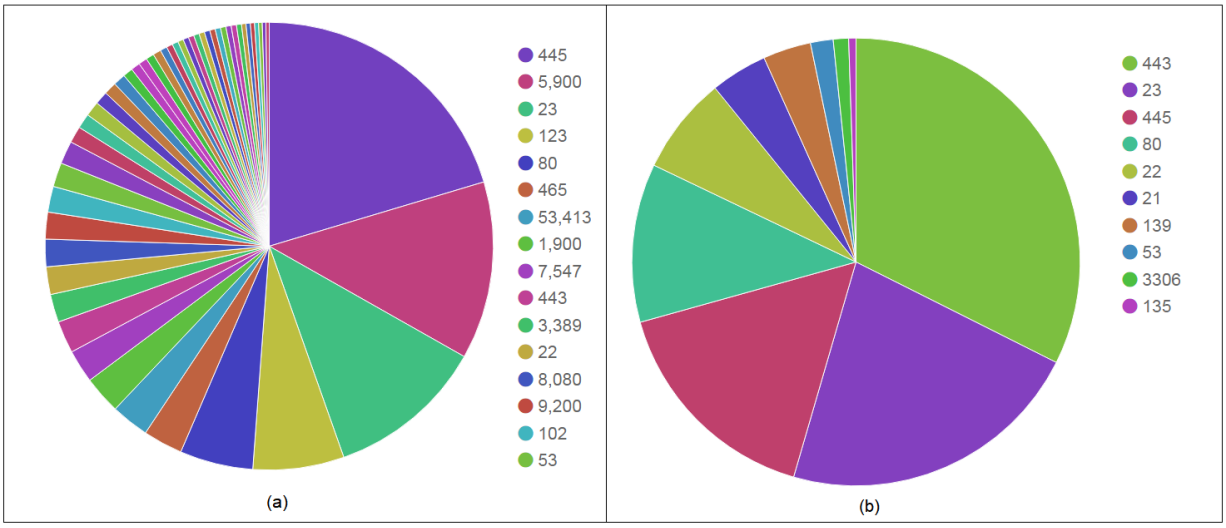

Figura 4.17: Distribuição das portas alvejadas pelos endereços IP em comum no conjunto de dados do Norse (a) e da Honeynet (b).

A Figura 4.16 mostra a contribuição desses endereços em comum no tráfego total em ambos os conjuntos de dados. Pela figura é possível notar que somente três endereços, dentre os endereços em comum, estão entre os mais frequentes nos dois conjuntos de dados, sendo eles 16x.22x.3.91, 19x.2x.69.74 e 18x.6x.48.25. A Figura 4.17 mostra as portas que esses endereços alvejaram. Os

endereços IP em comum são georreferenciados a 43 países diferentes, sendo a maioria aos Estados Unidos, de acordo com a Figura 4.18.

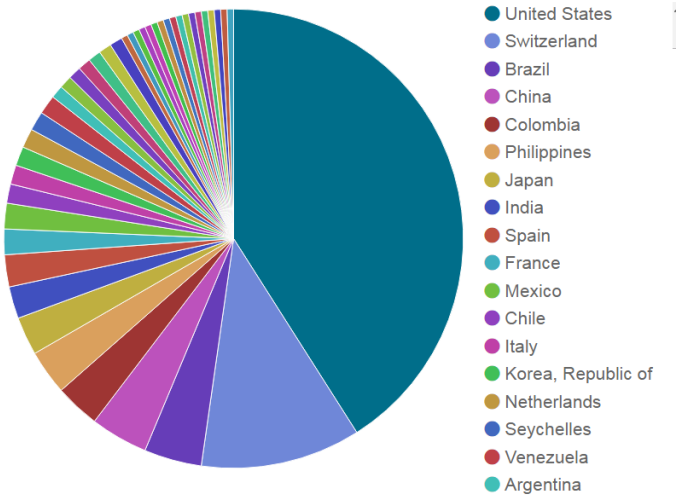

Figura 4.18: Países a que os endereços em comum são georreferenciados.

Dos 223 em comum, 209 conduziram ataques à mesma porta em ambos os conjuntos de dados, o que indica que efetuaram ataques semelhantes nos dois casos. Embora o Norse não forneça detalhes acerca do ataque, como o modus operandi dos atacantes ou identificação dos malwares, é provável que essas atividades correspondam àquelas descritas no Capítulo 4. A Figura 4.19 mostra quais foram as portas mais atacadas dentre esses endereços que praticaram ações similares.

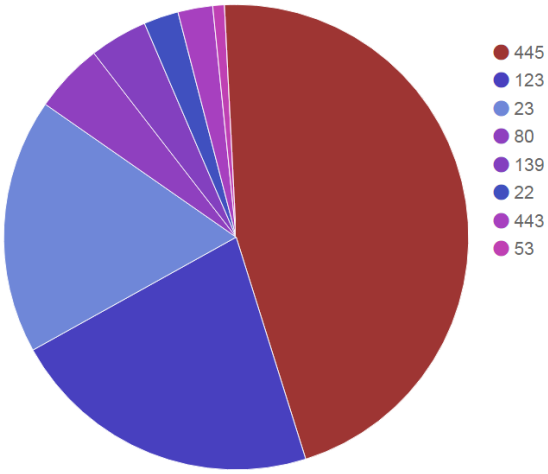

Figura 4.19: Portas alvejadas pelos atacantes em comum e que conduziram ataques similares em ambos os conjuntos de dados.

Dos endereços IP que efetuaram ataques NTP, uma parte expressiva dos atacantes que conduziram ataques similares em ambos os conjuntos de dados como mostra a Figura 4.19, a maioria refere-se aos endereços georreferenciados à Suíça, citados nas seções 4.1.1 e 4.4.6. Outras estatísticas relativas aos dados capturados pelos sensores do Norse são fornecidas no Anexo II.

Capítulo 5

Conclusão e Trabalhos Futuros

Este trabalho apresentou a eficiência do uso de inspeção de pacotes em forense de redes, gerando relatórios estatísticos das camadas de rede e transporte, que são úteis no reconhecimento de comportamento de atacantes, e identificação de anomalias em protocolos da camada de aplicação. Nas investigações em que a inspeção de pacotes não é eficaz, como a análise de tráfego criptografado, ambientes controlados e de pesquisa permitem a implementação de softwares que armazenam ações de usuários maliciosos na Honeypot, o que possibilita a análise, por exemplo, de ataques de força bruta em SSH. A análise mostra, também, que atividades maliciosas similares às presentes na Honeynet foram detectadas pelos sensores do Norse, algumas com o mesmo endereço IP de origem.

Entretanto, limitações na arquitetura proposta impedem a automação da investigação da rede. Portanto, trabalhos futuros podem incluir a implementação de uma arquitetura que automatiza a detecção de ataques, por exemplo com uso de Aprendizado de Máquina, com base nos resultados apresentados neste trabalho, automatizando a detecção dessas anomalias. Aprendizado de Máquina também pode ser usada para classificação de tráfego criptografado, por meio de análises estatísticas que, mesmo sem revelar o conteúdo do pacote, indicam potencial tráfego malicioso, que deve ser bloqueado ou investigado, permitindo a análise de tráfego criptografado na Honeynet, como HTTPS e SSH.

Trabalhos futuros também abrangem a expansão da análise para os demais serviços explorados, fornecendo informações sobre mais vulnerabilidades, e um estudo dos efeitos na memória dos Honeypots alvejados, com *memory dump*.

Bibliografia

ANDERSON, Ross et al. **The Economics of Information Security and Privacy**. 1. ed. Berlin: Springer, 2013. p. 265–300.

ARUKONDA, Srinivas; SINHA, Samta. The innocent perpetrators: reflectors and reflection attacks. **Advances in Computer Science: an International Journal**, v. 4, n. 1, p. 94–98, 2015.

BERRY, Alex; HOMAN, Josh; EITZMAN, Randi. **WannaCry Malware Profile**. 2017. Disponível em: <https://www.fireeye.com/blog/threat-research/2017/05/wannacry-malware-profile.html>. Acesso em: 27 out. 2017.

BUCZAK, Anna L; GUVEN, Erhan. A survey of data mining and machine learning methods for cyber security intrusion detection. **IEEE Communications Surveys & Tutorials**, IEEE, v. 18, n. 2, p. 1153–1176, 2016.

US-CERT. **Attack Possibilities by OSI Layer**. 2014. Disponível em: <https://www.us-cert.gov/sites/default/files/publications/DDoS%5C%20Quick%5C%20Guide.pdf>. Acesso em: 24 out. 2017.

CISCO. **Cisco Visual Networking Index: Forecast and Methodology, 2016–2021**. 2017. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>. Acesso em: 20 out. 2017.

COREY, Joseph. Advanced Honey Pot Identification And Exploitation. **Phrack Magazine**, 0x0b, 0x3f, 2003.

COUGHLIN, Tom. **WannaCry Ransomware Demonstrates The Value Of Better Security and Backups**. 2017. Disponível em: <https://www.forbes.com/sites/tomcoughlin/2017/05/14/wannacry-ransomware-demonstrations-the-value-of-better-security-and-backups/#252f282b70b8>. Acesso em: 24 out. 2017.

COWPERTHWAITE, Alex; SOMAYAJI, Anil. The futility of DNSSec. In: 5TH Annual Symposium Information Assurance, 16-17 Junho. Albany, NY, USA: [s.n.], 2010.

DEAN, Jeffrey; GHEMAWAT, Sanjay. MapReduce: simplified data processing on large clusters. **Communications of the ACM**, ACM, v. 51, n. 1, p. 107–113, 2008.

FLORES ARMAS, Denys; JHUMKA, Arshad. Implementing chain of custody requirements in database audit records for forensic purposes. In: IN Proceedings of the 16th International Conference on Trust, Security and Privacy in Computing and Communications, 01-04 Agosto. Sydney, NSW, Australia: IEEE, 2017.

GU, Guofei. **Correlation-based botnet detection in enterprise networks**. Atlanta, GA, USA: Georgia Institute of Technology, 2008.

ICAN-SSAC. **DNS Amplification Attacks**. 2006. Disponível em: <https://www.icann.org/en/system/files/files/dns-ddos-advisory-31mar06-en.pdf>. Acesso em: 25 nov. 2017.

KHAN, Suleman; ET AL. Network forensics: review, taxonomy, and open challenges. **Journal of Network and Computer Applications**, Elsevier, v. 66, p. 214–235, 2016.

KITCHIN, Rob; MCARDLE, Gavin. What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets. **Big Data & Society**, SAGE Publications Sage UK: London, England, v. 3, n. 1, p. 2053951716631130, 2016.

KOLIAS, Constantinos et al. Intrusion detection in 802.11 networks: empirical evaluation of threats and a public dataset. **IEEE Communications Surveys & Tutorials**, IEEE, v. 18, n. 1, p. 184–208, 2016.

KÜHRER, Marc et al. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In: USENIX Security Symposium, 20-22 Agosto, San Diego, CA, USA. [S.l.: s.n.], 2014. p. 111–125.

KUROSE, James F; ROSS, Keith W. **Computer Networking: A Top-Down Approach**. 6. ed. Boston, MA, USA: Addison-Wesley, 2013. p. 35–39.

MOKUBE, Iyatiti; ADAMS, Michele. Honeypots: concepts, approaches, and challenges. In: 45TH annual southeast regional conferencel, 23-24 Março. Winston-Salem, NC, USA: ACM, 2007. p. 321–326.

MUELLER, Milton. **DPI Technology from the standpoint of Internet governance studies: An introduction**. Syracuse, NY, USA: School of Information Studies, Syracuse University, Technical Report, 2011.

MUKKAMALA, S et al. Detection of virtual environments and low interaction honeypots. In: INFORMATION Assurance and Security Workshop, 20-22 Junho. West Point, NY, USA: IEEE, 2007. p. 92–98.

NAZARYAN, Gor. **W32.IRCBot**. Edição: Symantec. [S.l.: s.n.]. Disponível em: <https://www.symantec.com/security%5C_response/writeup.jsp?docid=2002-070818-0630-99>. Acesso em: 30 set. 2017.

NORSE. **Norse Attack Map**. 2017. Disponível em: <http://map.norsecorp.com/#/>. Acesso em: 25 nov. 2017.

OLIVEIRA JÚNIOR, Gildásio Antonio de; SOUSA JÚNIOR, Rafael Timóteo de; TENÓRIO, Danilo Fernandes. Desenvolvimento de um Ambiente Honeynet Virtual para Aplicação Governamental. In: 9TH International Conference on Forensic Computer Science, 23-25 Junho. Brasilia, DF, Brazil: [s.n.], 2015. p. 70–78.

PARSONS, Christopher. **Deep packet inspection in perspective: Tracing its lineage and surveillance potentials**. Kingston, Canadá: Queen's University, Surveillance Studies Centre, 2008.

PIMENTA RODRIGUES, Gabriel Arquelau et al. Cybersecurity and Network Forensics: Analysis of Malicious Traffic towards a Honeynet with Deep Packet Inspection. **Applied Sciences**, v. 7, n. 10, 2017. ISSN 2076-3417.

PRAYUDI, Yudi; SN, Azhari. Digital chain of custody: State of the art. **International Journal of Computer Applications**, v. 114, n. 5, 2015.

PROVOS, Niels; HOLZ, Thorsten. **Virtual Honeypots: From Botnet Tracking to Intrusion Detection**. 1. ed. Sebastopol, CA, USA: Addison-Wesley Professional, 2007.

RADWARE. **DDoS Attacks on DNS Services**. 2016. Disponível em: <https://security.radware.com/ddos-threats-attacks/threat-advisories-attack-reports/dns-services-under-attack/>. Acesso em: 28 out. 2017.

ROSSOW, Christian. Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In: SYMPOSIUM on Network and Distributed System Security, 23-26 Fevereiro. San Diego, CA, USA: [s.n.], 2014.

SAPUTRA, Ferry Astika; ET AL. Detecting and blocking onion router traffic using deep packet inspection. In: IN Proceedings of the International Electronics Symposium, 29-30 Setembro. Bali, Indonesia: IEEE, 2016. p. 283–288.

SPITZNER, Lance. The honeynet project: Trapping the hackers. **IEEE Security & Privacy**, IEEE, v. 99, n. 2, p. 15–23, 2003.

SYMANTEC. **Internet Security Threat Report**. 2017. Disponível em: <https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf>. Acesso em: 20 out. 2017.

TALABIS, Mark et al. **Information Security Analytics: Finding Security Insights, Patterns, and Anomalies in Big Data**. Waltham, MA, USA: Syngress, 2014. p. 1–12.

TEMPLETON, Steven J; LEVITT, Karl E. Detecting spoofed packets. In: IN Proceedings of the DARPA Information Survivability Conference and Exposition, 22-24 Abril. Washington, DC, USA: IEEE, 2003. v. 1, p. 164–175.

TRAPPE, Wade. The challenges facing physical layer security. **IEEE Communications Magazine**, IEEE, v. 53, n. 6, p. 16–20, 2015.

VAUGHN, Randal; EVRON, Gadi. **DNS Amplification Attacks**. 2006. Disponível em: <http://crt.io/DNS-Amplification-Attacks.pdf>. Acesso em: 25 nov. 2017.

VERMA, Abhilash. **Production Honeypots: An Organization's view**. 2003. Disponível em: <https://www.giac.org/paper/gsec/3585/production-honeypots-organizations-view/105831>. Acesso em: 22 out. 2017.

WARF, Barney. The Hermit Kingdom in cyberspace: unveiling the North Korean internet. **Information, Communication & Society**, Taylor & Francis, v. 18, n. 1, p. 109–120, 2015.

WHITE, Tom. **Hadoop: The definitive guide**. Sebastopol, CA, USA: "O'Reilly Media, Inc.", 2012.

XU, Chengcheng et al. A Survey on Regular Expression Matching for Deep Packet Inspection: Applications, Algorithms, and Hardware Platforms. **IEEE Communications Surveys & Tutorials**, IEEE, v. 18, n. 4, p. 2991–3029, 2016.

YU, Jaegwan et al. A framework for detecting MAC and IP spoofing attacks with network characteristics. In: INTERNATIONAL Conference on Software Security and Assurance, 24-25 Julho. Altoona, PA, USA: IEEE, 2016. p. 49–53.

YUAN, Xingliang; ET AL. Privacy-preserving deep packet inspection in outsourced middleboxes. In: IN Proceedings of the International Conference on Computer Communications, 10-15 Abril. San Francisco, CA, USA: IEEE, 2016. p. 1–9.

ZOU, Yulong et al. Improving physical-layer security in wireless communications using diversity techniques. **IEEE Network**, IEEE, v. 29, n. 1, p. 42–48, 2015.

ANEXOS

ANEXO I

Códigos Fonte Utilizados

Este Anexo apresenta os Códigos Fonte utilizados para desenvolvimento deste trabalho.

I.1 Conversão e Indexação dos Dados

Para converter os arquivos PCAP coletados da Honeynet, usa-se o comando TShark `tshark -T fields -n -r ./arquivo.pcap -E separator=} -E header=y -e field1 -e field2 ... -e fieldN> arquivo.csv`, em que `fieldi` é o i-ésimo campo analisado do tráfego. Todos os campos estudados são apresentados no código de PIPELINE.CONF, que é usado para indexação dos dados no ElasticSearch.

```
1  PIPELINE.CONF
2
3  # Arquivo a ser indexado e o CSV covnertido por tshark
4  input{
5     file{
6       path => "C:\ELK\logstash\arquivo.csv"
7       start_position => "beginning"
8       ignore_older => 0
9     }
10 }
11
12 # Usa-se filtro CSV para processar os dados
13
14 filter{
15   csv {
16   source => "message"
17
18   # Separador dos valores do arquivo
19         separator => "}"
20
21   # Campos presentes no arquivo
22         columns => [
23     # Camada 2
24   "hn_frame_protocols",
25     "hn_frame_time_epoch",
26   # Camada 3
27   "hn_ip_dst",
28   "hn_ip_geoip_dst_country",
```

```
    "hn_ip_geoip_dst_asnum",
    "hn_ip_geoip_dst_city",
    "hn_ip_src",
    "hn_ip_geoip_src_country",
    "hn_ip_geoip_src_asnum",
    "hn_ip_geoip_src_city",
    # Camada 4
    "hn_tcp_srcport",
    "hn_tcp_dstport",
    "hn_tcp_flags",
    "hn_udp_srcport",
    "hn_udp_dstport",
    # Camada 5
    "hn_dns_count_queries",
    "hn_dns_count_answers",
    "hn_dns_qry_name",
    "hn_dns_qry_type",
    "hn_dns_qry_class",
    "hn_dns_resp_name",
    "hn_dns_resp_type",
    "hn_dns_resp_class",
    "hn_ftp_request_command",
    "hn_ftp_request_arg",
    "hn_ftp_response_code",
    "hn_ftp_response_arg",
    "hn_http_request_method",
    "hn_http_request_uri",
    "hn_http_host",
    "hn_http_user_agent",
    "hn_http_request_full_uri",
    "hn_http_response_code",
    "hn_http_response_phrase",
    "hn_http_date",
    "hn_http_server",
    "hn_http_last_modified"
    "hn_mysql_error_message",
    "hn_nbss_called_name",
    "hn_nbss_calling_name",
    "hn_nbss_type",
    "hn_nbss_error_code",
    "hn_ntlmssp_auth_username",
    "hn_ntlmssp_auth_hostname",
    "hn_ntp_priv_reqcode",
    "hn_ntp_xmt",
    "hn_smb_server_component",
    "hn_smb_cmd",
    "hn_smb_nt_status",
    "hn_smb_flags",
    "hn_smb_flags2",
    "hn_smb_pid_high",
    "hn_smb_signature",
    "hn_smb_wct",
    "hn_smb_pwlen",
    "hn_smb_bcc",
    "hn_smb_password",
    "hn_smb_path",
    "hn_smb_service",
    "hn_smb_access_mask",
    "hn_smb_connect_support",
    "hn_smb_file",
```

```
    "hn_smb_share_access",
    "hn_smb_impersonation_level",
    "hn_smb_file_name_len",
    "hn_smb_file_attribute",
    "hn_smb_security_flags",
    "hn_smb_native_fs",
    "hn_smb_security_blob",
    "hn_smb_native_lanman",
    "hn_smb_native_os",
    "hn_telnet_cmd",
    "hn_telnet_subcmd",
    "hn_telnet_data"]

    skip_empty_columns => "true"
      }
}

# Saida dos dados resultantes
output{

   # Printar na tela
   stdout{
     codec => rubydebug
   }

   # Indexar no ElasticSearch
   elasticsearch {
       hosts => [ "localhost:9200" ]
   }
}
```

I.2 Georreferenciamento de endereços IP

Para adequada escala do tamanho dos círculos referentes ao ataque em função da quantidade de pacotes transmitidos, usa-se o comando Linux `sort tsharkOutput.csv | uniq -c > uniqueIP.csv`. Cada endereço IP desse arquivo é, então, georreferenciado usando o código Python dado em GEOREF.PY.

GEOREF.PY

```
#!/usr/bin/env python

# Biblioteca para georreferenciamento IP
import pygeoip

# Local do arquivo de database para GeoRef
geoIP = pygeoip.GeoIP('GeoLiteCity.dat')
output = open('fileGeoRef.csv', 'w')

# Abre e le linhas do arquivo de entrada
with open('uniqueIP.csv') as fileIP:
    fileContent = fileIP.readlines()
fileContent = [x.strip('\n') for x in fileContent]

# Varre todos IPs do arquivo
for i in xrange(0, len(fileContent)):
```

```
    try:
        # Quantidade de pacotes enviados por esse IP
        amountPackets = fileContent[i].split(' ')[0]
        packets = fileContent[i].split(' ')[1]

        # Le IP e faz GeoRef
        ip = packets.split(',')[0]
        ipInfo = geoIP.record_by_addr(ip)

        # Le lat e lon retornado
        ipInfo = str(ipInfo['latitude']) + ',' + str(ipInfo['longitude']) + ','
        ipInfo += packets.split(',')[1] + ',' + amountPackets

        # Escreve no arquivo e limpa variavel
        output.write("%s\n" % ipInfo)
        ipInfo = ''
    except (TypeError, IndexError):
        continue

output.close()
```

I.3 Aplicação de Mapa Estático

Os códigos HTML e JavaScript, usando a biblioteca p5.js, dessa aplicação são apresentados em INDEX.HTML e SKETCH.JS, respectivamente. Esses códigos foram adaptados de outro autor, e os créditos são dados no código.

INDEX.HTML

```
<!-- Adaptado de https://github.com/CodingTrain/Rainbow-Code/tree/master/
    CodingChallenges/CC_57_Earthquake_Viz -->

<html>
<head>
  <meta charset="UTF-8">
  <title>Honeynet Viz Estatico</title>
  <script language="javascript" type="text/javascript" src="libraries/p5.js"></script>
  <script language="javascript" src="libraries/p5.dom.js"></script>
  <script language="javascript" type="text/javascript" src="sketch.js"></script>
</head>

<body>
</body>
</html>
```

SKETCH.JS

```
// Adaptado de https://github.com/CodingTrain/Rainbow-Code/tree/master/CodingChallenges/
    CC_57_Earthquake_Viz

var mapImage;
var centralLat = 0;
var centralLon = 0;
var canvasWidth = 1024;
var canvasHeight = 600;
var zoomLevel = 1;
```

```
11 var ports = [];
12 var colors = [];
13 var type = 'dark'; // pode ser: basic, streets, bright, light, dark, satellite
14
15 function preload() {
16
17   // Chave da API do Mapbox
18   var APIkey = '<chave-vai-aqui>'
19
20   // Carrega a imagem estatica do mapa
21   mapImage = loadImage('https://api.mapbox.com/styles/v1/mapbox/' + type + '-v9/static/'
         +
22     centralLon + ',' + centralLat + ',' + zoomLevel + '/' + canvasWidth + 'x' +
          canvasHeight + '?access_token=' + APIkey);
23
24   // Carrega arquivo que contem dados dos ataques
25   attacks = loadTable('arquivoAtaques.csv', 'csv', 'header');
26 }
27
28 // Converte Latitude e Longitude em localizacao pixel XY
29 function getXYFromLatLon(lat, lon){
30   lat = radians(lat);
31   lon = radians(lon);
32
33   var x = ((256 / PI) * pow(2, zoomLevel))*(lon + PI);
34   var y = ((256 / PI) * pow(2, zoomLevel))*(PI - log(tan(PI / 4 + lat / 2)));
35
36   return [x,y];
37 }
38
39 // Retorna a quantidade de pacotes enviados pelo atacante que mais enviou pacotes
40 function getMaxPackets(){
41   var maxPackets = 0;
42
43   for (var i = 0; i < attacks.getRowCount(); i++){
44     if (parseInt(attacks.getString(i, 'numPackets')) > maxPackets)
45       maxPackets = parseInt(attacks.getString(i, 'numPackets'));
46   }
47
48   return maxPackets;
49 }
50
51 // Adiciona informacao na legenda
52 function setLegend(port, red, green, blue){
53   if (!ports.includes(port) && !isNaN(port)){
54     ports.push(port);
55     colors.push(red, green, blue);
56   }
57 }
58
59 // Desenha a legenda
60 function drawLegend(){
61   var legX = -500;
62   var legY = 200;
63   var textColor = type.localeCompare("dark") ? 0 : 255;
64   textSize(12);
65   noStroke();
66
67   for (var i = 0; i < ports.length; i++){
68    fill(colors[i*3], colors[i*3+1], colors[i*3+2], 200);
```

```
69      ellipse(legX, legY, 10, 10);
70      fill(textColor);
71      legX += 5;
72      legY += 5;
73      text(ports[i], legX, legY);
74      legX += 40;
75      legY -= 5;
76    }
77  }
78
79  function setup() {
80
81    // Cria a canvas com a imagem estatica do mapa no fundo
82    createCanvas(canvasWidth, canvasHeight);
83    translate(width / 2, height / 2);
84    imageMode(CENTER);
85    image(mapImage, 0, 0);
86
87    // Converte as coordenadas LatLon para pixel XY
88    var [centralX, centralY] = getXYFromLatLon(centralLat,centralLon);
89
90    // Maior quantidade de pacotes transmitidas por um unico atacante (para escala
          apropriada dos circulos)
91    var maxPackets = getMaxPackets();
92
93    // Desenha o circulo de cada ataque
94    for (var i = 0; i < attacks.getRowCount(); i++){
95      var latitude = parseFloat(attacks.getString(i, 'latitude'));
96      var longitude = parseFloat(attacks.getString(i, 'longitude'));
97      var port = parseInt(attacks.getString(i, 'port'));
98      var numPackets = parseInt(attacks.getString(i, 'numPackets'));
99
100     // Converte para LatLon para coordenada de pixel XY
101     var [atk_x,atk_y] = getXYFromLatLon(latitude,longitude);
102
103     // As coordenadas obtidas sao em relacao ao centro
104     atk_x -= centralX;
105     atk_y -= centralY;
106
107     // operacao com numeros arbitrarios para converter a porta atacada para cor, mas de
            forma que numeros proximos nao traduzam para cores semelhantes
108      var red = (port * port * 14543 + 1202) % 256;
109      var green = (port * port * 7654 + 351) % 256;
110      var blue = (port * port * 9999 - 0199) % 256;
111
112     // Adiciona a informacao da porta e cor na legenda
113     setLegend (port, red, green, blue);
114
115      // Determinacao do tamanho do circulo de acordo com numero de pacotes enviados
116      numPackets = sqrt(numPackets);  // Area do circulo cresce em funcao quadratica (A =
             pi*r^2). Essa linha anula esse crescimento
117      var diameter = map(numPackets, 0, maxPackets, 0, 15000);
118
119     // Cor do circulo de acordo com ataque
120     stroke(red,green,blue);
121     fill(red, green, blue, 200);
122
123      //Desenha o circulo
124      ellipse(atk_x, atk_y, diameter, diameter);
125    }
```

```
126
127     // Por fim, desenha a legenda
128     drawLegend();
129   }
```

I.4 Aplicação de Mapa Dinâmico

Os códigos HTML e JavaScript, usando a biblioteca p5.js, dessa aplicação são apresentados em INDEX.HTML e SCRIPT.JS, respectivamente. Esses códigos foram adaptados de outro autor, e os créditos são dados no código.

```
1  INDEX.HTML
2
3  <!-- Adaptado de https://github.com/cvalenzuela/Mappa/tree/master/examples/tile/Mapbox
       -->
4  <!DOCTYPE html>
5  <html lang="en">
6  <head>
7    <meta charset="UTF-8">
8    <meta name="viewport" content="width=device-width, initial-scale=1.0">
9    <meta http-equiv="X-UA-Compatible" content="ie=edge">
10   <title>Honeynet Viz Dinamico</title>
11   <script src="Caminho/ate/libraries/p5.js"></script>
12   <script src="https://unpkg.com/mappa-mundi/dist/mappa.min.js" type="text/javascript"><
         /script>
13 </head>
14
15 <body>
16   <script src="script.js"></script>
17 </body>
18 </html>
```

```
1  SCRIPT.JS
2
3  // Adaptado de https://github.com/cvalenzuela/Mappa/tree/master/examples/tile/Mapbox
4
5  // Chave da API do Mapbox
6  var APIkey = '<chave-vai-aqui>'
7
8  // Opcoes default do mapa
9  var defaultOptions = {
10   lat: -15.7941,  // Geolocalizacao de Brasilia - DF
11   lng: -47.8825,
12   zoom: 10,
13   studio: true,
14   style: 'mapbox://styles/mapbox/traffic-night-v2'
15 }
16
17 // Instancia de Mapbox
18 var mappa = new Mappa('Mapbox', APIkey);
19 var attackMap;
20 var canvas;
21 var attacks;
22
23 function setup() {
24   var height = 600;
```

```
25    var width = 1024;
26    canvas = createCanvas(width, height);
27
28    // Cria mapa
29    attackMap = mappa.tileMap(defaultOptions);
30    attackMap.overlay(canvas);
31
32    // Carrega arquivo que contem dados dos ataques
33    attacks = loadTable('arquivoAtaques.csv', 'csv', 'header');
34
35    // Recarrega os ataques quando o mapa mudar (pelo zoom, centro)
36    attackMap.onChange(drawAttacks);
37  }
38
39  // Retorna a quantidade de pacotes enviadas pelo atacante que mais enviou pacotes
40  function getMaxPackets(){
41    var maxPackets = 0;
42
43    for (var i = 0; i < attacks.getRowCount(); i++){
44      if (parseInt(attacks.getString(i, 'numPackets')) > maxPackets)
45        maxPackets = parseInt(attacks.getString(i, 'numPackets'));
46    }
47
48    return maxPackets;
49  }
50
51  // Desenha os circulos referentes aos ataques no mapa
52  function drawAttacks() {
53    clear();
54
55    // Retorna numero de pacotes enviados pelo mais frequente atacante
56    // Para uso na escala do tamanho dos circulos de ataque
57    var maxPackets = getMaxPackets();
58
59    for (var i = 0; i < attacks.getRowCount(); i++) {
60      var latitude = parseFloat(attacks.getString(i, 'latitude'));
61      var longitude = parseFloat(attacks.getString(i, 'longitude'));
62
63      // S desenha ataques visiveis na tela, para melhor desempenho
64      if (attackMap.map.getBounds().contains([latitude, longitude])) {
65
66        // Converte latitude e longitude para localizacao em pixel XY
67        var atkPos = attackMap.latLngToPixel(latitude, longitude);
68
69        // Porta atacada por esse atacante
70        var port = parseInt(attacks.getString(i, 'port'));
71
72        // operacao com numeros arbitrarios para converter a porta atacada para cor, mas
              de forma que numeros proximos nao traduzam para cores semelhantes
73        var red = (port * port * 14543 + 1202) % 256;
74        var green = (port * port * 7654 + 351) % 256;
75        var blue = (port * port * 9999 - 0199) % 256;
76
77        // Numero de pacotes enviados pelo i-esimo atacante e remapeia pra escala
              apropiada
78        var numPackets = attacks.getString(i, 'numPackets');
79        var diameter = map(numPackets, 0, maxPackets, 0, 40) + attackMap.zoom();
80
81        // Desenha o circulo na posicao do atacante e de tamanho de acordo com quantidade
82        // de pacotes enviados e cor de acordo com a porta atacada
```

```
        stroke(red,green,blue);
        fill(red, green, blue, 200);
        ellipse(atkPos.x, atkPos.y, diameter, diameter);
      }
    }
  }
   }
```

I.5 Norse Scraper

Para coletar os dados apresentados pelo mapa do Norse, usou-se a biblioteca BeautifulSoup para Python, que axulia em atividades de Web Scraping.

```
NORSEBS.PY

#!/usr/bin/env python

import time
import schedule
import datetime
from selenium import webdriver
from bs4 import BeautifulSoup

i = 0
region = 'global'       # Regiao padrao e global

# Funcao para mudanca dos filtros de geolocalizacao
def changeFilter():
    global i
    global region
    i = (i+1)%len(geoFilters)                       # Muda o filtro de acordo com o
        vetor que armazena os possiveis filtros
    newUrl = url + geoFilters[i]                    # Aplica o novo filtro na URL
    driver.get(newUrl)
    driver.get(newUrl)                              # Buscar a URL somente uma vez nao
        funciona
    region = geoFilters[i][8:]                      # O nome da regiao aparece a partir
        do 8o caracter da URL
    if region == '':
        region = 'global'

data = str(datetime.date.today()) + ", "            # String que aramazenara as
    informacoes de cada ataque comeca armazenando o dia do evento
information = []                                    # Lista que armazenara informacoes
    de 10 ataques simultaneamente
url = 'http://map.norsecorp.com'                    # URL do site Norse
driver = webdriver.Chrome('C:\Users\Gabriel\Documents\Python\chromedriver.exe')
                                # Caminho para o Driver do Chrome
geoFilters = ['', '/#/?geo=latAmer', '/#/?geo=seAsia', '/#/?geo=eu', '/#/?geo=westAsia',
     '/#/?geo=usChina']     # Possiveis regioes a serem filtradas
driver.get(url)

    # Vai para o site do Norse
output = open("norse.csv", "a")
schedule.every(60).minutes.do(changeFilter)         # A cada 60 minutos troca a regiao
    sendo filtrada

```

```
35 while True:
36     time.sleep(3)                                    # Espera 3 segundos para novos
           ataques carregarem
37     htmlSource = driver.page_source                  # Conteudo HTML da pagina
38
39     soup = BeautifulSoup(htmlSource, "lxml")         # BeautifulSoup faz o parse do HTML
40     attacks = soup.find_all("tr", {"class": "row ng-scope"})        # Os ataques
           aparecem sob a tag <tr> e classe: "row ng-scope"
41     attacks = attacks[-10:]                          # Os ataques correspondem as 10
           ultimas ocorrencias da tag <tr>
42     attacks = attacks[::-1]                          # Os coloca em ordem cronologica (no
            HTML, a ordem e invertida)
43
44     for attack in attacks:
45         attackFields = attack.find_all("td", {"class": "cell"})     # Acha todas as
               informacoes de cada ataque (cada ataque tem 6 campos)
46         data = str(datetime.date.today()) + ", "                    # Reseta 'data'
47         for x in xrange(0, 7):
48             data = data + attackFields[x].text + ", "               # Concatena os
                   campos sobre o ataque, os separando por virgulas
49         if str(attackFields[0].text) != '':                         # Norse tem um bug
               de nao mostrar ataques, entao so escreve no arquivo se capturou dados
50             data = data + region                                    # O ultimo dado
                   sobre o ataque e sua regiao
51             information.append(data)                                # Coloca os dados
                   sobre o ataque na lista
52         else:                                                       # Se o bug de nao
               mostrar dados aconteceu,
53             driver.refresh()                                        # Atualiza a pagina
                   e tenta novamente
54             time.sleep(7)                                           # Espera 7 segundos
                   para pagina carregar e ataques aparecerem
55             break
56
57
58     for info in information:
59         output.write("%s\n" % info)
60     del information[:]                                              # Deleta informacoes
61     schedule.run_pending()                                          # Executa '
           changeFilter' caso pendente (cada 60 minutos)
62
63 output.close()
```

ANEXO II

Estatísticas dos Dados do Norse

Este anexo apresenta estatísticas relativas às atividades maliciosas detectadas pelos sensores do Norse. As seções a seguir focalizam essas estatísticas em cada região disponibilizada nos filtros do site e, na última seção, apresenta-se estatísticas que abrangem todas regiões.

II.1 Global

A Figura II.1 mostra as estatísticas dos dados capturados com o filtro global.

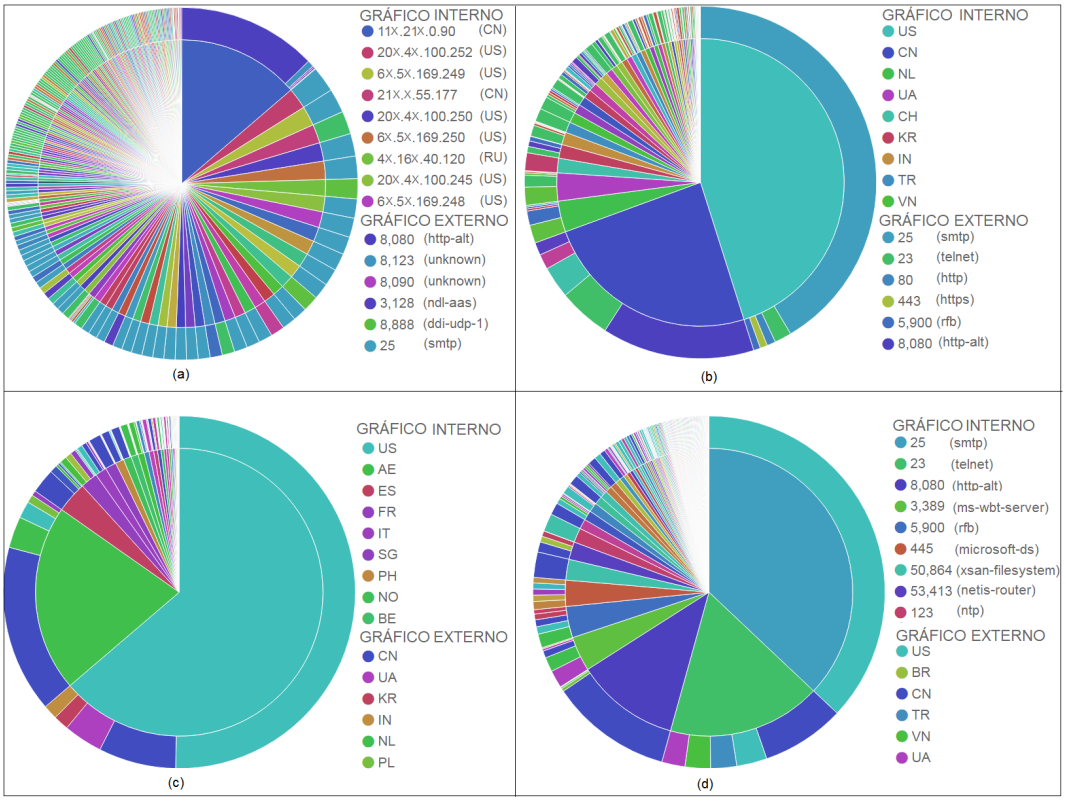

Figura II.1: Estatísticas do filtro "Global". IP atacante e porta alvejada (a) país de origem e porta alvejada (b) país de origem e país alvo (c) e porta alvejada e país de origem (d).

II.2 América Larina

A Figura II.2 apresenta as estatísticas focando os dados na América Latina.

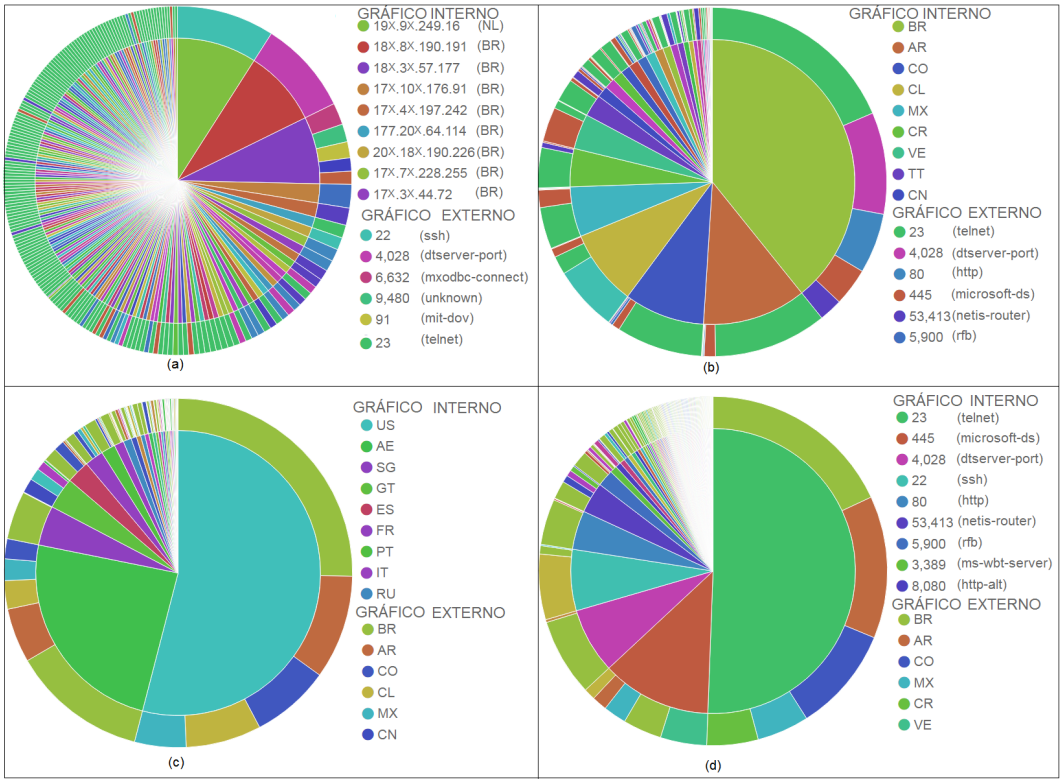

Figura II.2: Estatísticas do filtro "América Latina". IP atacante e porta alvejada (a) país de origem e porta alvejada (b) país de origem e país alvo (c) e porta alvejada e país de origem (d).

II.3 Sudeste Asiático

As estatísticas focando no Sudeste Asiático são apresentados na Figura II.3.

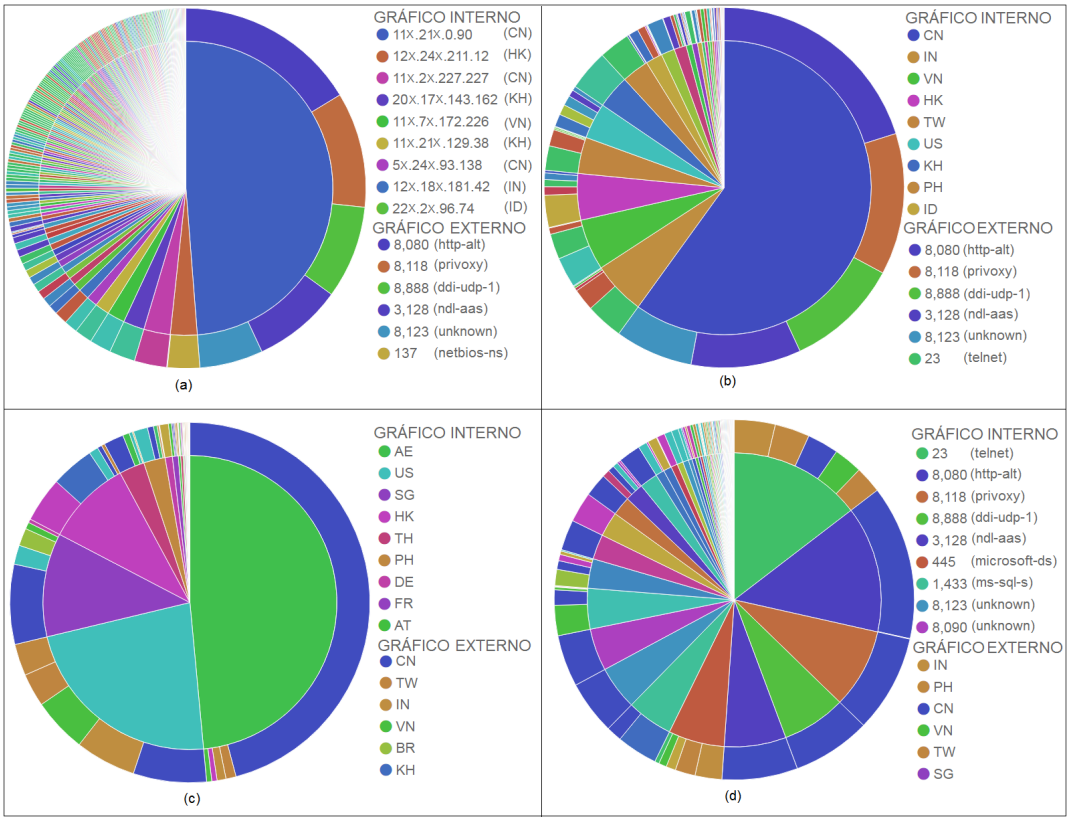

Figura II.3: Estatísticas do filtro "América Latina". IP atacante e porta alvejada (a) país de origem e porta alvejada (b) país de origem e país alvo (c) e porta alvejada e país de origem (d).

II.4 Europa

Filtrando os dados na região europeia, obtém-se as estatísticas mostradas na Figura II.4.

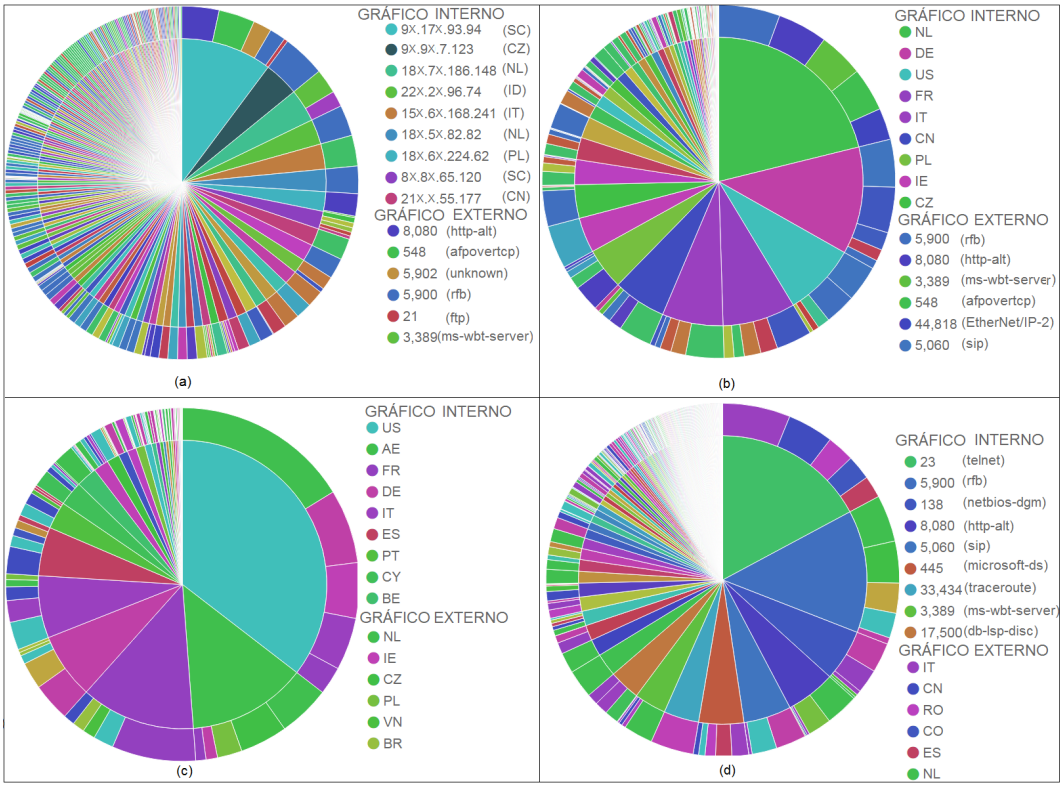

Figura II.4: Estatísticas do filtro "Europa". IP atacante e porta alvejada (a) país de origem e porta alvejada (b) país de origem e país alvo (c) e porta alvejada e país de origem (d).

II.5 Oeste Asiático

O filtro da região oeste da Ásia retorna dados cujas estatísticas são dadas na Figura II.5.

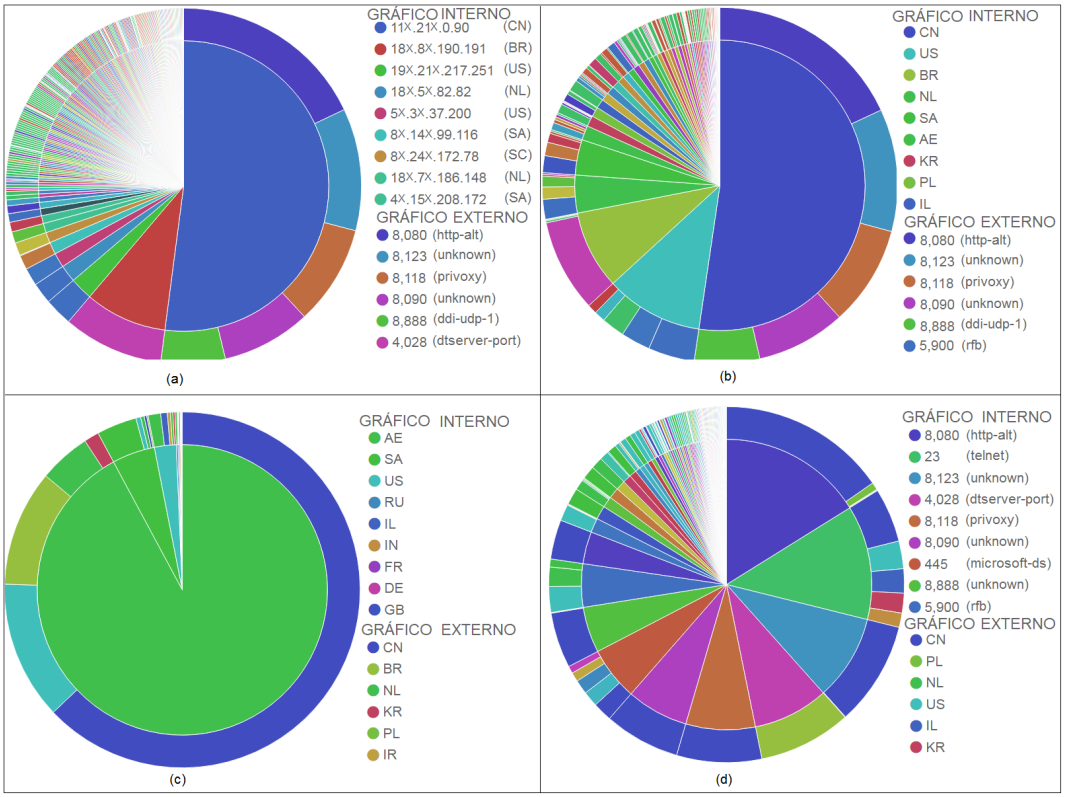

Figura II.5: Estatísticas do filtro "Oeste Asiático". IP atacante e porta alvejada (a) país de origem e porta alvejada (b) país de origem e país alvo (c) e porta alvejada e país de origem (d).

II.6 Estados Unidos & China

Ao limitar a análise somente nesses dois países, as estatísticas mostradas na Figura II.6 são obtidas.

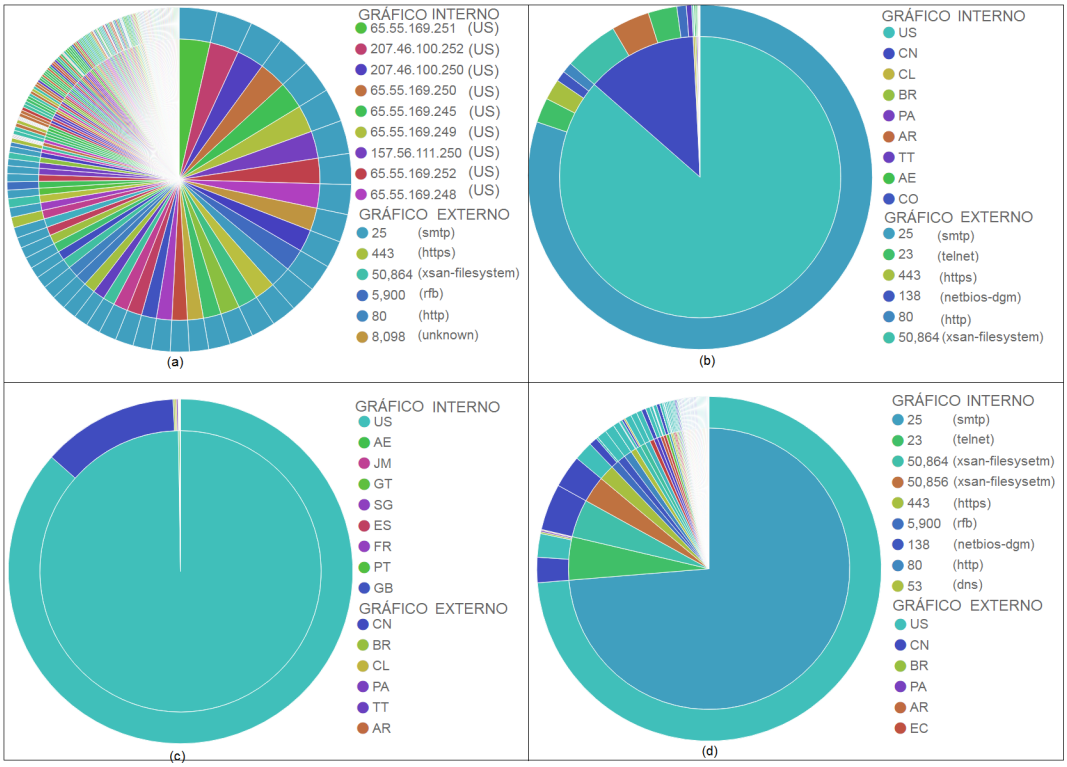

Figura II.6: Estatísticas do filtro "Estados Unidos & China". IP atacante e porta alvejada (a) país de origem e porta alvejada (b) país de origem e país alvo (c) e porta alvejada e país de origem (d).

II.7 Todas Regiões

Embora o Norse forneça a opção global de visualização de dados, cujas estatísticas são apresentadas na seção II.1, as estatísticas dos dados incluindo todas regiões, conforme a Figura II.7, mostrou-se diferente.

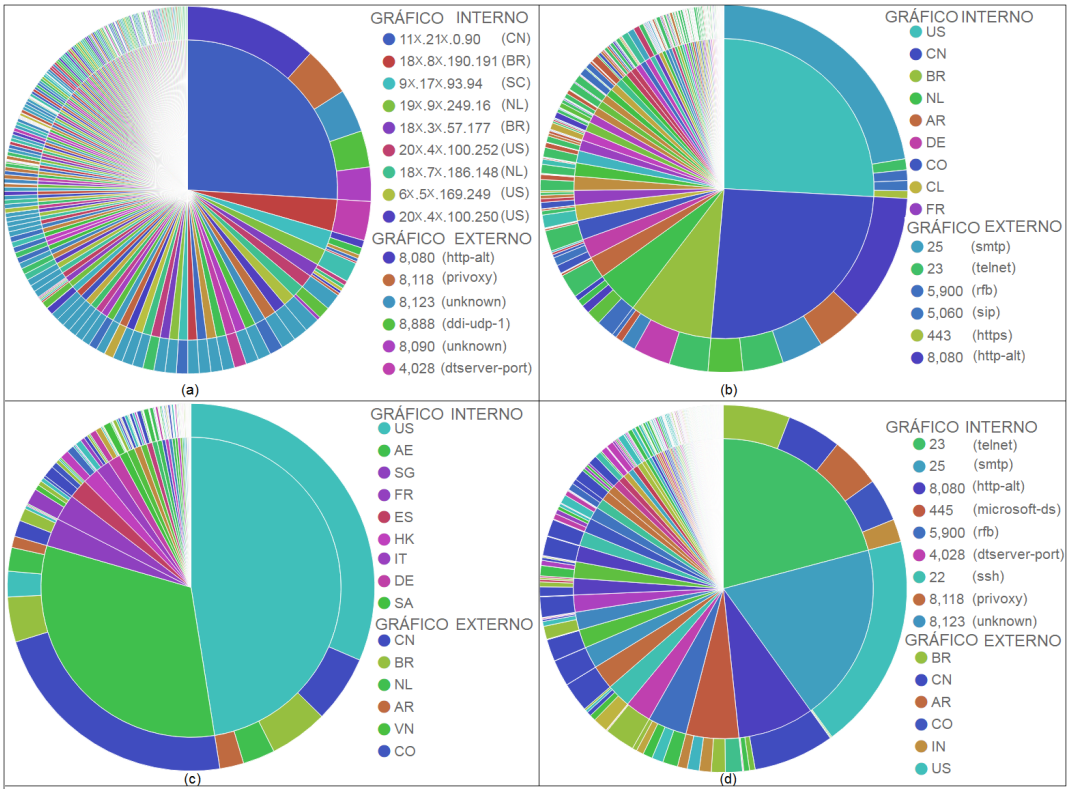

Figura II.7: Estatísticas do filtro "Estados Unidos & China". IP atacante e porta alvejada (a) país de origem e porta alvejada (b) país de origem e país alvo (c) e porta alvejada e país de origem (d).