UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

IVAN ROLIM SZESZ DE OLIVEIRA

LEONARDO PAES GONÇALVES

AVALIAÇÃO DE HONEYPOTS NA DETECÇÃO DE TRÁFEGO MALICIOSO EM REDES DE COMPUTADORES

PONTA GROSSA

2014

IVAN ROLIM SZESZ DE OLIVEIRA

LEONARDO PAES GONÇALVES

AVALIAÇÃO DE HONEYPOTS NA DETECÇÃO DE TRÁFEGO MALICIOSO EM REDES DE COMPUTADORES

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas da Coordenação de Análise e Desenvolvimento de Sistemas – COADS - da Universidade Tecnológica Federal do Paraná.

Orientadora: Profª. Msc. Thalita Scharr Rodrigues

PONTA GROSSA

2014

TERMO DE APROVAÇÃO

AVALIAÇÃO DE HONEYPOTS NA DETECÇÃO DE TRÁFEGO MALICIOSO EM REDES DE COMPUTADORES

por

IVAN ROLIM SZESZ DE OLIVEIRA
LEONARDO PAES GONÇALVES

Este (a) Trabalho de Conclusão de Curso (TCC) foi apresentado(a) em 24 de fevereiro de 2014 como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas. Os(a) candidatos(a) foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

_____
Thalita Scharr Rodrigues Pimenta
Prof.(a) Orientador(a)

_____
Mauren Louise Sguario Coelho de Andrade
Membro titular

_____
Tânia Lúcia Monteiro
Membro titular

- O Termo de Aprovação assinado encontra-se na Coordenação do Curso -

AGRADECIMENTOS

Ivan Rolim Szesz de Oliveira

Agradeço a Deus pela vida, saúde e força para superar as dificuldades enfrentadas ao longo dessa caminhada.

Agradeço a professora Thalita Scharr Rodrigues que, mesmo em situações difíceis, dedicou toda a atenção necessária para a realização desse trabalho. Graças a seu conhecimento, dedicação e profissionalismo foi possível transformar uma simples ideia em um trabalho de conclusão de curso.

Agradeço aos professores que contribuíram, tanto com conhecimento técnico quanto com sabedoria, para minha formação acadêmica.

Agradeço aos meus amigos Daniel Vaz e Márcio Barbato pelo auxílio fornecido durante a elaboração deste trabalho.

Enfim, agradeço a todos aqueles que contribuíram, de forma consciente ou não, para a realização do meu objetivo.

AGRADECIMENTOS

Leonardo Paes Gonçalves

Agradeço a Deus pela vida, saúde e força para superar as dificuldades enfrentadas ao longo dessa caminhada.

Agradeço a professora Thalita Scharr Rodrigues que, mesmo em situações difíceis, dedicou toda a atenção necessária para a realização desse trabalho. Graças a seu conhecimento, dedicação e profissionalismo foi possível transformar uma simples ideia em um trabalho de conclusão de curso.

Agradeço a minha namorada pelo amor e carinho. A minha família pelo apoio recebido ao longo dessa jornada.

Agradeço aos professores que contribuíram, tanto com conhecimento técnico quanto com sabedoria, para minha formação acadêmica.

Agradeço aos meus amigos Daniel Vaz, Larissa Benck e Márcio Barbato pelo auxílio inestimável durante a minha trajetória na UTFPR- PG.

Enfim, agradeço a todos aqueles que contribuíram, de forma consciente ou não, para a realização do meu objetivo.

RESUMO

DE OLIVEIRA, Ivan Rolim Szesz; GONÇALVES, Leonardo Paes. **Avaliação de honeypots na detecção de tráfego malicioso em redes de computadores**. 2014. 67 f. Trabalho de Conclusão de Curso - Curso Superior em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

Este trabalho tem por finalidade avaliar ferramentas relacionadas à tecnologia de honeypots, com o intuito de detectar o tráfego malicioso na rede de computadores, utilizando-se de uma abordagem de redes, de forma a garantir a sua segurança. A atenção é voltada para o funcionamento das ferramentas utilizadas neste trabalho. Por meio de experimentação em um ambiente virtual, ou seja, com a utilização de máquinas virtuais, os dados são coletados e analisados posteriormente pelo conjunto de ferramentas escolhidas e seus respectivos scripts de log, a fim de determinar as características de possíveis ataques. As conclusões extraídas destas análises, durante o desenvolvimento dos diferentes ambientes, também são descritas. Este trabalho servirá para outros trabalhos relacionados ao monitoramento de atividades maliciosas em redes de computadores.

Palavras-chave: Honeypots. Segurança de redes. Worm. Máquina Virtual. Tráfego malicioso.

ABSTRACT

DE OLIVEIRA, Ivan Rolim Szesz; GONÇALVES, Leonardo Paes. **Evaluation of honeypots at detecting malicious traffic in computer networks. 2014. 67 f.** Trabalho de Conclusão de Curso - Curso Superior em Análise e Desenvolvimento de Sistemas - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2014.

The main goal of this work is to evaluate the tools related to honeypots technology, in order to detect malicious traffic on the computer networks in order to ensure security even from worm attacks. The attention is focused on the operation of the tools used in this work. By means of experimentation in a virtual environment, or with the use of virtual machines, data are collected and later analyzed by the selected set of tools and their respective log in order to determine the characteristics of certain attacks. The conclusions extracted from these analyzes during the development of the different environments are also described. This work will be useful for other works related to monitoring malicious activities in computer networks.

Keywords: Honeypots. Network security. Worm. Virtual Machine. Malicious traffic.

LISTA DE FIGURAS

LISTA DE SIGLAS

FTP	File Transfer Protocol
HTML	*HyperText Markup Language*
IDS	Intrusion detection system
MV	Máquina Virtual
PHP	Personal Home Page
SO	*Sistema Operacional*

LISTA DE ACRÔNIMOS

DOS	*Denied of Service*

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos, a segurança das redes de computadores vem ganhando significativa importância. Isso acontece devido ao crescimento na utilização dos sistemas computacionais, principalmente daqueles voltados à plataforma *web*. Além disso, o fluxo de informações sigilosas trafegando pelas redes é cada vez mais intenso, gerando muitas tentativas de ataques e invasões para roubo de informações que acabam sendo bem-sucedidas.

Portanto, é de grande importância contar com ferramentas de segurança que possibilitem manter as trocas de informações de forma confiável entre usuários de uma rede ou até mesmo de redes diferentes.

Sabendo da importância dos serviços nas redes para as empresas, várias ferramentas começaram a ser usadas para se tratar da segurança. Dentre elas podem ser citados os *honeypot*s. H*oneypot* é “um recurso de rede cuja função é de ser atacado e comprometido” (SPITZNER, 2002, p.340).

Em outras palavras, *honeypots* são ferramentas que fazem um monitoramento das atividades do atacante, capturando seus dados, ou seja, gerando *logs* do sistema, e verificando as falhas utilizadas para atacar o sistema. Assim, possibilitam ao administrador da rede ou sistema fazer as necessárias mudanças ou reparos para que se melhore a segurança dos mesmos.

Com a possibilidade do atacante apagar os *logs* do sistema, surgiram as chamadas *honeynets*: são redes que têm em sua arquitetura, sub-redes de honeypots. Deste modo o gerenciador do sistema da rede pode guardar os *logs* gerados em diferentes computadores, tornando o seu ambiente de rede mais protegido (FRANCO, 2004, p.564).

Em suma, o presente trabalho visa estudar a detecção de comandos de ataques em uma rede de computadores utilizando *honeypots* e verificar seus comportamentos diante da execução do worm Meajay[1].

[1] Worm programado na linguagem de programação “C” que explora a vulnerabilidade de sistemas operacionais para se espalhar na rede.

1.1 OBJETIVOS

A seguir são descritos os objetivos gerais e específicos deste trabalho.

1.1.1 Objetivo Geral

Avaliar o uso de *honeypots* para detecção de worms como estratégia de segurança de redes.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Definir 4 (quatro) ferramentas *honeypots* para avaliação e detecção de um *worm (Meajay),* bem como comandos de ataque na rede de computadores;
- Utilizar e comparar as ferramentas em sistemas operacionais Windows e Linux;

1.2 JUSTIFICATIVA

Uma ferramenta *honeypot* pode ser utilizada para simular serviços com falhas de segurança, objetivo de ser atacada e invadida. Posteriormente, o responsável pela segurança de redes ou o administrador da mesma pode analisar as vulnerabilidades exploradas, tal como a utilização pelos invasores. A principal vantagem é que os *honeypots* não possuem nenhum serviço “real” disponível na rede, seus arquivos são falsos com a intenção de atraírem o invasor.

Normalmente são usados junto com um IDS (*Intrusion Detection System*) e um *Firewall* para um melhor resultado (BURDINO, 2006).

Existem vários tipos de ataques, dentre eles um dos mais conhecidos é o DOS (*Denial of Service*). Neste tipo de ataque ocorre uma quantidade excessiva de requisições de serviço, que são feitas por máquinas que estão contaminadas com programas chamados “zumbis”, que ao serem acionados, mandam várias

requisições a um determinado servidor, causando lentidão ou até deixando serviços indisponíveis (SANTANNA, 2006).

Umas das ferramentas de prevenção contra ataques é o *Firewall*, definida como uma parede corta-fogo, como se fosse uma proteção colocada entre o computador e a Internet, tendo como fogo os ataques e outros perigos da Internet, onde o Firewall tem como função bloquear esses perigos (BATTISTI, 2005).

Outro mecanismo de defesa contra ataques é o IDS (*Intrusion Detection System*). Essa ferramenta tem como objetivo monitorar uma rede ou computador a procura de sinais que caracterizem um ataque. É considerado limitado por apenas detectar a ameaça e emitir um sinal de alerta para o administrador de rede.

Para Trentin *et al*. (2002) IDS é um mecanismo seguro, que permite o monitoramento de um computador ou de uma rede.

O IDS proporciona um alerta ao administrador da rede para que este possa tomar ações corretivas, sejam elas: bloqueando determinadas portas vulneráveis, negando acesso ao endereço de um determinado IP ou desligar serviços em execução que possam proporcionar ataques de hackers (ASHOOR e GORE, 2011). As suspeitas são programadas a partir de ataques previamente já conhecidos pela ferramenta, ou seja, são armazenadas as assinaturas de vírus, malwares e ataques conhecidos.

Para Franco (2003) honeypots foram sistemas desenvolvidos para atender a necessidade de compreender o perfil dos ataques bem como de detectar as últimas tendências relativas às vulnerabilidades mais exploradas.

Uma vantagem da ferramenta *Honeypot* é que ela trabalha isoladamente, todas as informações sobre o comportamento do atacante são gravadas em *logs*, assim facilitando o entendimento na hora da análise para se tomar medidas de prevenção. Outra vantagem seria o fato dessa ferramenta diminuir a quantidade de alertas falsos gerados pelo IDS. Assim, o correto é estabelecer uma configuração de segurança de rede para que o IDS trabalhe da melhor maneira possível junto ao *Honeypot* (BURDINO, 2006).

Como todo sistema tem suas vantagens e desvantagens. Uma das desvantagens de determinados honeypots é a sua limitação de reconhecer determinados comandos de ataques por parte do invasor, bem como não coletar informações simultaneamente como a data do ataque e o IP do invasor (ANDRADE,

2009); deixando assim, a cargo do administrador da rede a verificação dos logs gerados pelo honeypot.

No entanto, determinados honeypots permitem ao atacante conseguir manipular as informações de acesso ao servidor, fazendo com que a análise das informações sobre os ataques se torne em vão, uma vez que muitos invasores apagam os logs que registram as suas identificações: IP e Endereço Mac (NED, 1999);

Durante o decorrer deste trabalho, o objetivo será comparar diferentes ferramentas *Honeypots* existentes, demonstrando eficiência e possibilidades oferecidas por cada sistema. Desse modo, pretende-se oferecer aos possíveis usuários o suporte para uma fácil escolha que atenda as necessidades reais do sistema de segurança a ser implantando.

1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em cinco capítulos.

No primeiro são descritos os objetivos e a motivação do mesmo.

O segundo capítulo apresenta a fundamentação teórica, onde são encontradas informações relacionadas aos conceitos de segurança relacionados à ciência da computação, vários tipos de ameaças e *honeypots*.

No terceiro capítulo aborda-se a metodologia do projeto, apresentando informações sobre a utilização de máquinas virtuais e as características dos *honeypots* escolhidos para serem comparados.

No quarto capítulo são externados os resultados obtidos com o desenvolvimento deste trabalho, abordando de forma detalhada as singularidades de cada *honeypot* escolhido.

E por fim, o quinto capítulo apresenta a conclusão do trabalho e as indicações para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentadas informações necessárias para o entendimento de termos que serão mencionados posteriormente, relacionados a tipos de ataques e ferramentas de segurança abordadas neste trabalho.

2.1 CONCEITO DE REDES DE COMPUTADORES

Conforme Tanembaum e Wetherall (2011) as redes de computadores significam um conjunto de computadores autônomos interconectados por uma única tecnologia.

Para SOARES, LEMOS e COLCHER (1995) uma rede de computadores é formada por um conjunto de módulos de processadores capazes de trocar informações e compartilhar recursos, interligados por um sistema de comunicação, conforme a Figura 1.

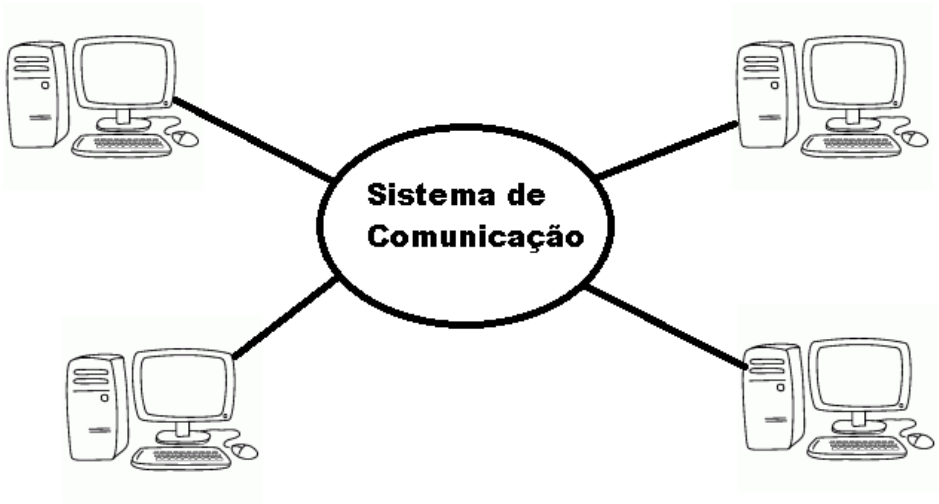

Figura 1 - Rede de computadores.
Fonte: Autoria própria.

Para William Stallings (2005) a troca de informações entre os computadores para fins de ação cooperativa geralmente é conhecida como comunicação de computadores. De modo semelhante, quando dois ou mais computadores estão interconectados por uma rede de comunicação, o conjunto de estações de computadores é chamado de rede de computadores.

2.1.1 Protocolo FTP

Para Tanembaum e Wetherall (2011) o FTP (*File Transfer Protocol*), é um protocolo de transferência de arquivos padrão, o qual insere endereços IP no corpo do pacote para o receptor extrair e usar.

Segundo William Stallings (2005) o protocolo FTP permite a transferência de arquivos de um computador para o outro utilizando a rede de computadores. Complementa Stallings (2005) que o protocolo FTP é usado para enviar arquivos de um sistema para o outro sob comando do usuário. Sendo que os arquivos de texto e binários são acomodados, e o FTP oferece recursos para controlar o acesso do usuário.

Para SOARES, LEMOS e COLCHER, *apud* Postel (1995) o FTP permite que um usuário em um computador transfira, renomeie ou remova arquivos remotos; ou crie, remova e modifique diretórios remotos.

A operação FTP baseia-se no estabelecimento de duas conexões entre o cliente (módulo FTP que está solicitando o acesso a arquivos remotos) e o servidor (módulo FTP que fornece acesso a seus arquivos locais), como ilustra a Figura 2. Sendo que esta conexão denomina-se conexão de controle, a qual é usada para transferência de comandos; e a outra denominada conexão de transferência de dados, a qual é usada para transferência de dados. A conexão permanece aberta enquanto durar a sessão FTP.

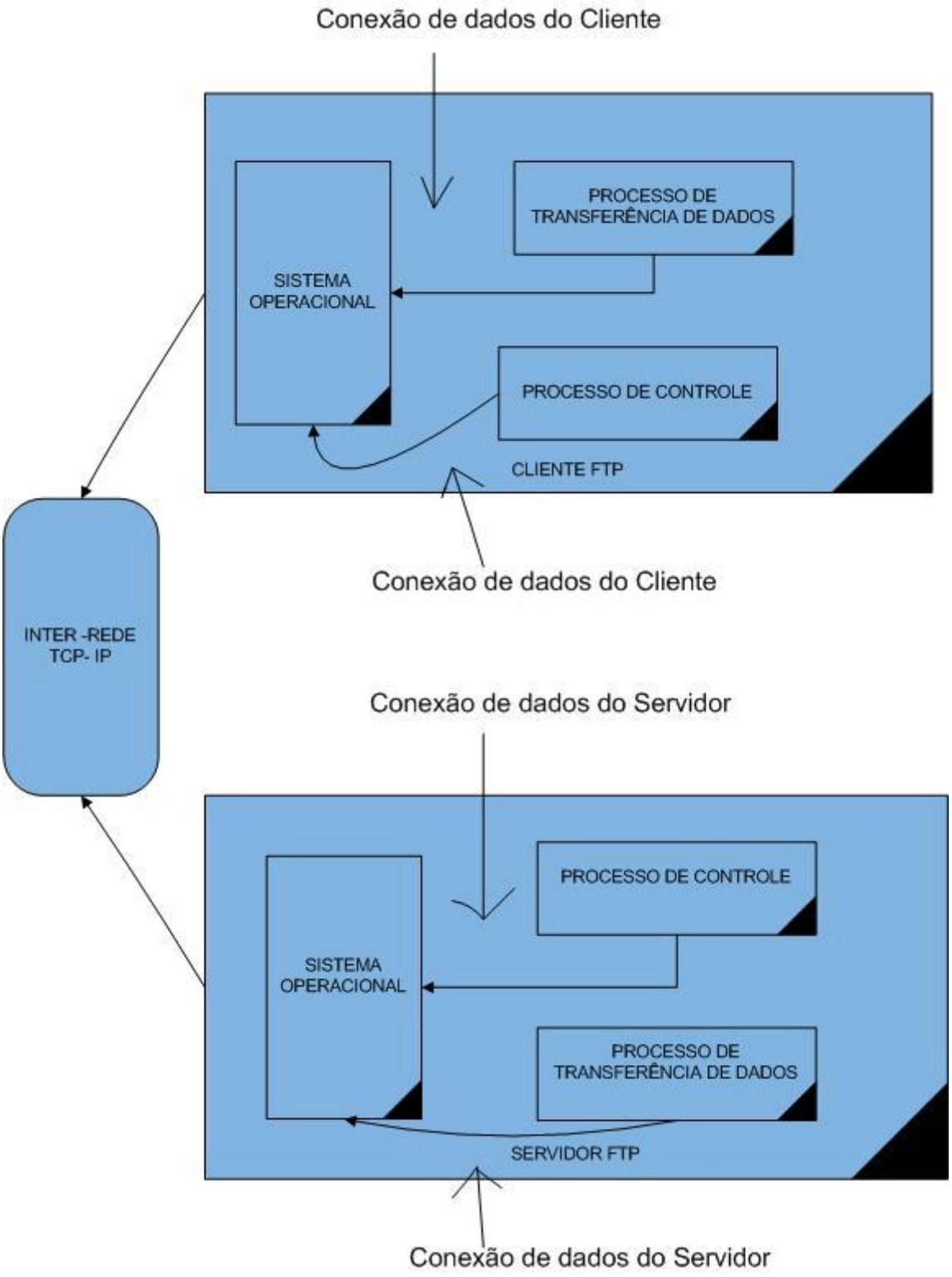

Figura 2 - Conexão FTP de controle de transferência de dados
Fonte: Autoria própria.

Para esse tipo de acesso, os seguintes parâmetros adicionais são obrigatórios: *name*, o nome do arquivo; e site, o nome do domínio do host onde o arquivo reside.

Já os parâmetros opcionais são: *directory*, o diretório em que o arquivo está localizado, e *mode*, que indica a utilização do FTP para transmitir os arquivos.

Antes de a transferência do arquivo ocorrer, o usuário terá de fornecer um ID de usuário e senha. Sendo que estes não são transmitidos com a mensagem, por motivos de segurança. (STALLINGS, 2005).

2.1.2 Protocolo TELNET

O protocolo TELNET oferece a capacidade de logon remoto, que permite que um usuário em um terminal ou computador pessoal efetue logon com um computador remoto e funcione como se estivesse conectado a esse computador.

O protocolo foi desenvolvido para funcionar com terminais simples, no modo de rolagem.

TELNET, na realidade é implementado em dois módulos: O usuário TELNET interage como modo de I/O (input e output) do terminal para se comunicar com o terminal local. Ele converte as características dos terminais reais para o padrão de rede e vice-versa. (STALLINGS, 2005).

O TELNET interage com uma aplicação, atuando como um manipulador de terminal substituto, de modo que os terminais remotos apareçam como locais à aplicação. O tráfego de terminais entre o Usuário e o Servidor TELNET é transportado em uma conexão TCP.

Ensinam SOARES, LEMOS e COLCHER, *apud* Postel (1995) que o protocolo TELNET permite que um usuário utilizando uma máquina “A” estabeleça uma sessão interativa com uma máquina “B” na rede. A partir daí, todas as teclas pressionadas na máquina “A” são repassadas para a máquina “B” como se o usuário estivesse utilizando um terminal ligado diretamente a ela.

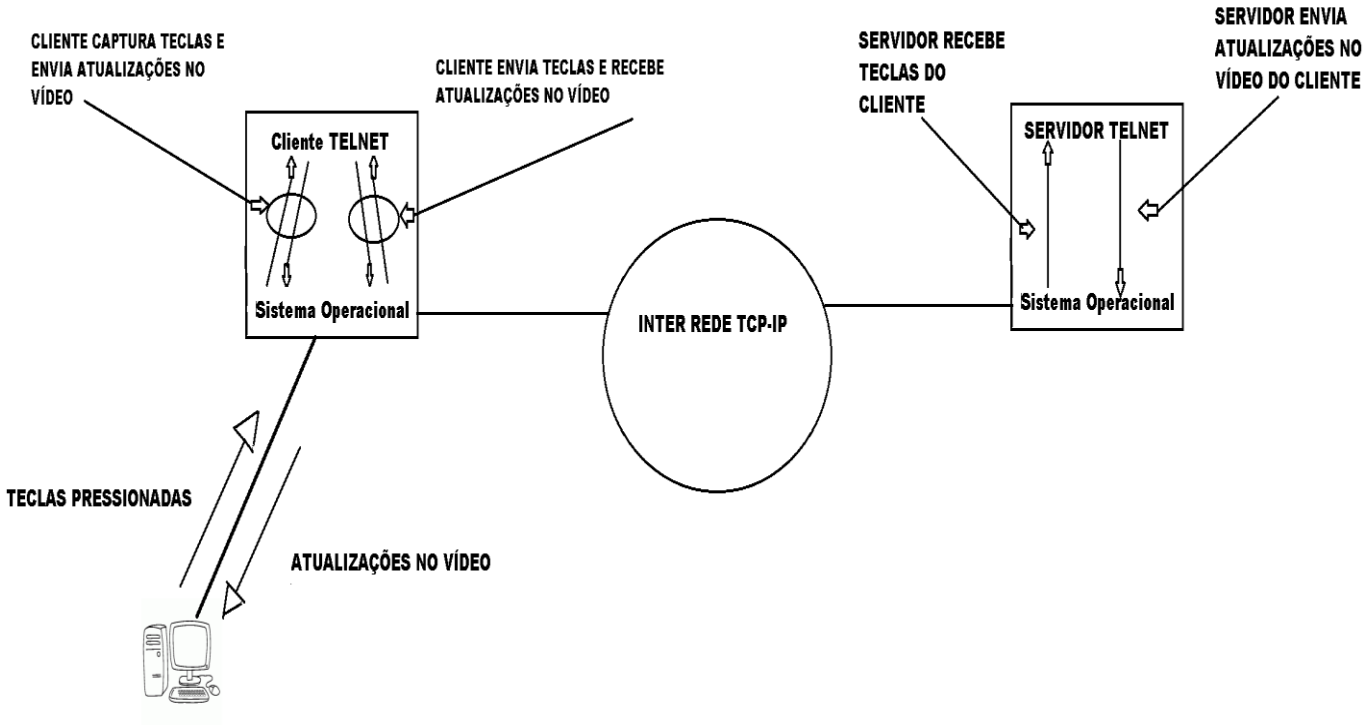

Figura 3 - Conexão TELNET
Fonte: Autoria própria.

A Figura 3 ilustra o funcionamento de uma conexão TELNET. Quando o programa TELNET começa a ser executado em uma máquina, ela passa a funcionar como cliente TELNET. Junto com o comando que dispara a execução do TELNET, o usuário informa o nome, ou o endereço IP, da máquina à qual deseja se conectar.

Uma conexão TCP é estabelecida, e a máquina de destino assume o papel de servidor TELNET.

Uma vez estabelecida a conexão, todas as teclas pressionadas pelo usuário são capturadas pelo cliente TELNET e enviadas, através da conexão TCP, ao processo servidor TELNET na máquina remota. O servidor processa as teclas e envia de volta para o cliente os caracteres que devem ser mostrados no vídeo do terminal (SOARES, LEMOS e COLCHER, 1995).

2.1.3 Protocolo SSH

O protocolo Shell (SSH) Protocol Secure é um protocolo para *login* remoto seguro a outros serviços de rede seguras que são acessados através de uma rede insegura (BARRETT, SILVERMAN e BYRNES, 2005).

Ele consiste em três componentes principais:

- O protocolo Transport Layer - SSH-TRANS - fornece serviço de autenticação, confidencialidade e integridade. Pode opcionalmente também fornecer compressão. A camada de transporte será tipicamente executada através de uma conexão TCP / IP, mas também pode ser usado sobre qualquer outro fluxo de dados confiável.

- O usuário autentica - SSH-USERAUTH - utilizador do lado do cliente para o servidor. Esta operação ocorre sobre a camada de transporte.

- A Conexão - SSH-CONNECT - multiplexa o túnel em vários canais lógicos. Funciona sobre o usuário protocolo de autenticação.

O Secure Shell é uma abordagem popular, poderosa, baseada em software de segurança de redes. Sempre que os dados são enviados por um computador da rede, automaticamente são criptografados pelo protocolo SSH. Quando os dados chegam ao seu destino, são decifrados automaticamente pelo protocolo SSH.

O resultado é a criptografia transparente: usuários podem trabalhar normalmente, sem saber que suas comunicações são encriptadas com segurança na rede.

Além disso, o SSH utiliza modernos algoritmos de criptografia seguros e é eficaz o suficiente para ser encontrado dentro de aplicações de missão crítica em grandes corporações (BARRETT, SILVERMAN e BYRNES, 2005).

SSH tem uma arquitetura cliente / servidor, como mostrado na Figura 4.

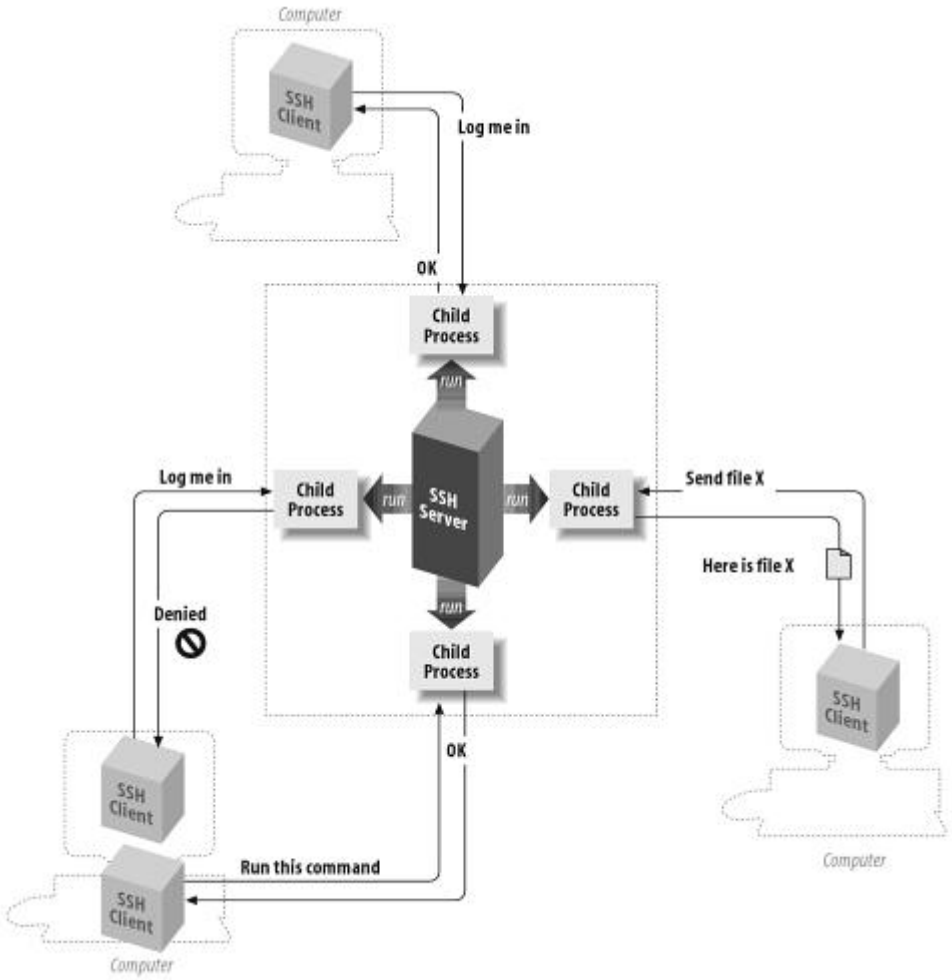

Figura 4 - Arquitetura Secure Shell.
Fonte: Adaptado do livro: SSH, The Secure Shell: The Definitive Guide (BARRETT, SILVERMAN e BYRNES, 2005)

Um programa servidor SSH normalmente instalado é executado por um administrador de sistema, ele aceita ou rejeita entrada de conexões com o seu computador/host. Os usuários executam programas clientes SSH, normalmente em outros computadores, para fazer solicitações ao servidor SSH, como "Por favor, me faça o *login*".

“Por favor, envie-me um arquivo", ou "Por favor, execute este comando”. Todas as comunicações entre clientes e servidores são criptografadas e protegidas contra modificações.

2.2 CONCEITO DE SEGURANÇA

O termo segurança é usado com o significado de minimizar a vulnerabilidade de bens e recursos. Vulnerabilidade é qualquer fraqueza que pode ser explorada

para se violar um sistema ou informação que ele contém (SOARES, LEMOS e COLCHER, 1995).

Par poder compreender os tipos de ameaças à segurança de redes que existem, é preciso ter a definição dos requisitos de segurança.

A segurança de computadores e de rede trata de quatro requisitos, conforme cita Stallings (2005):

- **PRIVACIDADE:** Exige que os dados sejam acessíveis apenas por pessoas autorizadas. Esse tipo de acesso inclui impressão, exibição e outras formas de exposição de dados, inclusive simplesmente revelar a existência de um objeto.
- **INTEGRIDADE:** Exige que apenas pessoas autorizadas possam modificar dados. Sendo que a modificação inclui escrever, alterar, mudar o estado, excluir e criar.
- **DISPONIBILIDADE:** Exige que os dados estejam disponíveis às pessoas autorizadas.
- **AUTENTICIDADE**: Exige que um host ou serviço seja capaz de verificar a identidade de usuário.

Uma ameaça consiste em uma possível violação da segurança de um sistema. Algumas das principais ameaças às redes de computadores são:

2.2.1 Engenharia Social

Termo usado para explicar um método de ataque quando se faz uso da persuasão, geralmente o usuário tem pouca experiência e acaba caindo em golpes, liberando informações pessoais e até de contexto financeiro.

Para Silva *et al.* (2012) a engenharia social é vista como um dos malwares mais velhos do mundo e mesmo assim continua muito utilizada. Controlar pessoas pelas informações é algo difícil, entretanto, possível. Declara-se que o “elo mais frágil” da segurança de dados e informações confidenciais não está no sistema, e sim, na pessoa que interage com este sistema.

Engenheiros sociais procuram explorar falhas de sistemas para atingir pessoas ingênuas e tirar informações sigilosas até de grandes empresas (BORGES, 2006).

2.2.2 Vírus

Vírus de um computador é um tipo de programa de software espúrio que infecta outros programas de software ou arquivos de dados, a fim de ser executado, geralmente sem conhecimento nem permissão do usuário. (LAUDON; LAUDON, 2007; TURBAN *et al*., 2009)

Borges (2006) ensina que Vírus são softwares programados com intenções maliciosas e que na maioria das vezes se multiplicam através de programas ou arquivos. É chamado desta forma por se parecer com o vírus que citamos na vida real, que se propaga rapidamente, vive num ambiente próprio e se move através de infecções.

2.2.3 Backdoors

Como a tradução diz “porta dos fundos”, trata-se de um processo que será referente a uma futura invasão no sistema. Existe uma variação entre seu grau de ocultação, geralmente trabalham em user-space (espaço de usuário) e em kernel-space (espaço de núcleo) (BORGES, 2006). Conforme Silva (2001), backdoors, ou porta dos fundos, são maneiras criadas por um usuário malicioso para invadir um sistema a qualquer momento.

2.2.4 Cavalo de Tróia

Também conhecido como “*trojan”* faz uma analogia com a história, com gregos presenteando os troianos com um enorme cavalo de madeira, porém com o exército escondido no interior do cavalo. Este tipo de ameaça é bastante similar, aparentemente é um arquivo inofensivo, no entanto tem como meta abrir portas para invasores. (BORGES, 2006).

Para DAMATTO e RALL (2010) cavalo de troia é um programa que parece ser útil ou inofensivo, porém tem códigos ocultos criados para explorar ou danificar o sistema no qual foi executado.

A diferença entre o cavalo de Tróia e o *backdoor* é quando se trata de inserção de cavalo de Tróia, o administrador do sistema esta ciente da inserção do arquivo, mas não sabe que este arquivo é uma ameaça a segurança do sistema.

Já com *backdoor* não existe esse consentimento do administrador.

2.2.5 Worms

Worm, ou seja, verme é um código mal intencionado autopropagável de um computador para outro, por meio da rede. (DAMATTO e RALL, 2010)

Possuem características semelhantes aos dos vírus, a diferença está no método de como ele é espalhado. Os *worms* se disseminam através de redes de computadores e correios eletrônicos e não em arquivos de um sistema operacional.

2.2.6 Spywares

Programas cujo objetivo é monitorar atividades de um sistema e envia-las para outras pessoas. (SILVEIRA, 2010)

Estas ameaças têm como função comprometer a privacidade do usuário. Algumas buscam informações sigilosas, outras tentam mudar configurações sem o consentimento do usuário (BURDINO, 2006).

Spywares se alocam geralmente em *browsers*, com isso mudam a página principal quando o usuário abre o navegador, também adicionam barras de busca que deixam a navegação mais lenta (BURDINO, 2006).

2.2.7 Rootkits

Em Shadowserver (2007) e Murilo (2006) rootkit é um conjunto de programas maliciosos, projetados para modificar o sistema operacional do computador atacado, para ocultar do usuário a presença de invasão de outros programas maliciosos.

Segundo Brudino (2006) rootkits buscam se esconder de softwares de proteção e segurança utilizando métodos avançados de programação. Escondem suas chaves no registro e também seus processos no Gerenciador de Tarefas, assim dificultando bastante sua remoção.

2.2.8 Sniffer

Silva (2001) ensina que sniffers são quaisquer procedimentos que capturem informações ao longo da rede, seja por hardware ou software. Os sniffers são muito utilizados para espionagem de informações.

As ferramentas sniffers fazem captura e análise de pacotes em redes de computadores. Não são necessariamente usadas para fins maliciosos, mas podem ser usadas para conseguir informações condifenciais, senhas e outras informações (BURDINO, 2006).

2.2.9 Phishing/Scam

Considerado uma fraude eletrônica que faz uso da engenharia social para realizar seus objetivos. Basicamente envia e-mails fraudulentos com o objetivo de obter senhas e dados bancários (BURDINO, 2006).

2.2.10 Brute Force

Técnica para romper a criptografia, tentando um grande número de possibilidades. Utilizado para descobrir senhas curtas (BURDINO, 2006).

2.2.11 Exploit

Sequencia de comandos, ou seja, porção de dados ou um trecho de código que se aproveita de vulnerabilidades de um software para se obter um acesso ilícito (BURDINO, 2006).

2.2.12 Arp Spoffing

Técnica onde o atacante se passa por vítima, recebendo todos os dados e fazendo um processo análogo ao servidor (BURDINO, 2006).

2.2.13 Denial of Service (Dos)

Os ataques de negação de serviço realizam uma sobrecarga, através de uma grande quantidade de solicitações de serviços, dessa forma o sistema fica instável e geralmente para de funcionar corretamente. Para fazer várias solicitações é preciso espalhar vários programas *zumbis* em máquinas de uma rede, assim pode ser possível que um usuário tenha sua máquina utilizada para fazer um ataque sem que se de conta disso (BORGES, 2006).

2.3 TIPOS DE ATACANTES

Os atacantes são classificados da seguinte forma:

- Script Kiddies: não possuem grande conhecimento em programação, porém desejam prejudicar o maior número de sistemas com o menor esforço possível.
- Advanced Blackhats: estes são hackers experientes e utilizam seus conhecimentos para invadir e comprometer sistemas importantes, geralmente criam suas próprias ferramentas tornando difícil sua identificação. Sabem como não deixar rastros limpando arquivos de logs ou modificando kernel da máquina.
- Lammers: estes são facilmente descobertos por usarem o mesmo tipo de método de invasão e por serem curiosos, deixam sua assinatura eletrônica por onde passam.
- Carders: criminosos que se aproveitam dos usuários da internet para roubar cartões de crédito e se beneficiar com isso (SYMANTEC, 2009).

2.4 TIPOS DE ATAQUES

2.4.1 Alvos

Primeiramente para existir um ataque é necessário eleger uma empresa que apresenta vulnerabilidades em sua estrutura. Na maioria das vezes os usuários pensam que não serão vítimas de ataques, mas suas máquinas podem ser utilizadas como cobaias em ataques e também podem ser usadas para armazenar informações roubadas (BURDINO, 2006).

2.4.2 Motivação

Os motivos são construtos hipotéticos, construídos por teóricos, para fazer o comportamento mais compreensível e previsível (STIPEK, 1993).

Motivação é o processo pelo qual uma ação ou um conjunto de ações são iniciados tendo em vista o alcance de uma meta estabelecida. Pode-se dizer que o processo motivacional dá início, dirige e integra o comportamento, sendo um dos principais determinantes do modo como uma pessoa se comporta (BORUCHOVITCH *et al*, 2013).

Para Ryan e Deci (2000) estar motivado significa ser movido para fazer alguma coisa. A pessoa que não sente nenhum impulso ou inspiração para agir é, portanto, caracterizada como desmotivada, enquanto alguém que está energizado ou ativado em direção a um fim é considerado motivado. As pessoas têm não só diferentes quantidades, mas também diferentes tipos de motivação.

Dessa forma, variam não apenas em nível de motivação (ou seja, o quanto de motivação), mas também na orientação de qual motivação (ou seja, qual tipo de motivação). Quando se trata da motivação para um ataque em sistemas de informação que leva as pessoas a ser corromperem normalmente é o dinheiro (BURDINO, 2006).

2.4.3 Ataques de exploração

Este tipo de ataque tem como foco se obter a maior quantidade de informações, porém sem causar dano ao sistema. Como exemplo, pode-se citar a espionagem digital, fraudes e roubo de senhas (BORGES, 2006).

2.4.4 Ataques de paralisação

Também conhecido como ataque de negação de serviço *(DOS- Denied of Service).* Neste tipo de ataque o objetivo é fazer com que um sistema fique indisponível durante um tempo (BORGES, 2006).

2.4.5 Ataques de comprometimento

Esse tipo de ataque compromete o funcionamento de mecanismos em um sistema através da desativação de serviços essenciais nos servidores, Trata-se de destruição de informações do alvo, gasto de recursos e também comprometimento da parte física*, hardware,* (BORGES, 2006).

2.5 HONEYPOTS

Esta seção apresentará primeiramente um desenvolvimento histórico, e após alguns conceitos e definições sobre *honeypot*.

2.5.1 CONCEITOS INICIAIS

Não existe uma definição aceita por toda a comunidade sobre o termo *honeypot*. Spitzner (2002) diz que “Honeypots são recursos computacionais dedicados a serem sondados, atacados ou comprometidos, num ambiente que permita o registro e controle dessas atividades”.

Os *honeypots* e *honeynets* foram criados com o objetivo principal de estudar

os ataques e seus atacantes. Por meio deles, é possível monitorar de forma eficiente grande parte dos ataques gerados para uma determinada máquina ou rede de máquinas conectadas ou não à Internet (SCHNEIER, 2000).

A Figura 5 apresenta a ilustração de uma rede com a presença de um *Firewall* e de um *Honeypot*.

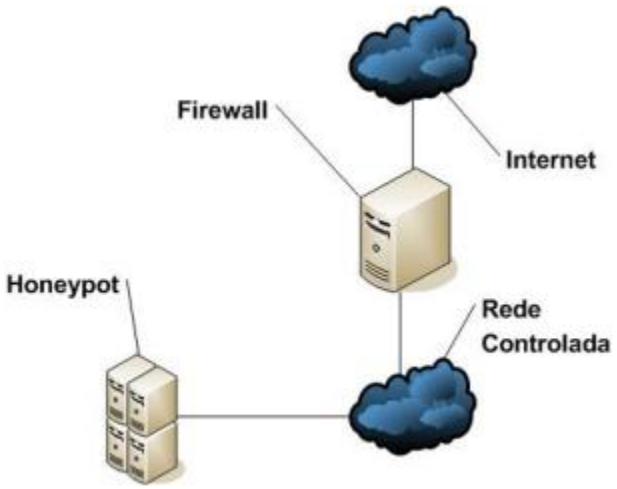

Figura 5 - Exemplo de uma rede com honeypot
Fonte: Adaptado de (HONEYNET, 2002)

Os *honeypots* podem atuar em uma rede de duas maneiras, como *honeypot* de pesquisa e ou de produção. O primeiro tipo é responsável por coletar informações sobre invasores e ferramentas utilizadas pelos mesmos. Em geral, esses *honeypots* são implementados em redes externas ou não interligados com a rede principal. Já o segundo modelo é aplicado em redes internas onde o objetivo é trabalhar como um elemento de distração para invasores, assim diminuindo o risco de ataque a rede principal. Consiste basicamente em direcionar o tráfego para máquinas preparadas para serem alvos de ataques (SILVA e SILVA, 2010).

2.5.2 Classificação de *Honeypots*

Quanto aos níveis, podem-se classificar os *honeypots* em: baixa, média e alta interação.

Conforme Hoepers et al (2007), *honeypots* de Baixa Interatividade podem ser tratados como aqueles que são emulados em serviços virtuais. Ou seja, dentro

de máquinas virtuais que emulam sistemas operacionais e serviços no qual o atacante pode interagir.

Em alguns tipos de Honeypots pode-se até criar arquivos para uma aproximação maior com a realidade. Para tanto o sistema operacional hospedeiro da máquina virtual deve ser muito bem configurado a fim de evitar comprometimento.

Ainda de acordo com Hoepers et al (2007), os honeypots de alta interatividade são dotados de serviços e máquinas reais. Assim, os atacantes interagem com sistemas operacionais, aplicações e serviços de cada máquina ligada na rede.

Entretanto, o invasor encontra uma simulação muito próxima da realidade da instituição.

Jessen e Chaves (2008) dizem que o principal objetivo quando se usa um *honeypot* de baixa interatividade é identificar ataques automatizados e varreduras, atrair atacantes para longe dos sistemas operacionais e coletar o máximo de assinaturas de ataques.

Já Marcelo e Pitanga (2003) comentam que quando é usado o *honeypot* de alta interatividade o objetivo é observar o comportamento e as atividades dos atacantes coletar o máximo de material para pesquisa e elaborar um treinamento com o objetivo de melhorar a segurança da informação.

Todavia, Marcelo e Pitanga (2009) ressaltam que, se os *honeypots* não forem configurados de modo correto e se não forem utilizados por especialistas que tenham grande conhecimento na área, provavelmente esses *honeypots* se tornarão uma grande oportunidade para o atacante invadir sistemas importantes de sua rede.

Portanto, má configuração e utilização do *honeypot* podem proporcionar danos graves.

2.6 *SOFTWARES* E OUTRAS FERRAMENTAS DE SEGURANÇA

Atualmente existe uma grande diversificação de ferramentas de segurança no mercado, a seguir são listadas algumas delas:

- MANTRAP: uma ferramenta comercial desenvolvida para plataforma Solaris, utilizada tanto para segurança interna quanto externa. O Mantrap cria 4 (quatro) ambientes e pode suportar 4 (quatro) Sistemas Operacionais.

Mantrap cria até quatro subsistemas, muitas vezes chamado de "cadeias". Estas cadeias são logicamente sistemas operacionais simulados separados a partir de um sistema operacional “mãe”, o qual os executam (RAO *et al*, 2013).

Cada ambiente é personalizado isoladamente e as atividades dos atacantes registradas no servidor remoto de logs ou localmente (BURDINO, 2006).

- SPECTER: Um das características únicas do Specter é que ele também permite a coleta de informações, ou a capacidade automatizada para reunir mais informações sobre o atacante (RAO et al., 2013).

Para grandes e pequenas empresas, é uma ferramenta para Windows. Monitora até 14 portas TCP, sendo 7 portas de serviços e 7 de armadilhas. Estas armadilhas registram as formas de ataque e as outras portas de serviços emulam a forma de serviço utilizado. (BURDINO, 2006);

- NETBAIT: redireciona ataques para IPs não utilizados, anulando o intruso por causa do uso de arquivos simulados. Ele cria um desvio da rede real e possui gerenciamento remoto centralizado e configuração de comportamento dinâmico. (BURDINO, 2006);

- SMOKE DETECTOR: esta ferramenta emula ate 19 sistemas operacionais, confundindo e aumentando o tempo de resposta ao atacante. Enviam alertas de invasões aos administradores da rede e disfarça servidores importantes. (BURDINO, 2006);

- KFSENSOR: projetado principalmente para proteção contra ataques, é voltado para Windows. Simula serviços do sistema na camada de aplicação (ALMUTAIRI, PHAN e PARISH, 2012). Bloqueia ataques de negação e estouro de buffer, registra os logs do atacante (BURDINO, 2006). Este honeypot atua como um host com base em Sistema de Detecção de Intrusão (IDS). Ele age como um *honeypot* para atrair e detectar *hackers* simulando serviços vulneráveis do sistema e *Trojans*. O sistema é altamente configurável e possui registro detalhado, análise de

ataques e alertas de segurança. Esta abordagem complementa outras formas de segurança e adiciona outra defesa contra a crescente ameaça de segurança enfrentada por todas as organizações;

Alguns Honeypots Free/ Open Source:

- BACKOFFICER FRIENDLY: Este produto é projetado para emular um servidor. BOF (como é comumente chamado) é um honeypot muito simples, mas muito útil desenvolvido por Marcus Ranum. É um excelente honeypot de baixa interação (RAO et al, 2013).

Emula serviços como TELNET, FTP, SMTP, POP3 (BURDINO, 2006);

- HONEYBOT: honeypot com abertura de mais de 1000 soquetes UDP e TCP. Esses soquetes são projetados para simular serviços vulneráveis. É um honeypot para a plataforma do Sistema Operacional Windows. (BROWN et al., 2012).

- VALHALA: Ferramenta de detecção de intrusos baseada em honeypot simula diversos servidores: Web, FTP, Telnet, Finger, Smtp, Pop3, Tftp e Port Forwarding. Ele é um honeypot de baixa interatividade, o que significa que possui um risco menor de comprometimento do que honeypots baseados em sistemas operacionais reais (MOREIRA, 2009). Quando alguém externo tenta acessar algum desses serviços, um aviso é gerado e o administrador pode ver em tempo real as ações do invasor, o qual pensa que realmente conseguiu acesso indevido ao sistema, quando na realidade tudo o que ele faz está sendo monitorado. O programa permite salvar os logs em disco ou enviar remotamente para outra máquina;

- KIPPO: é um honeypot com muitos recursos para visualizar as estatísticas de um SSH. Ele usa uma biblioteca gráfica chamada "Libchart", elaborada por Jean-Marc Trémeaux, QGoogleVisualizationAPI" Wrapper PHP para API Google Visualization, elaborada por Thomas Schäfer e tecnologia de geolocalização geoPlugin (geoplugin.com). (KIPPO, 2013).

Dessa forma, ele simula um servidor SSH. (SARAFIK, REZAK e VOZNAK, 2012) Kippo-Graph, na versão gráfica apresenta atualmente 24 gráficos, incluindo:

10 (dez) melhores senhas utilizadas, top 10 nomes de usuário, top 10 combinações de nomes de usuário e senha, índice de sucesso, ligações por IP, conexões por país, varreduras por dia, varreduras por semana, clientes SSH, top 10 de entrada bem sucedida, top 10 que falharam a entrada etc. Há também dados de geolocalização extraídos e exibidos com tecnologia de visualização do Google usando mapas do Google, um mapa de intensidade, etc.

Por fim, são apresentados dados estatísticos relacionados à entrada das informações dando uma visão geral sobre ações dentro do sistema.

- HONEYD: este foi projetado para Unix, se tornou uma das principais ferramentas para desenvolvimento de Honeypots. Pode assumir a identidade de um IP que não esteja sendo utilizado e interagir com o invasor, responde a suas requisições e registra seu comportamento (BURDINO, 2006);

- LABREA TARPIT: ferramenta feita para Windows ou Unix, tem o intuito de parar ataques ou diminuir a velocidade destes (BURDINO, 2006);

- DECEPTION TOOLKIT: simula várias vulnerabilidades em um sistema, utiliza rotinas para registrar as atitudes do atacante, voltado para plataforma Linux (BURDINO, 2006);

- TINY HONEYPOT: possui um propósito de dificultar a vida do atacante com falsos serviços, gera várias respostas falsas aos pedidos do atacante (BURDINO, 2006).

2.7 MÁQUINAS VIRTUAIS

Uma máquina virtual (VM) na Ciência da Computação é um software que emula o funcionamento de um hardware.

Para Tanembaum (2000, p.42), máquina virtual é idêntica ao hardware verdadeiro, cada uma pode executar qualquer sistema operacional.

Máquina virtual é o nome que se dá a um ambiente, como um programa ou sistema operacional, que não existe fisicamente, mas sim é criado dentro de outro

ambiente, podendo ser tanto na plataforma do SO do Windows quanto do Linux. Dessa forma a máquina virtual é chamada de *guest* (hóspede) e o ambiente que executa essa máquina virtual é chamado de *host* (hospedeiro) (GOMES, C.S; JIN, N.K; CREPALDI T.F, 2007).

Segundo Campos (2003), pode-se definir uma máquina virtual (VM) como uma máquina abstrata, ao contrário de uma máquina emulada, que permite que a máquina real seja particionada de tal modo que diversos sistemas operacionais possam ser executados ao mesmo tempo, e com sistemas operacionais distintos.

2.7.1 Justificativa da utilização de máquinas virtuais

A justificativa de se utilizar máquinas virtuais para desenvolvimento deste trabalho se deve aos seguintes fatores:

1- A possibilidade de se utilizar sistemas operacionais diferentes, consequentemente honeypots diversos tanto para a plataforma Linux quanto para a do Windows;
2- Possui a característica de poder emular uma máquina física, a VM. Permite ter maior segurança quanto a integridade da máquina hospedeira, máquina real, em virtude de um possível ataque de um invasor. Neste trabalho foi utilizado o ataque do tipo *worm Meajay*. Tendo em vista de que se trata de um worm, utilizou-se uma VM, a qual não o permite executar comandos de máquina; sendo assim, sem acesso ao Kernel (núcleo) da máquina real;
3- A facilidade de se instalar diferentes sistemas operacionais e poder encerrá-los sem prejudicar os demais sistemas em execução;
4- Confiança e disponibilidade, uma vez que a falha de um software não prejudica os demais serviços.

2.8 *WORM MEAJAY*

Segundo Burdino (2006) os worms podem ser interpretados como um tipo de vírus mais inteligente que os demais. A sua principal diferença está na forma de propagação.

Os worms podem se propagar rapidamente para outros computadores seja pela Internet, seja por meio de uma rede local.

Para Lucena e Moura (2008) os worms estão dentre os diversos tipos de anomalias que podem ocorrer nas redes de computadores, destacam-se, infestações viróticas automatizadas.

Geralmente, a contaminação ocorre de maneira discreta e o usuário só nota o problema quando o computador apresenta alguma anormalidade. O que faz destes vírus inteligentes é a gama de possibilidades de propagação. O worm pode capturar endereços de e-mail em arquivos do usuário, usar serviços de SMTP (sistema de envio de e-mails) próprios ou qualquer outro meio que permita a contaminação de computadores (normalmente milhares) em pouco tempo.

Foi desenvolvido na linguagem de programação “C”. Tem como intuito explorar a vulnerabilidade dos SO, e se espalhar na rede de computadores.

O seu código fonte pode ser obtido no endereço eletrônico: http://www.meajay.in/static/proj/worm-source.rar.

Na pasta worm-source, constam arquivos do código fonte do Meajay.

O worm Meajay será executado em todos os 4 (quatro) honeypots escolhidos neste trabalho.

3 CARACTERÍSTICAS E CONFIGURAÇÃO DAS FERRAMENTAS UTILIZADAS

3.1.1 Honeypot Honeybot:

Em relação ao HoneyBOT é uma ferramenta de abertura de mais de 1000 (mil) soquetes UDP e TCP, ”escutando” o computador. Estes soquetes são projetados para imitar serviços vulneráveis (BROWN et al., 2012).

Quando um atacante se conecta a esses serviços eles são enganados ao pensar que estão atacando um servidor real.

O honeypot captura de forma segura todas as comunicações com o atacante e registra estes resultados para futura análise. Se ocorrer uma tentativa atacante sobre o servidor, o ambiente do honeypot irá armazenar com segurança esses arquivos em seu computador para análise.

Os servidores dos desenvolvedores do HoneyBOT capturaram vários milhares de trojans e rootkits destes serviços simulados pelo honeypot.

3.1.2 Honeypot Kfsensor:

O KFSensor atua como um *honeypot* para atrair e detectar hackers e worms através da simulação de serviços. Simula serviços do sistema na camada de aplicação (ALMUTAIRI, PHAN e PARISH, 2012).

Ao agir como um servidor de chamariz pode desviar ataques de sistemas críticos e fornecer um maior nível de informação do que pode ser conseguido através de firewalls.

KFSensor é projetado para uso em um ambiente corporativo baseado em Windows e contém características inovadoras e únicas, como gerenciamento remoto e emulações de protocolos de rede do Windows.

Está disponível na versão trial, ou seja, sua duração de forma gratuita expira em 30 (trinta) dias. Possui console de gerenciamento baseado em GUI (*Graphical User Interface*), uma extensa documentação e baixa manutenção, KFSensor fornece uma forma rentável de melhorar a segurança da rede de uma organização.

De fácil configuração e operacionalização. Nenhum hardware especial é necessário e seu design eficiente permite que ele seja executado mesmo em

máquinas configuradas com o SO Windows. Sua interface simples do Windows controla todas as funcionalidades. Não há necessidade de editar arquivos de configuração complexos e vem pré-configurado com todos os principais serviços de sistemas necessários.

KFSensor funciona através da simulação de serviços de sistemas na camada de aplicação. Isto permite-lhe fazer uso dos mecanismos de segurança do Windows e bibliotecas de redes, reduzindo o risco de detecção.

A máquina rodando KFSensor Professional pode ser tratada como apenas mais um servidor na rede , sem a necessidade de fazer mudanças complexas nos roteadores e firewalls da rede.

KFSensor Professional proporciona benefícios imediatos ao revelar a natureza e a quantidade de ataques a uma rede. (ALMUTAIRI, PHAN e PARISH, 2012).

As informações que o KFSensor Professional gera podem ser usadas para refinar regras de *firewall* e produzir novas assinaturas para sistemas de detecção de intrusão de rede. (ALMUTAIRI, PHAN e PARISH, 2012).

3.1.3 Honeypot Valhala 1.8

Valhala Honeypot 1.8 traz o conceito de pote de mel para todos, permitindo facilidade e velocidade. Ele possui os seguintes servidores: WEB, FTP, TFTP, POP3, ECHO, DAYTIME, SMTP, FINGER e PORT FORWARDING. Simula portas de trojans conhecidos (como Netbus, subseven, etc) e ainda possibilita utilizar portas extras. Ele é um honeypot de baixa interatividade, o que significa que possui um risco menor de comprometimento do que honeypots baseados em sistemas operacionais reais (MOREIRA, 2009).

3.1.4 Honeypot Kippo

Kippo é um *honeypot* SSH de interação média projetada para registrar ataques de força bruta e, mais importante, a interação SHELL toda realizada pelo atacante.

Algumas de suas características são:

1- Um sistema de arquivo falsificado com a capacidade de adicionar ou remover arquivos;
2- Um sistema de arquivos falsos, semelhante a uma instalação Debian 5.0;
3- A possibilidade de adicionar arquivos falsos, os quais o agressor possa visualizar como: *'cat'*, etc e passwd.;

Assim, o sistema Kippo salva os arquivos baixados com *wget* para a futura inspeção por parte do administrador da rede (KIPPO).

Softwares necessários:

- Um sistema operacional compatível (Linux, CentOS, FreeBSD e Windows 7)
- Python 2.5 +
- Trançado 8.0 +
- PyCrypto

4 MATERIAL E MÉTODOS

4.1 METODOLOGIA

No presente trabalho foram utilizadas máquinas virtuais, ou seja, softwares que emulam o funcionamento de um hardware. Uma grande vantagem da utilização dessas máquinas é que se uma máquina virtual “travar”, durante os testes, o restante do sistema pode continuar funcionando.

Como cada máquina virtual é idêntica ao hardware verdadeiro, cada uma pode executar qualquer sistema operacional (TANEMBAUM, 2000, p.42). Neste trabalho forma utilizadas máquinas com sistemas operacionais, tanto o Linux e Windows.

A atividade desenvolvida neste trabalho consiste nas seguintes etapas, conforme a Figura 6: (1º) instalação do Sistema Operacional na máquina virtual, utilizando o software Virtual Box; (2º) instalação dos honeypots a serem analisados; (3º) execução de comandos de invasão e a execução de worm Meajay; (4º) monitoramento das informações coletadas pelo honeypot em execução; (5º) geração de logs para a base de dados do honeypot – scripts de monitoramento; e análise de logs armazenados.

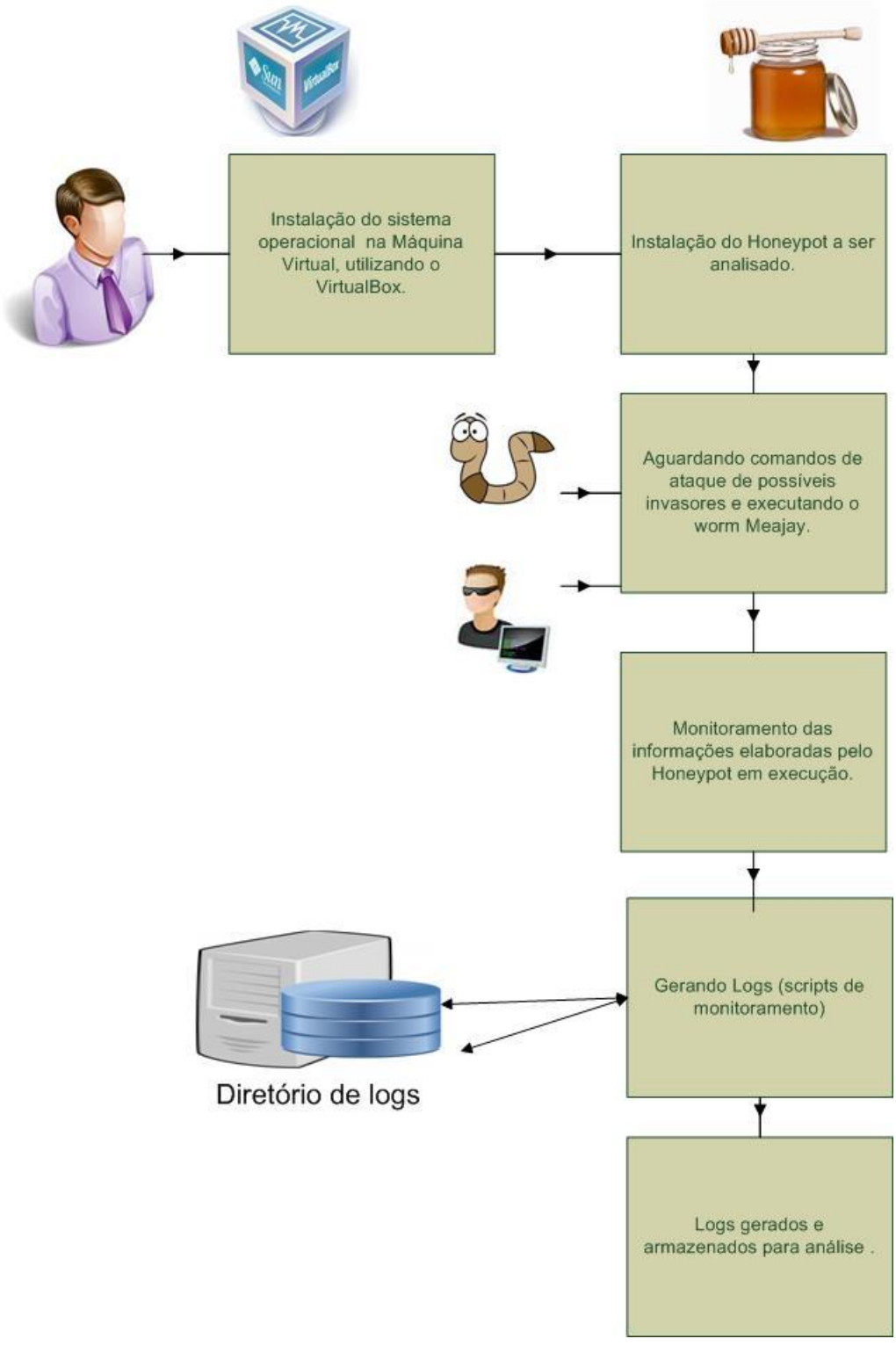

Figura 6 - Atividades relacionadas à detecção de tráfego malicioso na rede de computadores
Fonte: Autoria própria.

Na primeira etapa, ocorreu a instalação do sistema operacional na máquina virtual, utilizou-se o programa Virtual Box, pois é um software de virtualização que

contém também uma versão livre chamada de Virtual Box OSE (Open Source Edition).

Ele permite a instalação de diversas máquinas virtuais, o que pode estar limitado à máquina real, ou seja, a máquina hospedeira da máquina virtual, levando-se em conta as suas configurações: memória RAM, característica do processador e a capacidade de armazenamento do HD (Hard Disk).

O Virtual Box permite a instalação de diversos sistemas operacionais nas máquinas virtuais criadas, entre eles a família Windows, Linux e FreeBSD. Esses ambientes podem ser criados em discos virtuais (arquivos) ou em partições do próprio disco, a opção deve ser realizada de acordo com as necessidades e limitações do ambiente (ORACLE, 2013).

Na segunda etapa, ocorreu a instalação dos 4 (quatro) honeypots, HONEYBOT (plataforma do SO Windows), KFSENSOR (plataforma do SO Windows), VALHALLA 1.8 (plataforma do SO Windows) e KIPPO (plataforma do SO Linux).

Na terceira etapa, foram executados os comandos de ataque e invasão, com a utilização dos protocolos FTP, Telnet e SSH. Inicialmente estes comandos de ataques foram realizados na rede de computadores da Universidade Tecnológica Federal do Paraná- Campus Ponta Grossa (UTFPR-PG).

Muito embora os comandos de ataque estivessem corretos, os honeypots escolhidos não apresentaram nenhuma anomalia e nem mesmo geraram logs desses comandos. Na sequência utilizou-se uma rede doméstica, a qual não apresentava nenhuma configuração de proteção computacional.

Ao refazer os ataques, os resultados gerados pelos honeypots surgiram, ou seja, os logs e os gráficos gerados pelos honeypots estavam em conformidade com os ataques realizados.

Portanto, o que nos levou a compreender que as regras estabelecidas, na configuração da rede UTFPR-PG, dificultaram as análises em relação aos honeypots.

Após, foi executado em cada máquina virtual com o seu honeypot correspondente, o arquivo executável do worm Meajay. Este arquivo executável foi gerado a partir da compilação do seu código fonte.

Na quarta etapa, os logs dos honeypots, em relação a execução do worm, foram checados e guardados, juntamente com os logs de comandos de ataques, para futura exibição dos resultados.

Vale ressaltar que o único honeypot para Linux testado neste trabalho foi o Kippo. Os demais Valhalla, Kfsensor e Honeybot foram executados na plataforma do Sistema Operacional Windows.

As ferramentas elencadas acima foram instaladas e configuradas em máquinas virtuais no Virtual Box.

No Capítulo 4 serão apresentados os resultados de cada ferramenta e, por fim, um comparativo entre as mesmas.

5 RESULTADOS E DISCUSSÕES

Primeiramente, são apresentados os resultados dos ataques utilizando linhas de comando. A seguir, são apresentados os resultados do funcionamento dos *honeypots* utilizados.

5.1 ATAQUES E PROTOCOLOS UTILIZADOS

Para realizar o ataque de invasão foram utilizados os protocolos FTP, SSH, Telnet e a execução do worm Meajay. Assim, após as suas utilizações foram observados os diferentes comportamentos dos *honeypots*, sendo eles a geração de *log*, emissão de efeitos sonoros e a geração de gráficos.

Primeiramente, para se obter o IP das máquinas foi utilizado o comando <ipconfig>, no *prompt* de comando, no caso das máquinas com o sistema operacional Windows XP; já para as máquinas com sistema operacional Linux foi utilizado o comando <ifconfig> no terminal de linha de comando.

Conforme pode ser visualizado na Figura 7, foi utilizado o comando de Telnet, o qual estabeleceu a conexão com a máquina invadida, cujo IP 192.168.1.4.

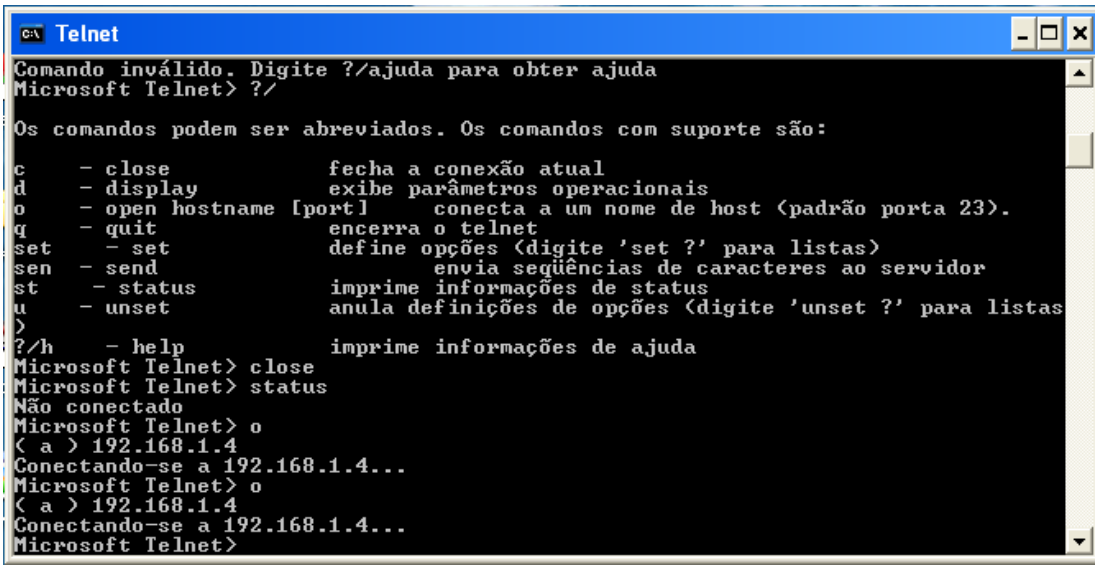

Figura 7 - Comando de invasão por Telnet
Fonte: Autoria própria.

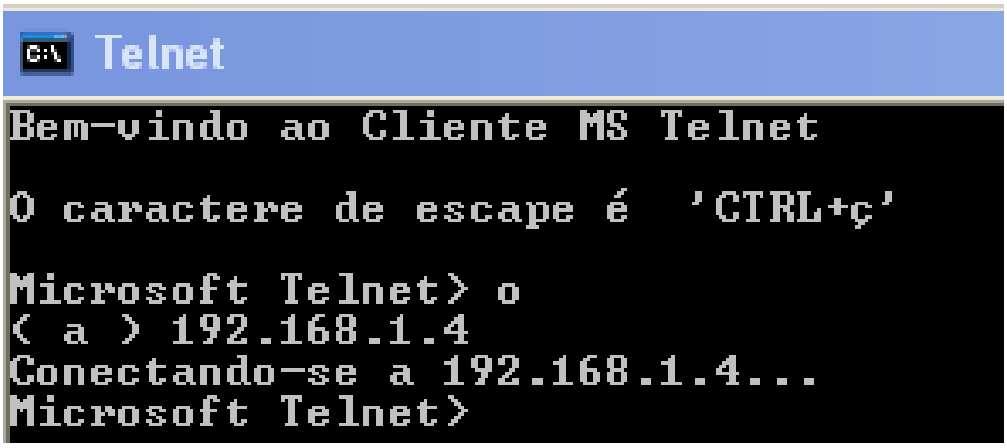

Figura 8 - Conexão com o IP 192.168.1.4
Fonte: Autoria própria.

Na Figura 8 fica evidente que a invasão foi concluída, com a mensagem de “Bem- Vindo ao Cliente MS Telnet”. Com a máquina já invadida foi executado o comando <dir>, o qual mostra para o invasor todos os diretórios constituídos na máquina da vítima, como ilustra a Figura 9.

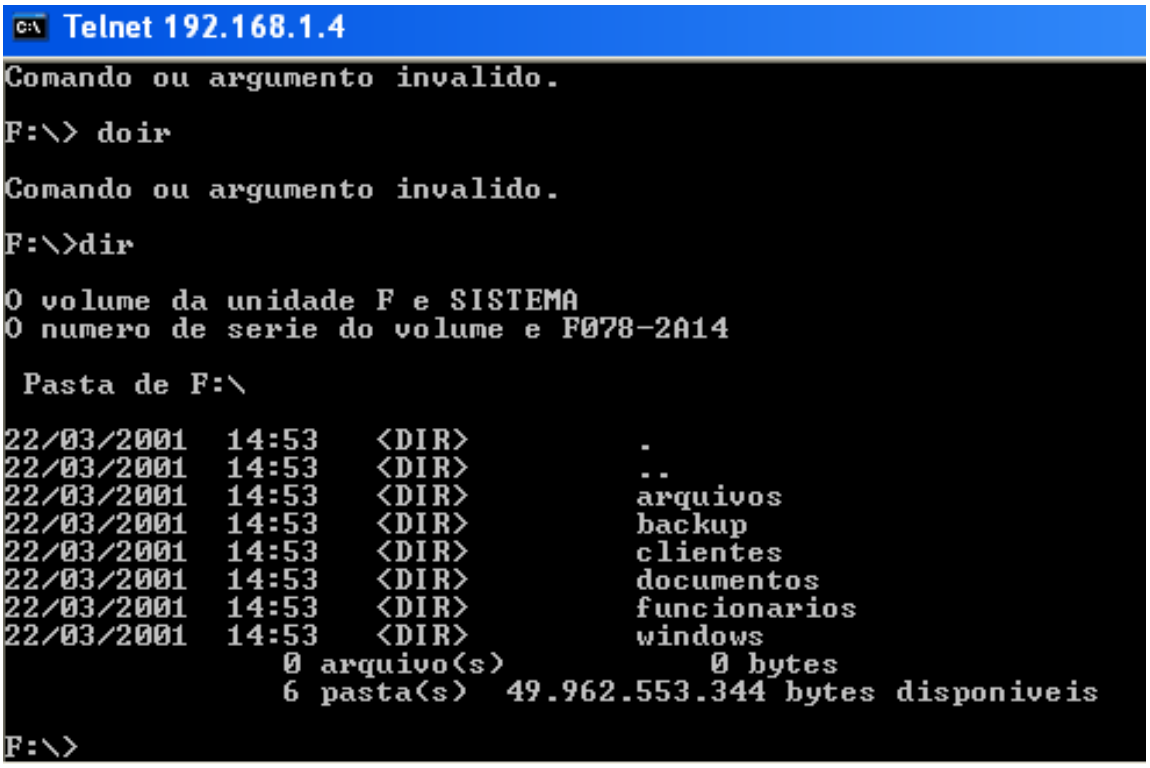

Figura 9 - Diretórios da vítima expostos
Fonte: Autoria própria.

Após a confirmação de sucesso com o Telnet, foi executado o comando de invasão por FTP, conforme ilustra a Figura 10.

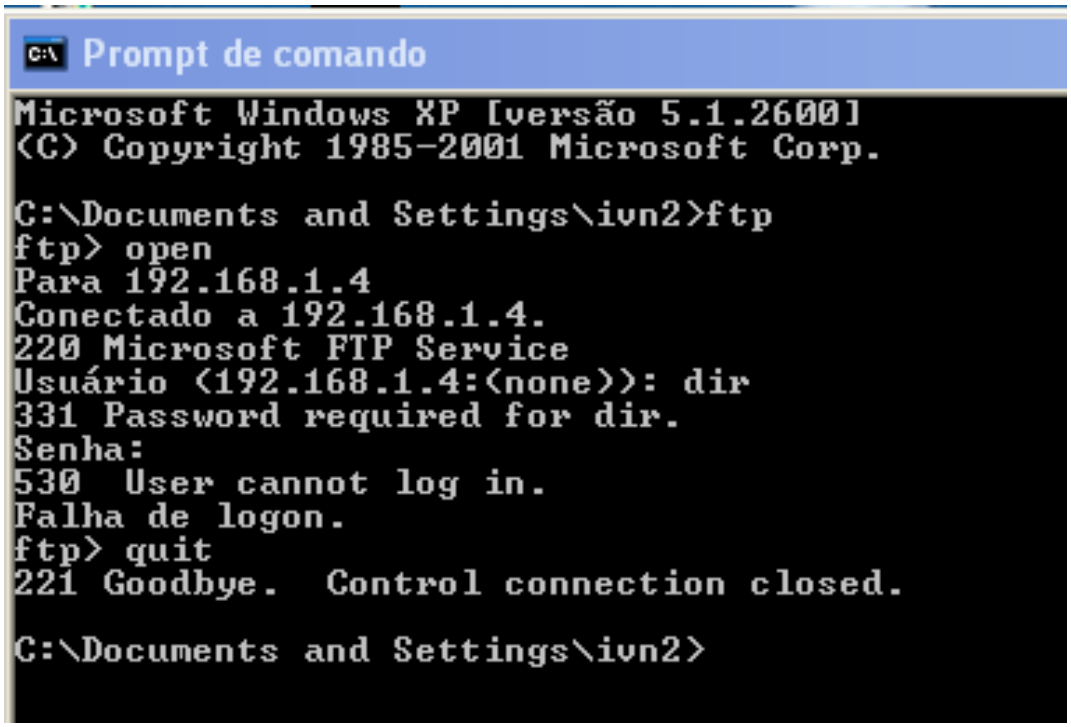

Figura 10 - Invasão pelo protocolo FTP
Fonte: Autoria própria.

Conforme se apresenta na Figura 10, foi executado o comando <open> e após inserido o IP da máquina invadida, neste caso 192.168.1.4. Como a máquina continha senha, foi solicitado na sequência o nome do usuário e a senha correspondente.

5.2 Honeypot Honeybot

Com a execução do Honeybot foi detectado a invasão dos IPs 192.168.56.1 e 192.168.1.6, conforme se vê da Figura 11.

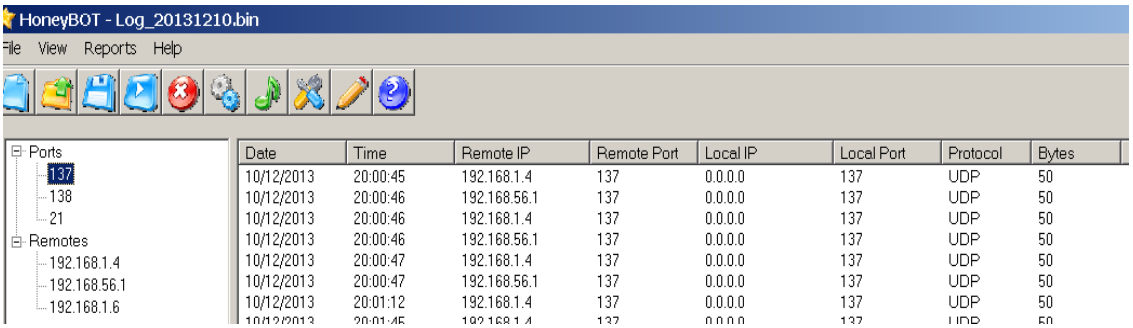

Figura 11 - Logs gerados pelo honeypot Honeybot
Fonte: Autoria própria

Diante das tabelas das Figuras 11 e 12, podem ser observadas as portas 137, 138 e 21, pelas quais foram feitas as invasões (requisições), o IP da máquina correspondente, a data da invasão, o protocolo UDP e a quantidade de Bytes trafegados na respectiva porta.

No momento em que é realizada a invasão, o Honeybot começa a emitir um som correspondente ao de uma coruja.

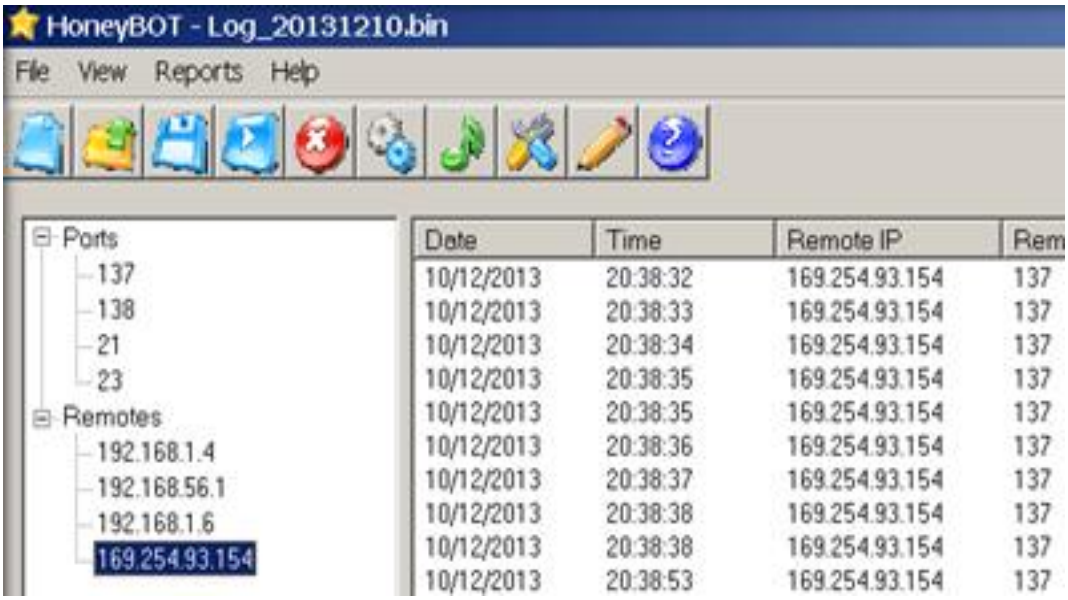

Figura 12 - Listagem de IP desconhecido
Fonte: Autoria própria

Com a execução do *worm Meajay*, foi esperada a reação do Honeybot, o qual imediatamente começou a emitir efeito sonoro. Concomitantemente, o mesmo listou o IP 169.254.93.154 como possível invasor, conforme pode ser observado na

Figura 12. O Honeybot foi o único dos honeypots escolhidos a detectar o worm após a sua execução.

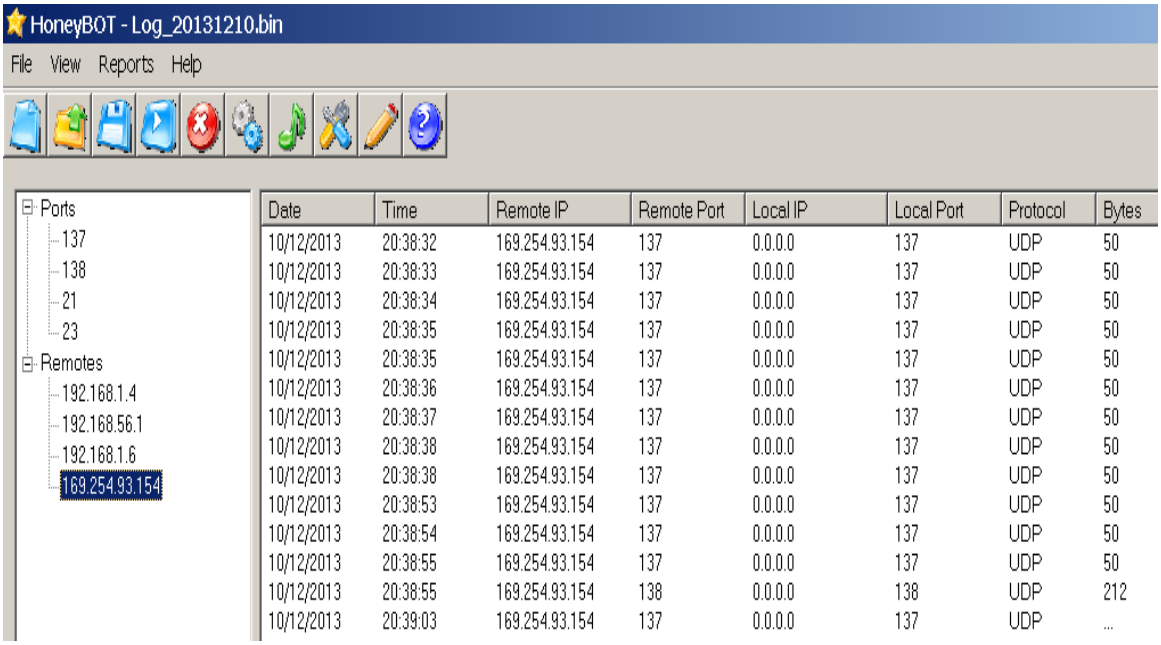

Figura 13 - Quantidades de Bytes gerados pelo IP 169.254.93.154.
Fonte: Autoria própria

A Figura 13 apresenta a quantidade de Bytes utilizados pelo IP 169.254.93.154, cujo valor chega a 212 Bytes. Cabe salientar também a geração de uma nova porta aberta, número 138, na base de *logs* do Honeybot.

5.3 Honeypot Kfsensor

Conforme a Figura 14, o Kfsensor, de forma semelhante ao Honeybot, apresenta uma listagem à esquerda dos IPs das máquinas que estão na rede.

Já a listagem da direita mostra o ID, identificador, de cada máquina e o nome do usuário desta máquina, no caso trata-se do usuário IVN-PC.

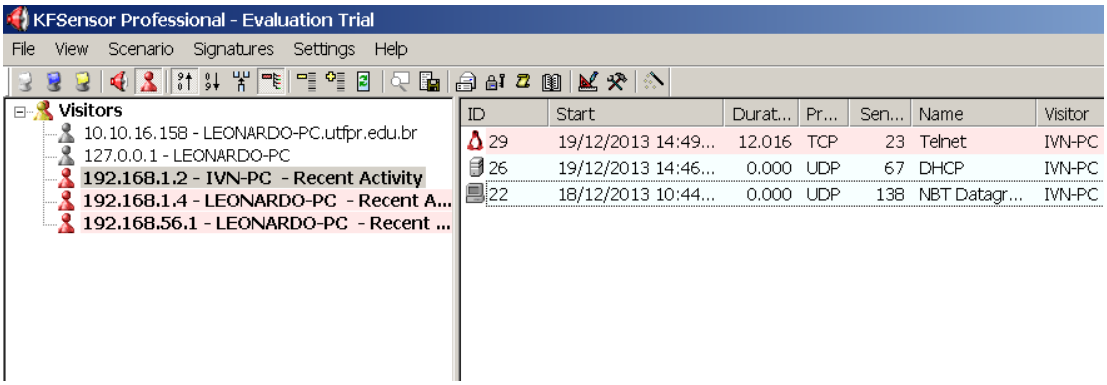

Figura 14 - Listagem KFSensor.
Fonte: Autoria própria

Pode-se observar também que a linha do usuário está com uma marca rosada, indicando que se trata de um possível invasor, que se utilizou do comando Telnet , e de forma simultânea é emitido um som parecido ao de uma sirene para alertar a vítima sobre o atacante em potencial.

A Figura 15 apresenta de forma mais detalhada o *log* gerado pelo Kfsensor.

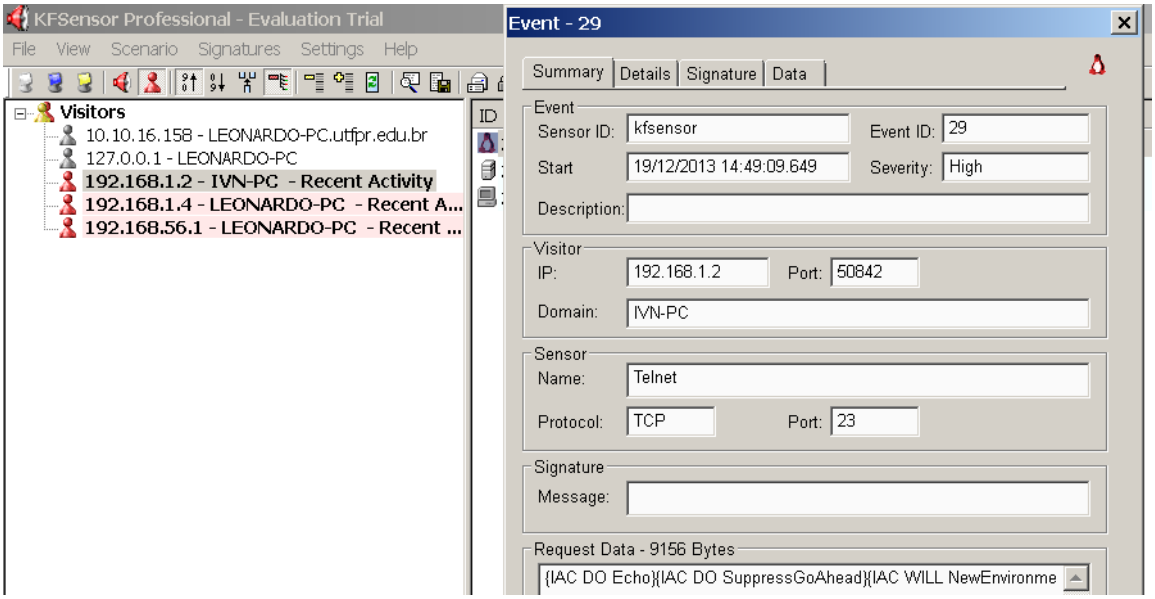

Figura 15 - Dados do invasor com Telnet
Fonte: Autoria própria

É gerado o sumário do ataque efetuado constando: o seu respectivo ID (29), a data da invasão (19/12/2013), o IP (192.168.1.2) e a porta (50842) utilizada pelo invasor, o seu domínio (IVN-PC) e a maneira de invasão (Telnet) com o protocolo (TCP) e a porta (23) utilizada no computador de quem foi atacado.

A Figura 16 apresenta outro sumário do ataque efetuado constando: o seu respectivo ID (46), a data da invasão (19/12/2013), o IP (192.168.1.2) e a porta (51092) utilizada pelo invasor, o seu domínio (IVN-PC) e a operação de invasão (FTP) com o protocolo (TCP) e a porta (21) utilizada no computador de quem foi atacado.

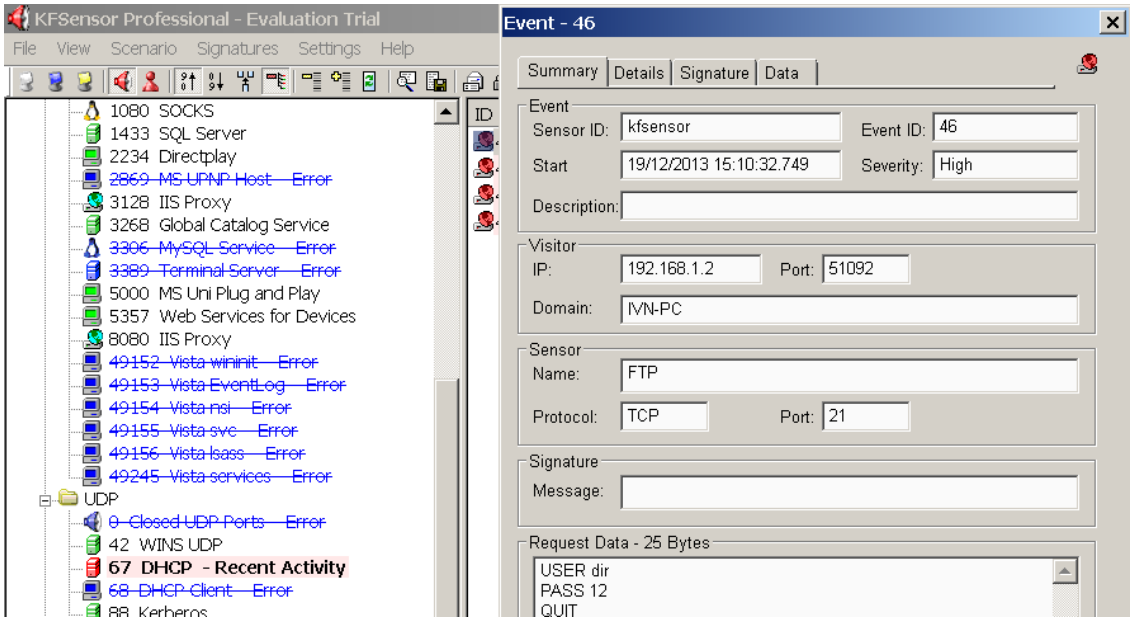

Figura 16 - Dados do invasor por FTP
Fonte: Autoria própria

As Figuras 17 e 18 apresentam as atividades exercidas pelo atacante, Telnet e FTP respectivamente. Como também pode ser visto o registro da duração do ataque sendo 59 segundos (Figura 18) com a utilização do FTP e entre 12 (doze) e 15 (quinze) segundos com o Telnet (Figura 17).

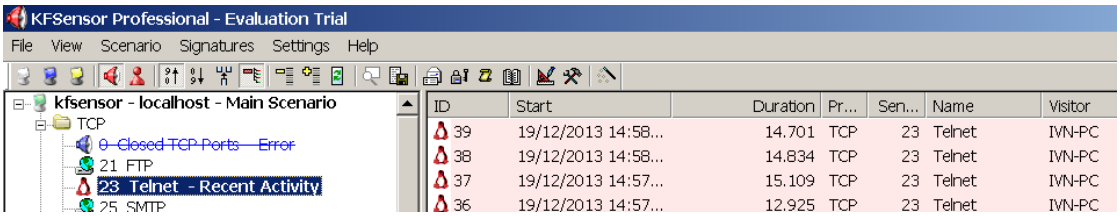

Figura 17 - Atividade Telnet
Fonte: Autoria própria

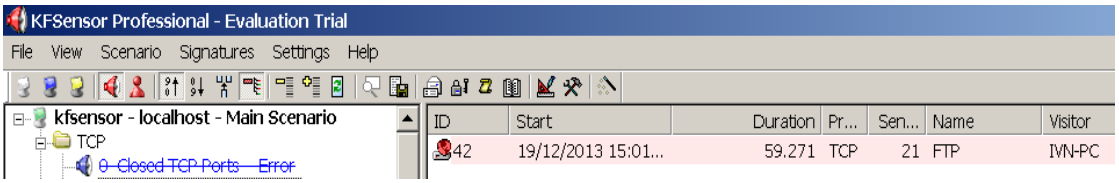

Figura 18 - Atividade FTP
Fonte: Autoria própria

Por fim, com a execução do arquivo executável do *worm,* já explicado no Capítulo 2, o Kfsensor não apresentou nenhuma anormalidade em seus *logs* de verificação, ou seja, esse *honeypot* não gerou dados a serem apresentados, diferentemente do Honeybot.

5.4 Honeypot Valhalla 1.8

Com a execução do Valhalla 1.8 e as invasões por Telnet e FTP foram gerados os *logs* conforme se pode inferir na Figura 19.

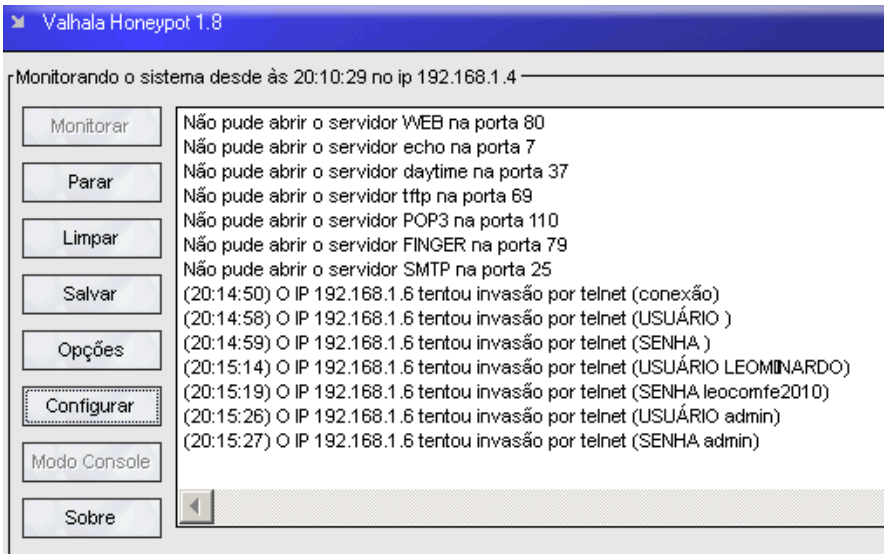

Figura 19 - Invasão por Telnet
Fonte: Autoria própria

A Figura 19 mostra a hora em que foi feita a invasão, bem como os comandos digitados pelo invasor.

Ademais, é mostrado no cabeçalho o IP (192.168.1.4) da suposta vítima e nos *logs* gerados é mostrado o IP (192.168.1.6) do suposto invasor.

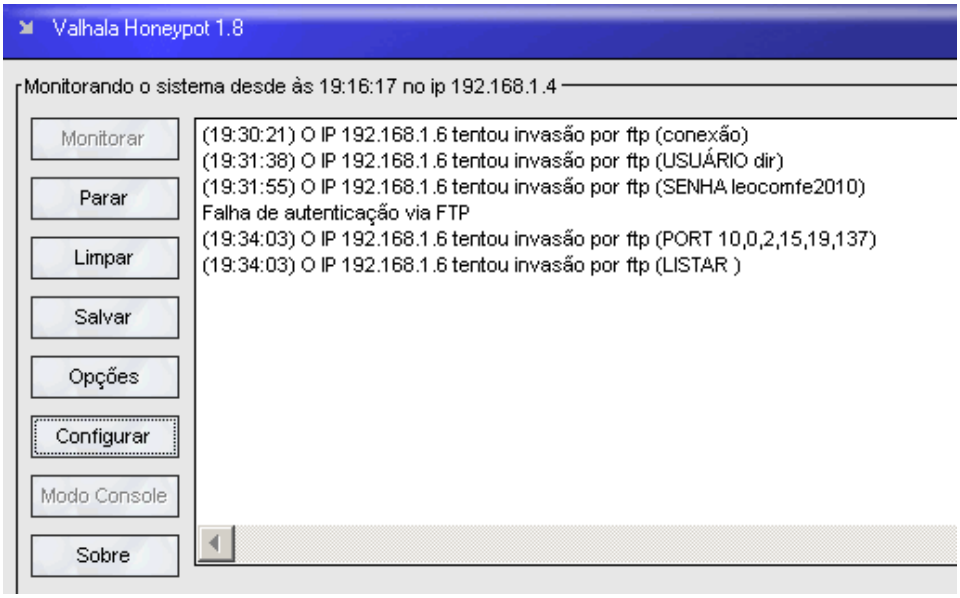

Figura 20 - Invasão por FTP
Fonte: Autoria própria

A Figura 20 nos mostra basicamente as mesmas descrições da Figura 19, no entanto pode se observar que neste momento o invasor está utilizando do protocolo FTP diferentemente do mostrado na Figura 19.

Durante a invasão não é emitido nenhum efeito sonoro por parte deste *honeypot*. Durante a execução do *worm Meajay,* o Valhalla não apresentou nenhum *log* de registro.

5.5 HONEYPOT KIPPO

O Kippo demonstra seus *logs* de detecção em formato texto ou de forma gráfica. Com a inicialização do Kippo começou-se a fazer o ataque por meio do protocolo SSH (*Secure Shell).* O SSH faz parte da suíte de protocolos TCP/IP que torna segura a administração remota de servidores do tipo Unix.

O SSH possui as mesmas funcionalidades do TELNET, com a vantagem da criptografia na conexão entre o cliente e o servidor.

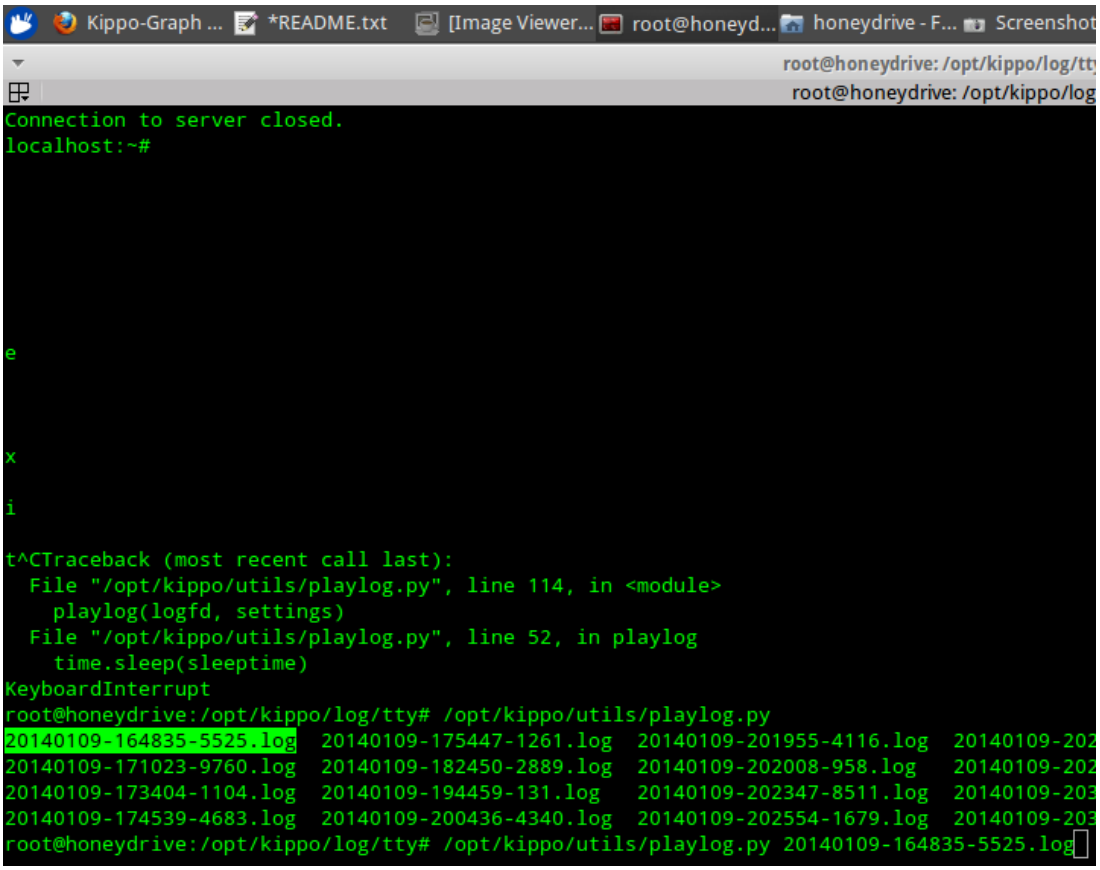

Figura 21 - Logs gerados pelo Kippo
Fonte: Autoria própria.

A Figura 21 apresenta a lista de *logs* gerados pelo Kippo durante a sua execução. Com o comando <*playlog.py*> e o número do *log* desejado, o Kippo demonstra os comandos executados pelo invasor durante o seu ataque no computador da vítima, como se fosse um vídeo.

Durante a invasão não é emitido nenhum som por parte do Kippo, Diferentemente do que ocorreu com os honeypots Kfsensor e Honeybot.

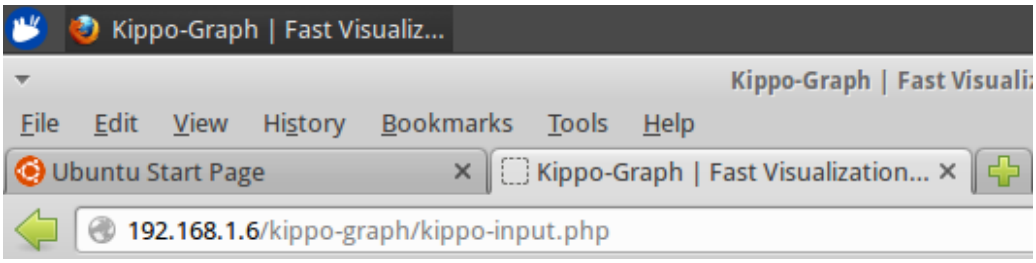

Figura 22 - Localhost do Kippo modo gráfico.
Fonte: Autoria própria.

Para acessar os logs do Kippo no modo gráfico, basta abrir o navegador e no campo da URL digitar o IP da máquina, neste caso 192.168.1.6 e acrescentar

</kippo-graph/kippo- input.php>. Conforme ilustrado na Figura 22. Na Figura 23 é apresentada a página inicial do Kippo – GEO.

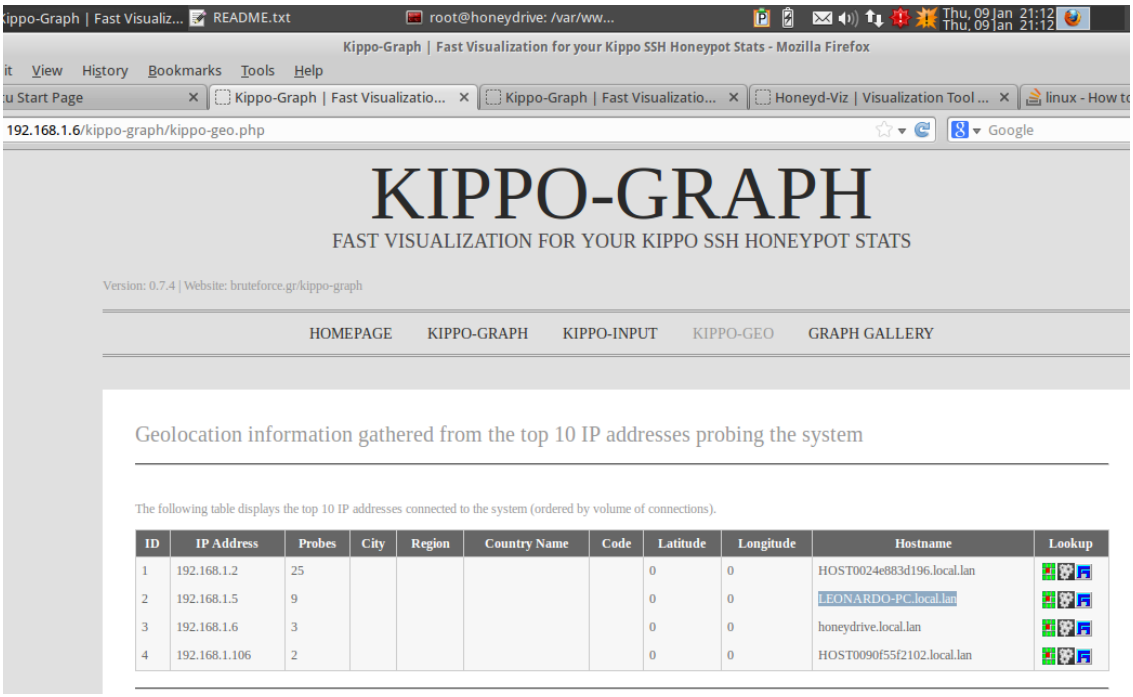

Figura 23 - Página Kippo - GEO.
Fonte: Autoria própria.

Essa ferramenta possibilita ao usuário saber a localidade, com as coordenadas geográficas do invasor; além de mostrar também, o IP correspondente a cada intruso.

A Figura 24 apresenta a quantidade de ataques feitos pelo IP correspondente.

ID	IP Address	Probes
1	192.168.1.2	25
2	192.168.1.5	9
3	192.168.1.6	3
4	192.168.1.106	2

Figura 24 - IP do Invasor e a quantidade de ataques
Fonte: Autoria própria.

Latitude	Longitude	Hostname	Lookup
0	0	HOST0024e883d196.local.lan	
0	0	LEONARDO-PC.local.lan	
0	0	honeydrive.local.lan	
0	0	HOST0090f55f2102.local.lan	

Figura 25 - *Hostname* dos invasores
Fonte: Autoria própria.

A Figura 25 ilustra a tabela gerada pelo Kippo com as coordenadas geográficas dos intrusos e seus respectivos nomes, exemplo: **LEONARDO – PC**, em destaque nessa figura.

A Figura 26 apresenta em forma de gráfico – barra a quantidade dos 10(dez) comandos que mais falharam durante a invasão. Cada qual com o respectivo comando utilizado e a quantidade usada pelo invasor; todos devidamente identificados pelo ID (identificador) gerado pelo Kippo, ideal para ser apresentado em reuniões.

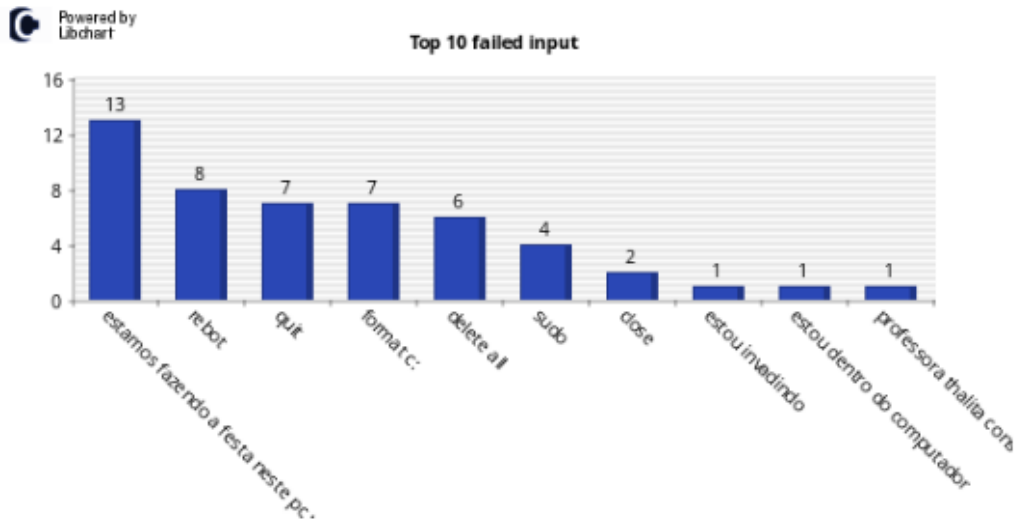

Figura 26 - Códigos que falharam em modo barra.
Fonte: Autoria própria.

A Figura 27 apresenta, em gráfico modo pizza; a quantidade, em porcentagem, de conexões realizadas por cada IP.

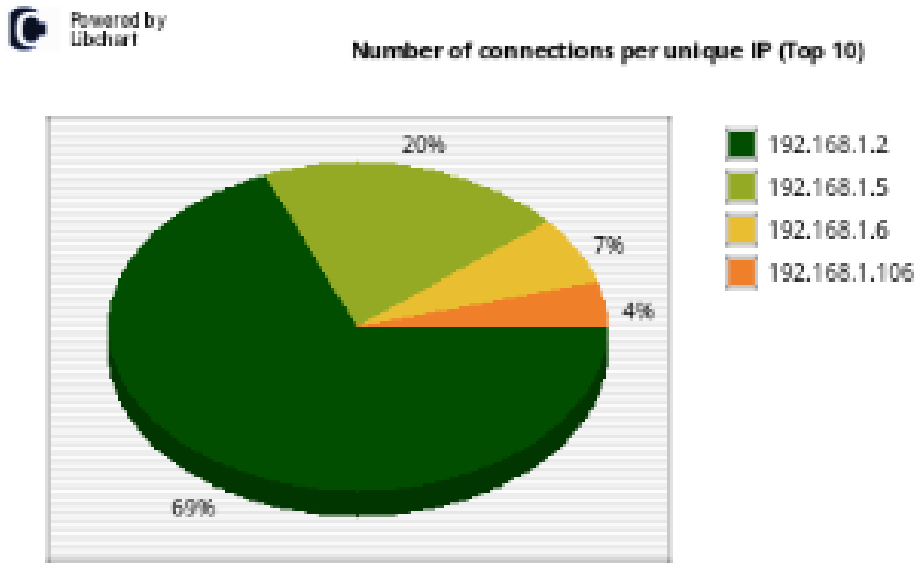

Figura 27 - Número de conexões por IP
Fonte: Autoria própria.

Conforme a Figura 27, verifica-se que o IP 192.168.1.2 acessou 69%, o IP 192.168.1.5 acessou 20%, o IP 192.168.1.6 acessou 7% e o IP que menos acessou foi o 192.168.1.106 com a porcentagem de acesso em 4%.

A Figura 28 ilustra a quantidade de invasões ocorridas no dia 09.01.2014.

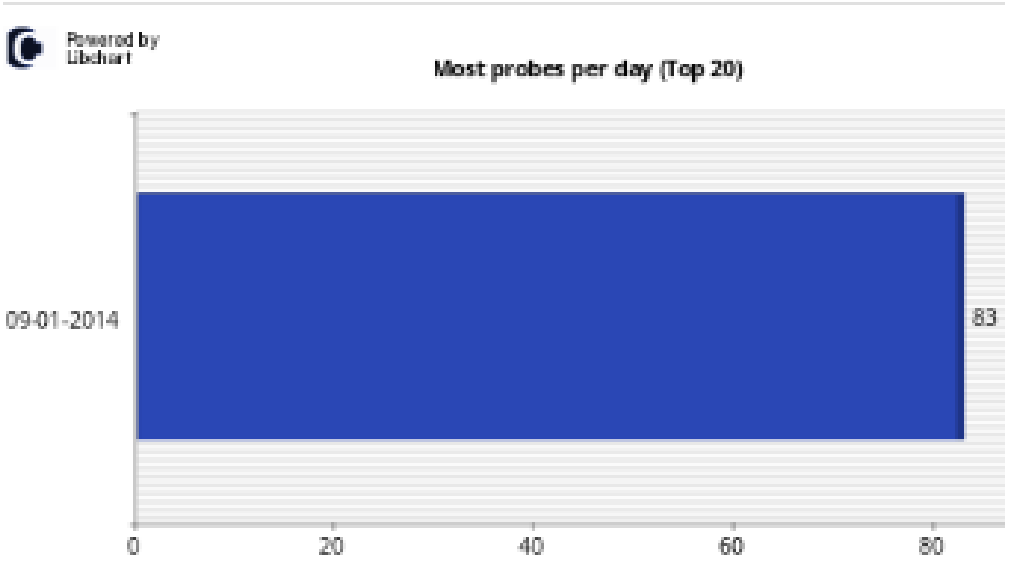

Figura 28 - Quantidade de invasões feitas no dia 09.01.2014
Fonte: Autoria própria.

A Figura 29 apresenta os 10 (dez) comandos mais realizados com sucesso, sendo: 9 (nove) com o comando *ls*; 5 (cinco) com o comando *cd ..;* 3 (três) com o comando *exit*; 2 (dois) com o comando *cd opt*; 1 (um) com o comando su e por fim 1 (um) com o comando *cat passwd*.

Verificam-se 83 (oitenta e três) invasões realizadas. Este gráfico importante gerado pelo Kippo permite saber a quantidade de ataques sofridos por dia.

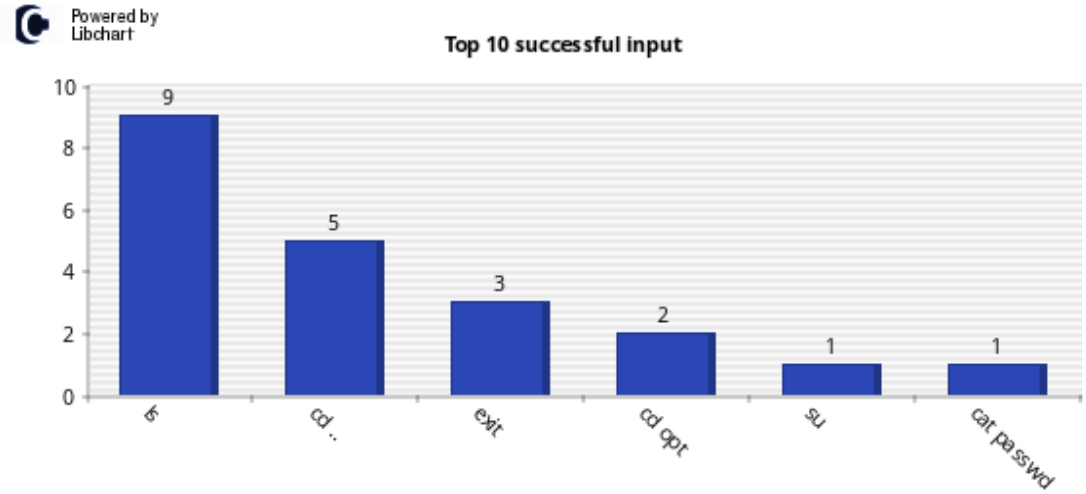

Figura 29 - Os 10 (dez) comandos mais realizados com sucesso.
Fonte: Autoria própria.

A Figura 30 apresenta a quantidade em porcentual de dados inseridos pelo invasor para tentar acessar a máquina da vítima. Sendo registrada em 56% a tentativa de se logar com o nome root e a senha 123456.

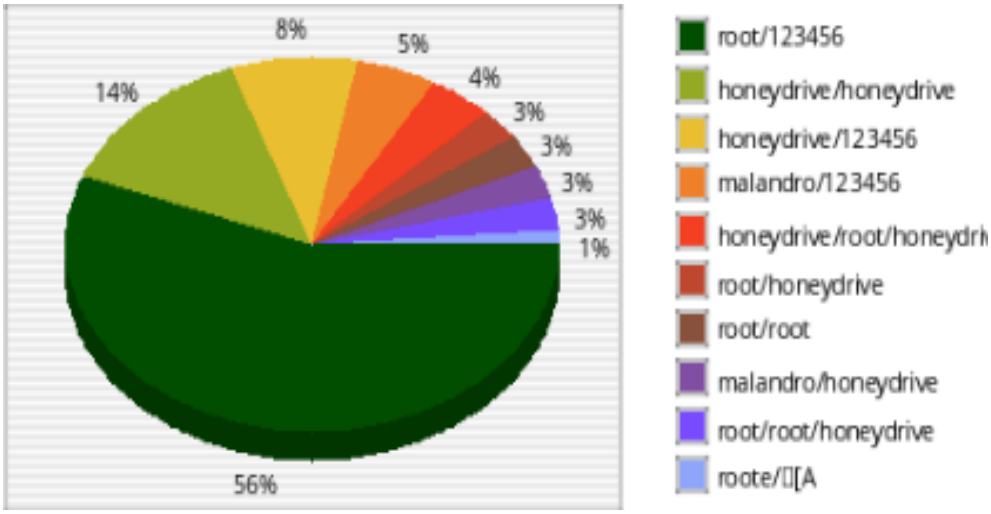

Figura 30 - Invasão com login e senha
Fonte: Autoria própria.

A Figura 31 apresenta a quantidade de invasões realizadas por diferentes tipos de Sistemas Operacionais.

Debian- 6 ficou com 35 (trinta e cinco) invasões, Windows 7-usando o Putty Release com 14(quatorze) invasões, Debian 5 com 3 (três) invasões e o Debian 3 com uma única invasão.

Por fim, a Figura 32 apresenta de forma decrescente, com base na data os comandos potencialmente mais lesivos à manutenção do Sistema, denominados pelo próprio Kippo como sendo comandos interessantes a serem analisados.

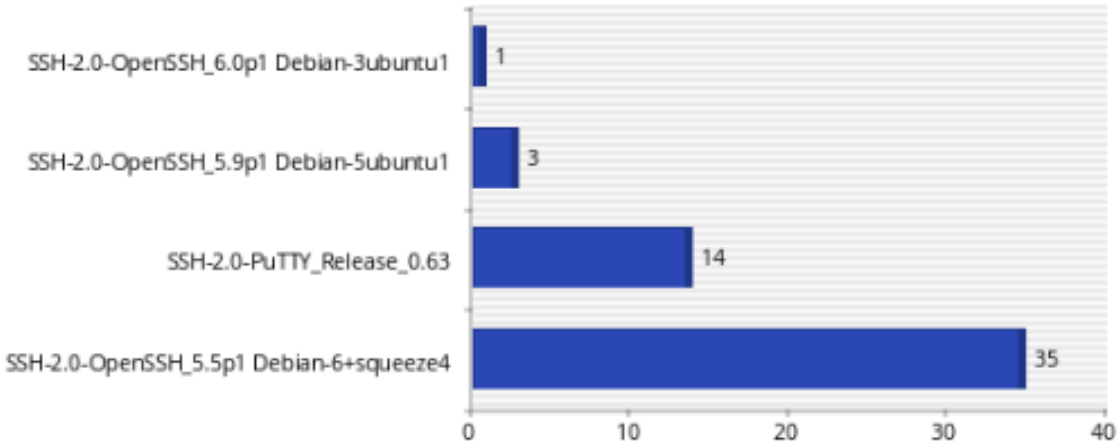

Figura 31 – Top 10 (dez) da quantidade de invasões por Sistema Operacional
Fonte: Autoria própria.

Interesting commands

The following table diplays other interesting commands executed by attackers in the honeypot system.

ID	Timestamp	Input
1	Thursday, 09-Jan-2014, 21:34 PM	ssh root@192.168.1.6
2	Thursday, 09-Jan-2014, 21:05 PM	scp /etc/passwd root@10.0.2.15:/home/goodspeed
3	Thursday, 09-Jan-2014, 20:55 PM	scp
4	Thursday, 09-Jan-2014, 20:54 PM	scp /etc/passwd root@192.168.1.6:/home/goodspeed
5	Thursday, 09-Jan-2014, 20:36 PM	adduser macaco
6	Thursday, 09-Jan-2014, 20:35 PM	adduser
7	Thursday, 09-Jan-2014, 20:26 PM	cat
8	Thursday, 09-Jan-2014, 20:04 PM	cd dev
9	Thursday, 09-Jan-2014, 19:58 PM	ssh malandro@192.168.1.6
10	Thursday, 09-Jan-2014, 19:46 PM	mv dev
11	Thursday, 09-Jan-2014, 17:55 PM	cat passwd-
12	Thursday, 09-Jan-2014, 17:51 PM	ping 192.168.1.6
13	Thursday, 09-Jan-2014, 17:36 PM	ifconfig
14	Thursday, 09-Jan-2014, 17:10 PM	cat passwd

Figura 32 - Comandos que podem prejudicar o Sistema
Fonte: Autoria própria.

6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Honeypots são ferramentas que podem ser utilizadas como estratégia de segurança de redes de computadores. Isso acontece porque são softwares que simulam serviços, atraem atacantes, coletam dados e alertam o administrador da rede que existem anomalias acontecendo na rede.

Como resultado deste trabalho, tem-se a comparação dos quatro honeypots selecionados, bem como a maneira que cada um se comporta durante a invasão de um agente malicioso.

De acordo com os testes realizados, demonstrou-se de forma detalhada e o quanto pode ser útil a escolha correta de qual honeypot escolher em cada caso concreto. Em se tratando do Honeybot e o Kfsensor, ambos foram os únicos a emitirem sons no decorrer das invasões. Uma das desvantagens do Kfsensor, utilizado, é de se tratar de uma ferramenta *trial*, ou seja, expira em 30 (trinta) dias e a sua complexidade em detrimento das suas diversas funções.

Levando-se em consideração os *scripts* gerados em todos os *honeypots* testados, o Kippo se destacou pela sua geração de gráficos em diversos modos e em situações diferentes, tratando-se do honeypot mais completo em relação aos logs. A sua única desvantagem a não emissão de som durante a invasão; obrigando, portanto, que os *logs* sejam checados por um determinado período de tempo.

Quanto ao Valhala 1.8, este é de fácil instalação e bem simples de se manusear, no entanto sua simplicidade o deixa a desejar quando se trata da necessidade de registros mais robustos e complexos.

Em relação à detecção do *worm* escolhido, o único a apresentar a detecção durante os testes foi o Honeybot.

Em suma, conclui-se que o honeypot que se demonstrou mais eficiente no que tange a geração de logs foi o Kippo, uma vez que suas ferramentas de geração de gráfico proporcionam ao administrador da rede uma visualização detalhada dos comandos de invasão. Apesar das qualidades elencadas sobre o Kippo, foi o Honeybot o único a detectar o worm Meajay, de forma a ser ele o honeypot mais indicado em se tratando de detecção deste worm estudado.

6.1 TRABALHOS FUTUROS

Os testes efetuados e as ferramentas examinadas neste trabalho não cobriram todos os tipos de *honeypots*. Assim, sugerem-se como trabalhos futuros relacionados a esta área de detecção de tráfego malicioso em rede de computadores, a utilização de outros tipos *honeypots* aqui não abordados.

Outra sugestão seria a possibilidade da utilização de outros *honeypots* que sejam compatíveis com outros sistemas operacionais, em destaque ao MacOS e o Android, devido ao aumento nas vendas de *smartphones,* que utilizam estes sistemas operacionais*,* sem a devida proteção nos dias atuais.

7 REFERÊNCIAS

ANDRADE L. A. A. **Avaliação experimental do uso de honeypots de baixa interatividade para detecção de ataques em redes de computadores.** Disponível em<http://pop-es.rnp.br/pd/2009_09-ProjetoGraduacao-AndradeLeonardo.pdf>. Acessado em 23 dez 2012.

ALMUTAIRI, A; PARISH, D; PHAN, R. **Survey of High Interaction Honeypot Tools: Merits and Shortcomings.** Disponível em < http://www.cms.livjm.ac.uk/pgnet2012/Proceedings/Papers/1569604821.pdf >. Acessado em 12.jun.2013.

ASHOOR, A.S; GORE, S. **Importance of Intrusion Detection System (IDS).** Disponível em < http://www.ijser.org/researchpaper%5CImportance_of_Intrusion_Detection_System.pdf >. Acessado em 05.jul.2013

ASSUNÇÃO, M.F. **Valhala Honeypot - Free OpenSource Honeypot for the Windows system**. Disponível em <http://valhalahoneypot.sourceforge.net/>. Acessado em 04.jul.2013.

BATTISTI, Júlio Cesar Fabris. **Tutorial de TCP/IP – Parte 17 – IFC – Internet Firewall**
Conection (Windows XP). Disponível em <http://www.linhadecodigo.com.br/artigos.asp?id_ac=651&pag=1>. Acesso em: 03.dez.2013.

BARRETT, D.J; SILVERMAN, R.E; BYRNES, R.G. **SSH, the Secure Shell The Definitive Guide.** 2005. Disponível em < http://basie.exp.sis.pitt.edu/~christomer/lis2600/readings/SSH_Second_Edition.pdf >. Acessado em 23.jul.2012.

BIZSYSTEM, 2009. **LaBrea::Tarpit SUMMARY.** Disponível em <http://scans.bizsystems.net/paged_report.plx>, Acesso em 15. Jul.2009.

BROWN, S; LAM, R; PRASAD, S; RAMASUBRAMANIAN, S; SLAUSON, J. **Honeypots in the Cloud.** Disponível em < http://www.joshslauson.com/pdf/cs642_project.pdf >**.** Acessado em 20.set.2013

BORGES; Ciro Fernando Preto,BENTO**;** Paulo Diego Nogueira ;**Segurança De Redes Utilizando Honeypot**.Belém 2006.

BURDINO, Luis Claudio; **Honeypots como Estratégia de Segurança para Redes.** Brasília.2006.

BRUTEFORCE, **BruteForce Lab's Blog security, programming, administration, visualization, virtualization.** Disponível em < http://bruteforce.gr/kippo-graph>. Acessado em 15. Out. 2013.

CAMPOS, V. R. A. **Estudo comparativo de linguagens de programação**. Salvador, 2003.Disponível em: <http://twiki.im.ufba.br/pub/MAT052/RelacaoMonografias/monografia.doc>. Acesso em: 03 nov 2013.

DAMATTO, F.C; RALL, R; **ESTUDO DOS POSSÍVEIS MOTIVOS DO AUMENTO DE INCIDENTES DE MALWARES NAS EMPRESAS.** Disponível em <http://www.fatecbt.edu.br/seer/index.php/tl/article/download/107/66>.Acessado em 24.nov.2013

FRANCO, Lúcio H; BARATO, Luís G; Montes, Antonio. **Instalação e Uso de Honeypot de Baixa Interatvidade –** 17 ª Reunião do Grupo de Trabalho em Engenharia de Redes – GTER 17 -2004- on line- HTTP://eng.resgistro .br/gter17/vídeos, acessado em 20.jul.2013.

FRANCO, L. H. and MONTES. A. **Desvio de Tráfego Malicioso Destinado a redes de Produção para uma Honeynet**. In Grupo de Trabalho em Segurança de Redes, Rio de Janeiro, 2003.

GOMES, C.S; JIN, N.K; CREPALDI T.F, **Máquina Virtuais Java e .NET.** Disponível em <http://www.ic.unicamp.br/~rodolfo/Cursos/mc722/2s2007/trabalhos/g04_texto.pdf> .Acessado em 20. Jul.2013.

GOMES, A.S; MEDEIROS, F.P. A; ARAÚJO, T.S, VASCONCELOS, B.Q; ALBUQUERQUE, F.A; Paiva, P.V. **Implantação de um Modelo de Monitoria Virtual Suportado por Softwares Livres.** Disponível em <>.Acessado em 12.12.2013

HOEPERS, Cristine; JESSEN, Klaus Steding; MONTES, Antônio. Projeto Honeypots Distribuidos. Disponível em: <http://www.honeynet.org.br/presentations/hnbr-gts2003-slides.pdf> Acesso em 10 de Dezembro de 2013.

JESSEN, Klaus Steding; CHAVES, Marcelo H. P. C. Implantação de Honeypots de Baixa Interatividade com Honeyd e Nepenthes. Disponível em: <http://www.cert.br/docs/palestras/certbr-campusparty2008-2.pdf> Acesso em 14 de Dez de 2013.

KIPPO, K**ippo SSH Honeypot. Disponível em <** https://code.google.com/p/kippo/**>.** Acessado em 12. dez. 2013

KUMAR J**;** Yogendra, SINGHI**,** Surabhi; **Honeypot based Secure Network System**.Fevereiro de 2011.

LAUDON, K.C; LAUDON, JP. **Sistemas de informações gerenciais**. 7. ed. São Paulo: Pearson Prentice Hall, 2007

LUCENA, S.C; MOURA, A. S. **Detecção de Anomalias Baseada em Análise de Entropia no Tráfego da RNP.** Disponível em < http://www.lbd.dcc.ufmg.br/colecoes/wgrs/2008/012.pdf >. Acessado em 08.jun.2013

MARCELO, A; PITANGA, M. **Honeypots: A arte de iludir hackers**. Rio de Janeiro: Brasport, 2003.

MOREIRA, F. L. A**. ANÁLISE DE TRÁFEGO EM REDES TCP/IP NA DETECÇÃO DE INTRUSÃO.** Disponível em < http://fatecsjc.edu.br/trabalhos-de-graduacao/wp-content/uploads/2012/03/BDR1_flavio2009.pdf >. Acessado em 20.jun.2013.

NED, Frank. **Introdução a IDS.** Disponível em <http://www.rnp.br/newsgen/9909/ids.html>. Acesso em: 05. Dez.2013.

ORACLE, 2013. **User Manual**. Disponível em < https://www.virtualbox.org/manual/UserManual.html >. Acessado em 12.dez.2013

RUVALCABA, Cristian. **SMART IDS – HYBRID LABREA TARPIT.** 2009**.** Disponível em <http://www.sans.org/reading-room/whitepapers/casestudies/smart-ids-hybrid-labrea-tarpit-33254**>**.Acessado em 20.jul.2013.

SANTANNA, João. **Notas de aula – Unidade 1.2 – Ataques de Segurança – Modelo de Segurança em redes**. IESAM, 2006.

SARAFIK, J; REZAK, F; VOZNAK, M. **Monitoring of Malicious Traffic in IP Telephony Infrastructure.** Disponível em <http://www.cesnet.cz/wp-content/uploads/2013/02/ip-telephony-malicious-traffic-monitoring.pdf>.Acessado em 24.dez.2013

SHADOWSERVER**. What is malware?.** Disponível em <http://www.shadowserver.org/wiki/pmwiki.php/Information/Malware>. Acessado em 24.jul.2013

SHION, Dark. **Honeypot: Aprendendo com o intruso.** 2011. Disponível em < http://mafialinux.forumer.com/honeypot-aprendendo-com-o-intruso-t2299833.html >. Acessado em 12. Ago.2013.

SILVA, C.S; ROSA, A.C.M; CHAIM, D.F; CARVALHO, R.J; CHIMENDES, V.C.G.**ENGENHARIA SOCIAL: O ELO MAIS FRÁGIL DA SEGURANÇA NAS EMPRESAS.** Disponível em < http://www.revistas.udesc.br/index.php/reavi/article/view/2840/2172>. Acessado em 14.jul.2013

SILVA, Alexandre W. B. da e SILVA, Paulo H. de M. **Estudo Analítico e Comparativo de Honeypots**. 2010.93f. Trabalho de Conclusão de Curso (Tecnologia em Redes de Computadores)- Faculdade de Tecnologia de São José dos Campos, São José dos Campos, 2010.

SILVA, P. F. **Protótipo de software de segurança em redes para a monitoração de pacotes em uma conexão TCP/IP**. 2001. 112 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SILVEIRA, C.B.V. **UM ESTUDO DAS NECESSIDADES DAS MICROEMPRESAS EM GARANTIR A INTEGRIDADE DOS SEUS DADOS DIGITAIS.** Disponível em <http://www.lume.ufrgs.br/bitstream/handle/10183/26185/000752420.pdf?sequence=1>.Acessado em 23.set.2013.

SOARES, L.F.G; LEMOS, G; COLCHER, S; **Rede de computadores: das LANs MANs e WAMs às redes ATM**. Rio de Janeiro: Campus, 1995.

SPITZNER, Lance. **Honeypots: Tracking Hackers**. Ed.Person, 2002.

SPITZNER, Lance. Honeytokens: **The Other Honeypot**.Disponível em<http://www.securityfocus. com/infocus/1713>, acessado em 21. Jul. 2013.
TANEMBAUM, Andrew S. **Redes de computadores**. Tradução: VANDENBERG, D. Souza. Rio de Janeiro: Elsevier, 2003 – 15ª reimpressão.

STALLINGS, W. **Redes e sistemas de comunicação de dados: teoria e aplicações corporativas**. Rio de Janeiro: Elsevier. 2005

STIPEK, Debora. J. **Motivation to learn: from theory to practice**. Englewood Cliffs: Prentice Hall, 1993.

SYMANTEC, 2009. Disponível em <http://www.symantec.com>, Acesso em 13.ago.2013

RAO, S.S; 1, HEGDE, V; MANEESH,B; PRASAD,J.N.M; Suresh, S. **WEB BASED HONEYPOTS NETWORK.** Disponível em < http://www.ijsrp.org/research-paper-0813/ijsrp-p2078.pdf >. Acessado em 02.dez.2013

Ryan, R. M., Deci, E. L. (2000). **Intrinsic and Extrinsic Motivations**: Classic Definitions and New Directions. Contemporary Educational Psychology, 25, 54–67.

TANEMBAUM, Andrew S; WETHERALL, D. **Redes de computadores.** Tradução Daniel Vieira.São Paulo: Perarson Prentice Hall. 5 ed 2011.

TANEMBAUM, Andrew S; WOODHULL, Albert S. **Sistemas operacionais: Projeto e implementação**.Tradução: Edson Furmankiewicz.2.ed.Porto Alegre-RS: Bookman, 2000.

TRENTIN, M.A; LINDEN, G.S; JÚNIOR, A.A.S.C; Fávero, A.L. **Proposta de Implementação de uma Honeypot para Detecção de Vulnerabilidades.** Disponível em < http://www.sirc.unifra.br/artigos2002/artigo12.pdf>. Acessado em 21.jun.2013.

TURBAN, E; LEIDNER, D; MCLEAN, E; WETHERBE, J. **Tecnologia da informação para gestão – transformando os negócios na economia digital**. 6.ed. Porto alegre: Bookman, 2009.