SISTEMA DE DETECÇÃO DE INTRUSOS PARA IDENTIFICAR ATAQUES DE BOTNET EM COMPUTAÇÃO EM NUVEM.

Lúcio André Costa Cruz Marcos Gomes da Silva Rocha Raimundo Pereira da Cunha Neto

andre@facema.edu.br , marcosgsrocha@bol.com.br, netocunhathe@gmail.com

Resumo

Nos últimos anos o uso da internet está aumentando a cada dia, as organizações iniciaram o uso de suas aplicações na Computação em Nuvem, com a grande vantagem oferecida pelos provedores de serviços, devido ao poder de processamento das nuvens, além do baixo custo benefício. Em virtude ao crescimento do serviço de Computação em Nuvem mundialmente, isto vem provocando grandes ameaças as organizações tornando alvo de ataques cibernéticos, entre os vários tipos de ataques, a Botnet vem se tornando uma das maiores ameaça, com sua capacidade de controle e da potencialidade em ataques distribuídos. Com isso essas empresas ficam refém dos mecanismos de defesas oferecidos pelo serviço adquirido, que atualmente vem sendo necessária a ampliação de novas técnicas de segurança. Esse trabalho apresenta um modelo de (IDS) Sistema de Detecção de Intrusos em computação em nuvem, que visa na ampliação de detectores de Botnet através da utilização de ataques por assinatura em rede, analisando e comparando assinaturas dos pacotes que trafegam em uma rede, propondo uma tecnologia de detecção baseada em assinatura.

1. INTRODUÇÃO

A TI (Tecnologia da Informação) está crescendo rapidamente e as empresas utilizam este crescimento para melhorar nas atividades e operações da sua estrutura organizacional visando o potencial de mercado. A Computação em Nuvens ou *Cloud Computing* é uma tecnologia que se baseia na computação atual que vem recebendo a cada dia grandes investimentos de empresas. Uma das principais vantagens é a virtualização de serviços e produtos computacionais, obtendo uma redução de alto custo com infraestrutura local, ganhando praticidade e comodidade em seus serviços prestados com rapidez. No entanto a utilização desse novo paradigma contribui com a evolução de novas tecnologias exclusivamente nas redes de computadores seja em ambientes domésticos ou em corporações. Isso provocou um grande desafio à segurança e privacidade da informação, fazendo com que inúmeras formas de ataques surjam a cada dia.

Entre estes ataques, as *Botnets* se destacam pela sua forma de realizar ataques distribuídos e provocando impacto em redes pelo mundo, desta forma as *Botnets* podem realizar um grande estrago em uma plataforma de computação em nuvem.

Então Como garantir que dados e informações estarão protegidos e preservados? Será que podemos confiar nas empresas que fornecem a Computação nas Nuvens? E o sigilo dos dados? Nota-se que a grande preocupação é com a Segurança da Informação contida em Nuvem.

O artigo esta organizado em cinco partes, na segunda seção descrevemos Computação em Nuvem. Na terceira seção sobre ataques de BOTNET, os Sistemas de Detecção de Intrusos são tratados na quarta seção. Já na quinta seção são explorado o Modelo Proposto, sendo na sexta seção Testes e os Resultados avaliados na seção seguinte. Para finalizar a Conclusão na seção oito.

2. COMPUTAÇÃO EM NUVEM

A computação em nuvem é um modelo que permite acesso onipresente e conveniente à rede, sob demanda a um pool compartilhado de recursos computacionais configuráveis que podem ser rapidamente provisionados e liberados com um esforço mínimo de gestão ou interação com o provedor de serviços.

3. BOTNET

A *Botnet* é uma rede de computadores comprometidos, denominados de *Bot*, sob o controle remoto de um operador humano, o "*Botmaster*". O termo "*Bot*" é derivado da palavra *"Robot"*. Os *Bots* são dispositivos hospedeiros projetados para executar algumas funções pré-definidas de forma

automatizada, que permite ao *Botmaster* controlar remotamente as ações de ataque.
Ataques por *Botnet* são comuns e incluem o lançamento de ataques distribuídos de negação de serviço (DDoS), enviando *e-mails* de *spam* e espalhar *malware*, furto de informações privadas e realizar a fraude do clique[1].

3.1. Ataque em Computação em Nuvem

Ataque de zumbis: Através da Internet, o atacante tenta inundar a vítima através do envio de pedidos de hosts inocentes na rede. Estes tipos de hospedeiros são chamados de zumbis. Na nuvem, os pedidos de máquinas virtuais (VMs) são acessíveis por cada usuário através da Internet. Um atacante pode inundar o grande número de solicitações por zumbis. Tal ataque interrompe o comportamento esperado da nuvem e afetar a disponibilidade de serviços em nuvem. A nuvem pode ser sobrecarregada para atender a uma série de pedidos, e, portanto, esgotado, o que pode causar DoS (*Denialof Service*) ou DDoS (Negação de Serviço Distribuída) para os servidores. Fazendo com que vários pedidos de requisições de serviços de um atacante, a nuvem não possa atender a estas solicitações de um usuário válido. (MUNIR, PALANIAPPAN, 2012).

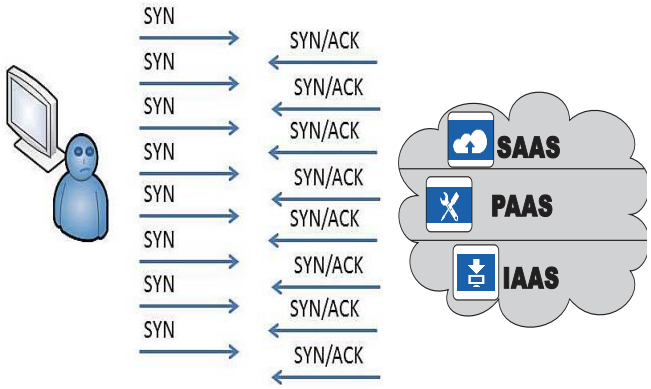

Figura 01: Ataque em computação em Nuvem.

4. SISTEMA DE DETECÇÃO DE INTRUSOS

4.1 Detecção de Intrusos

Detecção de Intrusão é o processo de monitoramento dos eventos que ocorrem em um sistema de computador ou rede e analisá-los para sinais de intrusos, como entrada não autorizada, atividade, ou de modificação do arquivo. Existem três fases do processo de detecção de intrusão, que são os seguintes:

- Monitoramento e análise de tráfego;
- Identificar as atividades anormais;
- Avaliação da gravidade e levantar alarme.

4.2 TRABALHOS RELACIONADOS

Algumas IDS famosas não possuem o poder de detectar *Botnets*, isso pelo fato destas possuírem características dinâmicas e também por serem constantemente melhoradas por desenvolvedores mal intencionados, para se tornarem mais resistentes contra detecção e técnicas de depuração [2]. Novas técnicas de detecção estão sendo desenvolvidas, a seguir vamos ressaltar algumas delas.

Snort é um IDS *open-source* que faz monitoramento no tráfego de rede para encontrar sinais de invasão. Que contém um conjunto de regras ou assinaturas para registrar o tráfego que seja suspeito utilizando técnicas baseadas em assinaturas que pode ser usada para detecção de ataques conhecidos, portanto esta não é uma solução funcional para ataques desconhecidos na computação em nuvem.

Botsniffer é um NIDS baseado em detecção de ataque que explora a correlação espaço-temporal e similaridade e detecta qualquer suspeito C&C, emitindo o alerta. Utiliza várias relações e análise de comparações para analisar o tráfego da rede. Foi projetado para detectar *Botnets* usando o protocolo IRC ou HTTP, mas pode ser facilmente estendido para incluir outros protocolos. Resultados demostram que o *Botsniffer* pode detectar *Botnets* com alta precisão e com uma taxa muito baixa de falsos positivos, tem a capacidade de detectar *Botnet* C&C, mesmo quando existe apenas um *bot* na rede em que está sendo monitorada. Porém seu mecanismo de relação gera relatórios de forma precisa e concisa em vez de produzir alertas de eventos maliciosos como um IDS tradicional faz. É aplicável a um conjunto restrito de *Botnets*, não se aplicando a estrutura de *Botnets* descentralizadas[3].

Já o método de detecção da ferramenta *BotGAD* centraliza-se no comportamento da ação do atacante, ou seja, do *Botnet*, e não nos dados ou em assinaturas do tráfego. Com a grande possibilidade de detectar *Botnets* em redes de banda larga em tempo real, deste que trabalhando com uma pequena quantidade de dados. Como os *Bots* frequentemente usam DNS para reunir hosts infectados, para lançar ataques e atualizar seus códigos, o modelo proposto do *BotGAD* é exibir os

nomes de domínio dos C&C desconhecidos e os endereços IP de hosts infectados. Lembrando que as *Botnets* frequentemente usam DNS para a sua ação. A detecção de *Botnet* baseada em DNS vem ser uma das mais promissoras, pois detecta redes de *Bots* independentemente de sua estrutura[4].

BotMiner tem como objetivo detectar grupos de máquinas comprometidas dentro de uma rede de monitorização que são parte de uma *Botnet*. Fazemos isso por meio da análise passivamente com o tráfego da rede monitorada. Note que nós não pretendemos detectar *Botnets* no exato momento em que as máquinas das vítimas são comprometidas e infectadas com *malware* (*bot*) de código. Em muitos casos, estes eventos podem não ser visíveis para passivamente monitorar o tráfego de rede. Por exemplo, um computador portátil já infectado pode ser ligado à rede monitorada, ou um utilizador pode clicar em um *e-mail* malicioso e fica infectado. Neste trabalho, nesse caso a preocupação é com a forma como os hosts internos são infectados (por exemplo, por *e-mail* anexos maliciosos, remoto aproveitando-se de um *download*). Esse modelo se concentra na detecção de grupos de máquinas já comprometidas dentro da rede monitorada que fazem parte de uma *Botnet[5]*.

5 MODELO PROPOSTO

Este capítulo apresenta a proposta de uma arquitetura para um Sistema de Detecção de Intrusos Baseado em Ataques de *Botnet* na situação de Negação de Serviço. Uma visão geral da arquitetura e do seu funcionamento será apresentada, bem como as principais interações entre os módulos. Ao final do capítulo, a integração da arquitetura proposta do Sistema será apresentada.

5.1 Arquitetura Proposta

A arquitetura proposta utiliza um modelo de detecção que possibilita a identificação de possíveis ataques (C2IDS), assim, condensa técnicas de coleta e análise baseada em assinatura em pacotes que trafegam em uma nuvem. Conforme descrito na Figura 02, a C2IDS tem o poder de detecção de intrusos incrementado, pois detectam ataques previamente conhecidos e identificado na assinatura.

Figura 02: Modelo Proposto

O modelo proposto inicia-se com a coleta de pacotes pelo sensor de monitoramento da nuvem. Após a coleta dos dados, os pacotes são armazenados em uma base de dados de coleta para posteriormente serem analisados por detecção baseada em assinatura, com a finalidade de identificação dos ataques na nuvem. Assim, caso algum intruso seja encontrado, a informação é enviada para um agente de reação, que atua de forma passiva, ou seja, gera uma informação para o administrador da nuvem sobre o ataque detectado, caso contrário, os pacotes são descartados pelo C2IDS, para reduzir a base de dados, não ocupando tanto espaço na nuvem.

5.1.1 Sensor de Monitoramento

A função do sensor de monitoramento é capturar os pacotes da nuvem. Atuando de forma passiva, trabalhando de forma ágil, assim não interferindo no desempenho e no tráfego da nuvem.

5.1.2 Análise por Assinatura

Na análise por assinatura, o sistema C2IDS utilizando o paradigma de comparação de assinaturas com os pacotes que foram coletados diretamente na nuvem. As assinaturas coletadas são armazenadas em uma base dados de assinatura, onde ao serem comparadas com os pacotes coletados pelo sensor de monitoramento para identificação dos ataques, caso seja positivo o resultado é encaminhado para o agente de reação ou o pacote é descartado.

5.1.3 Agente de Reação

O agente de reação funciona de forma passiva, esperando a análise dos pacotes serem concluída, para que assim possam alertar o administrador da nuvem sobre algum tipo de ataque encontrado. Sendo responsável por tomar

contramedidas caso um incidente de segurança seja detectado. Com base na avaliação da coleta de pacotes e análise de assinaturas.

A questão mais importante é a velocidade de notificação quando os ataques ocorrem na rede[6]. O C2IDS também gera uma reação ativa quando ocorrem ataques e resposta a situações críticas. As contramedidas ativas ocorrem através do bloqueio do sinal de um invasor ou na criação de arquivo log.

5.1.4 Base de Dados

Base de Dados é um banco de dados que contêm os endereços de IP, nomes de domínio, e outros dados que permitem reconhecer se esse endereço é malicioso ou não. E com isso, informa o agente de reação uma ação, seja negar o seu acesso, caso for identificado como suspeito. É possível também, detectar a origem do pacote e bloqueia-lo, para impedir o seu tráfego na rede. Entre as bases de dados existentes, podemos descrever a base de dados de coleto, que contém as informações dos pacotes coletados, as bases de dados das assinaturas, que descreve quais assinaturas existem dentro da C2IDS e a base de dados de reação, onde estão as informações sobre as medidas preventivas do IDS.

5.2 Implementação

A solução emprega a detecção de *Botnet* por uso estratégico de análise por assinaturas, visto que os trabalhos relacionados estudados utilizam técnicas de diferentes tipos para identificar ataques de *Botnet* em nuvem. Este monitoramento baseia-se em informações coletadas do tráfego de nuvem capturadas por modo promiscuo. A implementação utilizada no C2IDS é descrita através do diagrama de atividades.

5.2.1 Diagrama de atividade

O diagrama de atividade é projetado com objetivo de simplifica à visão do que acontece durante uma operação ou processo dentro de um sistema.

A modelagem de atividades tem como proposito a execução e fluxo do comportamento de um determinado sistema. Quando se usa para a modelagem de software, normalmente as atividades representam um comportamento que resulta na chamada de um método. Quando se usa para modelagem de negócios, as atividades podem ser desencadeadas por eventos externos, bem como uma ordem a serem determinadas ações desenvolvidas[7].

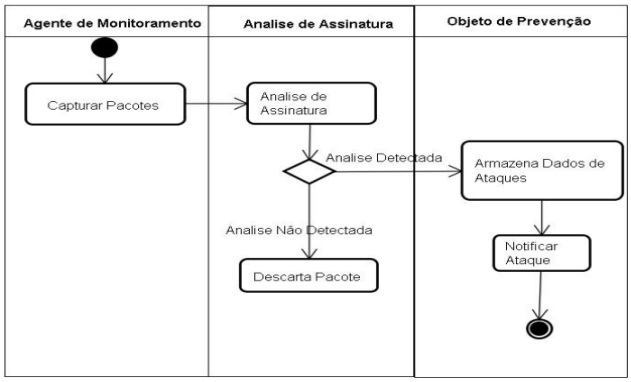

Figura 03: Diagrama de Atividades do C2IDS

A figura 03 demostra o processo de Análise por Assinatura, aonde o agente verifica se algumas das assinaturas de ataques de *Botnet* foram identificadas, a partir deste ponto será decidida ação segundo ponto de decisão. Caso uma assinatura seja encontrada é encaminhada uma notificação ao agente de reação, onde o mesmo tomará as medidas em notificar o ataque encontrado, finalizando assim o processo existente. Caso contrário, o agente de assinatura não detecte o ataque nos pacotes, esses serão descartados para obter um melhor desempenho nos próximos processos.

5.2.3 Protótipo

O protótipo do C2IDS possui uma interface simples para o usuário, na figura 05. Os componentes construídos no modelo foram desenvolvidos em Java [8], proporcionando características de portabilidade em várias plataformas de sistemas operacionais.

Figura 04: Tela de cadastro de Host

Foi utilizado também o método de análise e captura de pacotes para plataforma *Windows,* com a utilização da ferramenta *Cacti* que tem como objetivo administra redes, que recolhe e exibe informações sobre o estado de uma rede de computadores através de gráficos. Permitindo o monitoramento e gerenciamento de redes simples até redes complexas, com centenas de dispositivos. Esta ferramenta foi desenvolvida para ser flexível de modo a se adaptar facilmente a diversas necessidades, monitora o estado de elementos de rede e programas bem como largura de banda utilizada e uso de CPU. Com o apoio de *RRDTool* também pode produzir gráficos que permitem ter uma ideia visual dos dados armazenados, os quais podem ser utilizados ou exibidos por outros sistemas. Podendo assim ter total controle de monitoramento e armazenamento de dados que trafegam em uma nuvem.

Conforme descrito na figura 06, o C2IDS tem o poder de detecção de intrusos em todos os tipos de nuvem, pois detectam ataques conhecidos e não conhecidos em todas as camadas de nuvem.

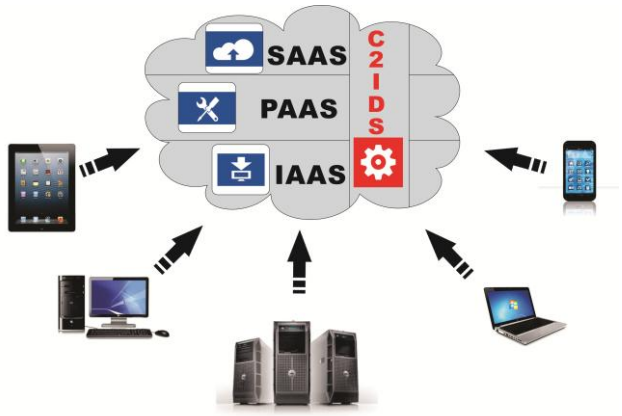

Figura 05: C2IDS – *Cloud Computing Intrusion Detection System*

6. TESTES E RESULTADOS

Neste capítulo, é apresentada a implementação de um protótipo para o modelo proposto na pesquisa. Um mecanismo de detecção de intrusos foi implementado, usando técnicas de análise a ocorrência de *Botnets*.

Para finalizar, a avaliação da solução proposta é apresentada. Para que esse mecanismo de detecção de intruso pudesse funcionar, foi realizado um treinamento com registros de conexões normais e conexões sob ataques de *Botnet* tanto em ambiente simulado, quando em ambiente da nuvem desenvolvida. Os dados obtidos com essa implantação são discutidos no decorrer do capítulo e servirão de base para avaliar a eficiência do sistema.

6.1 Cenário

Durante o desenvolvimento dessa ferramenta foi utilizado o *software VMware* que é uma ferramenta poderosa quando se fala em máquina virtual industrial que contém três níveis de VM produtos: *VMware Workstation, VMware GSX Server* e *VMware ESX Server*[9]. Neste trabalho, vamos nos concentrar no produto *VMware Workstation* para usuários normais de PC que é muito comum e algumas características em relação às outras versões.

O forte crescimento da computação em nuvem tornou-se vital para o sucesso do futuro do sistema operacional *Ubuntu Server*, que tem um *software cloud* embutido no produto tornando-se uma escolha ideal para todos os modelos de nuvem, seja para nuvens públicas, privada, híbrida ou comunitária[10]. Esse sistema operacional foi implementado no uso deste trabalho.

O uso destas ferramentas foi de suma importância no teste em modelo de nuvem privada. Foi utilizado duas formas de coleta e análise dos pacotes nos seguintes cenário, utilizando o conceito de virtualização, foi criada várias máquinas virtuais utilizando o *software VMware*, depois da virtualização pronta, foi instalado o sistema operacional *ubuntu server* e implementada com o pacote LAMP (*Linux, Apache, MySQL e PHP*). Desta forma foi possível fazer vários tipos de testes, como o ataque de *Bots* na situação de Negação de Serviço.

A base de dados da Análise por Assinatura foi coletada de ferramentas *Sandboxes*, que identificam características de vários tipos de ataques. Enquanto que, na fase de treinamento da Análise de detecção, foram utilizados arquivos contendo pacotes infectados sob situação de ataque de *Bots* na situação de Negação de Serviço, para a determinação do perfil do ataque de *Bot* e pacotes em tráfego normal para identificar o tráfego padrão da rede.

Os pacotes foram coletados armazenados, logo após foram abertos e verificados cada assinatura e comprado com cada assinatura que estava na base de dados de assinaturas, no final desse processo o agente de reação recebe uma ação.

De acordo com as bases de dados de assinatura, a ferramenta foi testada em uma nuvem do tipo pública e em uma nuvem do tipo privada localizada na Faculdade FACEMA, durante um período de 10 dias, onde foram coletados aproximadamente 2GB de tráfego da rede. Durante esse processo foram identificados 147 ataques, deste total, 113 foram encontradas na análise por assinatura, aproximadamente 76,87%, sendo 22,13% na análise por detecção, num total de 34 ataques. Nesta análise verificou-se uma taxa de falsos positivos de 0,012%.

6.2 Testes de Ataque

De acordo com o modelo proposto foi desenvolvido um protótipo do C2IDS e realizado testes da ferramenta, Para realização dos testes de detecção foi utilizado como ambiente de captura uma nuvem criada nos servidores da empresa www.blunethost.com.br. Esta nuvem se tornava do tipo híbrida, que poderia ser acessada de qualquer lugar deste de que o cliente tivesse a acesso à internet e autorização dos administradores para *login*.

O ambiente foi composto de um servidor com processador *Core* i7 3.33*GHz* com memória RAM de 256 MB, HD 20GB, contendo um IP válido com o tráfego de até 1000 GB, disponível em um *Link Full Duplex* de 100Mbps com o sistema operacional *Ubuntu™ Linux 64bits*.
Com relação às características das máquinas utilizadas temos:

Máquinas de Teste: Sistema operacional *Windows*XP SP3, utilizando software *VMware*para emular duas máquinas virtuais. Nas máquinas virtuais foram usados Sistema operacional *Windows XP SP3* e *Linux Ubuntu*. Como recurso físico as máquinas contam com 20 Gigabytes de disco rígido, memória 1GB e processador *Core* i5.

Notebook da C2IDS: Sistema operacional *Windows*8 64bis, utilizando *software*C2IDS. Como recurso físico a máquina conta com 450*GIGABYTE* de disco rígido, memória 6GB e processador *Intel(R) Core (TM) i5-2.3GHz*. O *hardware* se mostrou mais do que suficiente, já que esta máquina é apenas responsável pela coleta dos dados. Para a realização dos testes de detecção da ferramenta foi necessário um treinamento da nuvem para determina um perfil padrão. Esta fase teve duração de 05 horas, realizada no dia 04 de abril, onde foram coletados 2.096.010 pacotes, ou seja, aproximadamente 780MB de dados em tráfego normal como: verificação de e-mail, navegação em páginas, downloads, entre outras atividades básicas de uma nuvem como: enviando dados para nuvem, aumentando e diminuindo sua infraestrutura, trocam de serviço da nuvem, esses resultados estão especificados posteriormente nos resultados. O mesmo procedimento foi aplicado a um conjunto de dados em ataque, sendo coletados 2.364.130 pacotes ou 702MB de dados desse tráfego, no dia 05 de abril.

6.2.1 Servidor Socket

O servidor foi criado para receber e identificar as informações sobre o ataque foi instanciado um objeto do tipo *Server Java*, inicializando um objeto informando somente a Porta. Como se trata de um servidor o mesmo já é determinado por padrão que seu IP deve ser do tipo *localhost* ou o que se encontrar na placa de rede que vai receber as configurações da rede...

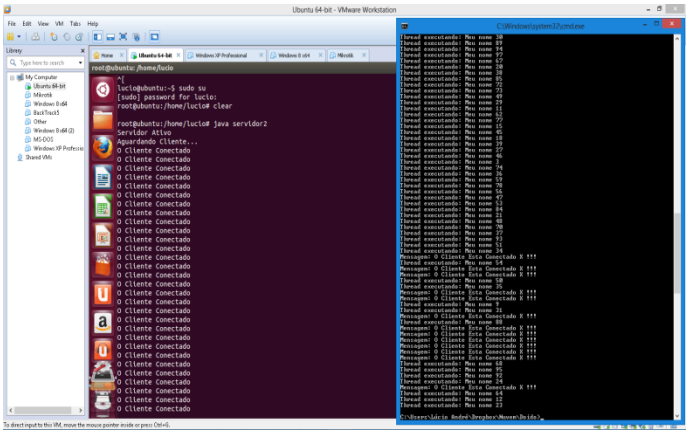

Figura 06: Teste de Ataque**.**

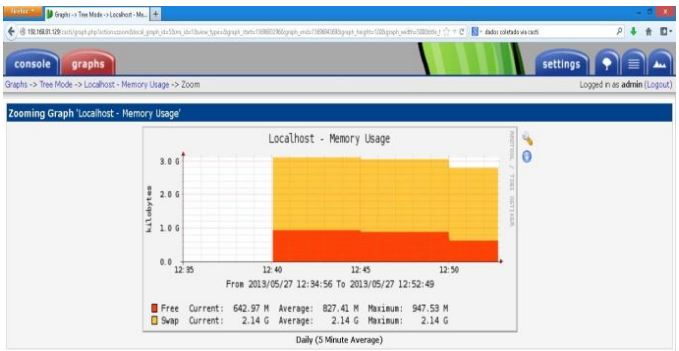

Figura 07: Captura de tráfego.

Durante os teste de ataque foi coletado o trafego da nuvem em tempo real pela ferramenta *cacti* como é mostrado na figura 09, baseado nas especificações

acima foi utilizado 2.14 G de tráfego em modo de armazenamento *Swap* e em modo *Free* foi coletado 947.53M.

7 CONSIDERAÇÕES FINAIS

Com a utilização da técnica de detecção por assinatura podemos identificar *Bot* que operam isoladamente ou em conjunto com outros *Bots*, no entanto, está técnica só identifica *Bots* conhecidos que se encontram na base de dados de assinaturas. No entanto durante o processo de treinamento da rede, visto que é de fácil implementação, não precisa de um grande volume de dados para determinação de um perfil padrão, além de executar em menor intervalo de tempo.

O estudo confirmou um resultado satisfatório da ferramenta de detecção de intrusos com a identificação de *Botnet*s na situação de negação de serviço na computação em nuvem, e que nas redes privadas, públicas, hibridas ou comunitária, não é necessário à coleta de grandes volumes de dados, pois a ação deste tipo de ataque não são duradouras dentro de uma rede, portanto é necessária a identificação em tempo real do intruso, para que assim o processo de proteção possa ser mais funcional com a relação aos outros métodos já utilizados.

Com o desenvolvimento desse trabalho foi possível mostrar a potencialidade dos Sistemas de Detecção de Intrusos Identificando ataques de *Botnet* em situação de Negação de Serviço em Computação em Nuvem, principalmente pelo uso de agentes para a defesa e a manutenção da segurança de nuvens. No entanto, apesar das vantagens apresentadas pela solução proposta é importante relatar a necessidade de um planejamento para a implantação da ferramenta, visto que, há inúmeros riscos e limitações dessa tecnologia.

7.1 Sugestões para Trabalhos Futuros

Para a continuidade deste trabalho e sugestão para trabalhos futuros nessa área, podemos citar:

- A implementação de *Honeypots* para Computação em Nuvem

- Criando uma nuvem de forma que na segurança seja utilizada IDS com detecção de ataques com técnicas de IA (Inteligência Artificial).

- Testes utilizando *Botnet* em redes sem fio;

- Criando aplicações para otimizar processamento de banco de dados em nuvem.

8 REFERENCIA BIBLIOGRÁFICA

[1] IANELLI, Nicholas.; HACKWORTH, Aaron, **Botnets as a Vehicle for Online Crime**, 2005, Disponível em: <http://www.cert.org/archive/pdf/Botnets.pdf>, Acessado em: 30 mar. 2013**.**

.

[2] MUNIR, Kashif.; PALANIAPPAN, Sellapan, **Security Threats Attacks Present in Cloud Environment**, 2012, Disponível em: <http://la.trendmicro.com/media/report/virtualization-and-cloud-security-report-en.pdf >, Acessado em: 11 maio 2013.

.

[3] CHOI, Hyunsang.; LEE, Heejo.; KIM, Hyogon, **BotGAD: Detecting Botnets by Capturing Group Activities in Network Traffic**, 2009, Disponível em: <http://ants.iis.sinica.edu.tw/3bkmj9ltewxtsrrvnoknfdxrm3zfwrr/84/BotGAD.pdf >, Acessado em: 20 abr. 2013.

[4] GU, Guofei.: ZHANG, Junjie.: LEE,Wenke, **Botsniffer –Detecting Botnet Command and Control Channels in Netw,** 2007, Disponível em: <http://corescholar.libraries.wright.edu/cgi/viewcontent.cgi?article=1006&context=cse >, Acessado em: 30 jan. 2013.

[5] CHOI, Hyunsang.; LEE, Heejo.; KIM, Hyogon, **BotGAD: Detecting Botnets by Capturing Group Activities in Network Traffic**, 2009, Disponível em: <http://ants.iis.sinica.edu.tw/3bkmj9ltewxtsrrvnoknfdxrm3zfwrr/84/BotGAD.pdf >, Acessado em: 20 abr. 2013.

[6] GU, Guofei.; PERDISCI, Roberto.; Zhang ,Junjie.; LEE, Wenke, **BotMiner: Clustering Analysis of Network Traffic for Protocol and Structure Independent Botnet Detection,** 2008, Disponível em: <https://www.damballa.com/downloads/a_pubs/Usenix08.pdf >, Acessado em: 20 jan. 2013.

[7] PILONE, Dan., PITMAN, Neil. (2005) “UML 2.0 in a Nutshell”.

[8] LAZAREVIC, Aleksandar.; KUMAR,Vipin.; SRIVASTAVA, Jaideep, **Intrusion Detection:** A Survey**,** 2001, Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=

10.1.1.144.8462&rep=rep1&type=pdf >, Acessado em: 20 jan. 2013.

[9] NANDA, Susanta.; CHIUEH, Tzi-cker, **A SurveyonVirtualization Technologies,** 2000, Disponível em:< http://www.ecsl.cs.sunysb.edu/tr/TR179.pdf >, Acessado em: 27 abr. 2013.

[10] CARR, Gerry Carr, **SurveyUbuntu Server**, 2012, Disponível em: <http://www.canonical.com/sites/www.canonical.com/files/active/images/server_survey_2012.pdf >, Acessado em: 27 abr. 2013.