MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
SEÇÃO DE ENGENHARIA DE COMPUTAÇÃO - SE/8

Jonas Rocha Lima Amaro, Yago Guimarães Coimbra

Ferramenta para detecção de padrões de botnet baseado em algoritmos de agrupamento de aprendizagem de máquina

Rio de Janeiro
2016

Jonas Rocha Lima Amaro, Yago Guimarães Coimbra

Ferramenta para detecção de padrões de botnet baseado em algoritmos de agrupamento de aprendizagem de máquina

Trabalho Apresentado ao Curso de Engenharia de Computação do Instituto Militar de Engenharia como Verificação Especial do Projeto de Fim de Curso.

Instituto Militar de Engenharia

Orientador: Sergio dos Santos Cardoso Silva

Rio de Janeiro
23 de Maio de 2016

Instituto Militar de Engenharia
Praça General Tibúrcio, 80 - Praia Vermelha
Rio de Janeiro - RJ CEP: 22290-270

Amaro, Jonas e Guimarães, Yago
S586d Ferramenta para detecção de padrões de botnet baseado em algoritmos de agrupamento de aprendizagem de máquina / Jonas Rocha Lima Amaro, Yago Guimarães Coimbra. - Rio de Janeiro: Instituto Militar de Engenharia, 2016.

28f. : il., graf., tab. : -cm.

Projeto de Fim de Curso - Instituto Militar de Engenharia
Orientador: Sergio dos Santos Cardoso Silva.

1 - Botnets 2 - Clustering

CDU 631.317.35

Jonas Rocha Lima Amaro, Yago Guimarães Coimbra

Ferramenta para detecção de padrões de botnet baseado em algoritmos de agrupamento de aprendizagem de máquina

Trabalho Apresentado ao Curso de Engenharia de Computação do Instituto Militar de Engenharia como Verificação Especial do Projeto de Fim de Curso.

Trabalho aprovado. Rio de Janeiro, 23 de Maio de 2016:

Sergio dos Santos Cardoso Silva - D.Sc., do IME
Orientador

Raquel Coelho Gomes Pinto - D.Sc., do IME

Julio Cesar Duarte - D.Sc., do IME

Rio de Janeiro
23 de Maio de 2016

Resumo

Botnets são uma ameaça cibernética que já trouxe muito prejuízo[1]. Essa ameaça utiliza computadores infectados para realizar atividades fraudulentas como servir páginas piratas para roubar informações sensíveis, enviar *spam* para usuários comuns e enviar sucessivas requisições para derrubar servidores. Por ser uma atividade ilegal, os criminosos realizam a comunicação entre as máquinas com comportamentos divergentes.
Baseado nessa premissa, esse projeto se propõe a detectar máquinas que pertencem a botnets a partir de algoritmos orientados à Detecção de Anomalia.
Dentro do contexto Exército Brasileiro, o projeto auxiliará a Inteligência do Exército Brasileiro na prevenção de ataques por botnets. Esse sistema será o raciocinador para um sistema que desarticula botnets.

Palavras-chave: botnets, bots, detecção de botnets, aprendizagem de máquinas, clustering, detecção de anomalia.

Abstract

Botnets are a cyber threat that already brought plenty of money dispend[1]. This threat uses infected computers to perform fraudulent activities, such as serving pirated sites to steal sensible information, sending spam to common users and sending successive requests to get servers down. Because it is an illegal activity, the criminals do the communication between the machines with divergent behaviours.
Based on that premise, this project proposes to detect machines that are part of a botnet using Anomaly Detection oriented algorithm.
Inside the Brazilian Army context, the project will help the Inteligence of Brazilian Army preventing botnet attacks. This system will be the Decision Maker for a botnet desarticulating system.

Keywords: botnets, bots, botnet detection, machine learning, clustering, anomaly detection.

Lista de ilustrações

Lista de abreviaturas e siglas

C&C	Comando e Controle
IRC	*Internet Relay Chat*
HTTP	*HyperText Transfer Protocol*
P2P	*Peer-to-peer*
IDS	*Intrusion Detection System*
DNS	*Domain Name System*
CDCiber	Centro de Cibernética

Sumário

1 Introdução

As botnets, que são redes formadas por máquinas infectadas por alguma forma de *malware*, apresentaram um elevado crescimento, se tornando uma das maiores ameaças da área de segurança da informação. A detecção dessas redes se tornou um tópico muito importante na área de segurança da informação, devido ao desafio que é realizar essa detecção. Pois, as botnets são muito flexíveis e robustas, além de estarem em processo de aprimoramento contínuo.

Com a necessidade de realizar essa detecção, mesmo com as mudanças realizadas no funcionamento das botnets ao longo do tempo, o emprego de técnicas de aprendizagem de máquina é muito promissor, especialmente as técnicas de agrupamento. Porque é esperado, que os bots que formam as botnets, tenham comportamento diferente de usuários convencionais, e, dessa forma, espera-se que eles sejam agrupados em clusters menores dos formados pela massa de usuários normais.

Levando em conta essas características, o presente trabalho busca desenvolver uma ferramenta para a detecção de botnets, utilizando as informações das consultas ao servidor DNS realizadas na rede através de algoritmos de aprendizagem de máquina.

1.1 Objetivo

O objetivo deste trabalho é desenvolver uma ferramenta para detectar possíveis hospedeiros de bots em uma botnet, reduzindo o trabalho humano utilizado na detecção das botnets. Isso será feito utilizando algoritmos de agrupamento, que utilizarão dados que serão calculados através das informações contidas em consultas de DNS realizadas.

Além disso, o desempenho dessa ferramenta será avaliado, utilizando as informações do log de DNS dos servidores do IME, para o qual já temos algumas máquinas infectadas previamente mapeadas.

1.2 Motivação

O crescimento e diversificação do uso da Internet, criaram o cenário ideal para o desenvolvimento das botnets, que são consideradas uma das maiores ameaças atuais na área de segurança da informação[2].

A detecção das botnets é um grande desafio, pois os atacantes estão sempre aprimorando o funcionamento dos bots, com isso os mecanismos atuais muitas vezes falham ao detectar novas implementações de botnets. Isso motivou o uso de algoritmos de agru-

pamento para serem utilizados na detecção de botnets, de forma que a ferramenta seja capaz de detectar inclusive botnets que eram desconhecidas.

1.3 Justificativa

A justificativa desse trabalho é contribuir para a área de defessa cibernética, uma área de interesse do IME, apoiando à pesquisa com o conhecimento de características de consultas de DNS que podem ser usadas para realiar a detecção de botnets. Além disso, a ferramenta pode ser utilizado pelo Centro de Defesa Cibernética (CDCiber) ou servir como um módulo de um sistema integrado de detecção de botnets[3].

1.4 Metodologia

Este trabalho foi dividido em três fases principais: estudo teórico, preparação dos dados, implementação e análise e testes dos resultados.

Na etapa de estudo teórico foi feito um levantamento da funcionalidade das botnets e do funcionamento e aplicações dos algoritmos de aprendizagem de máquina, com o objetivo de identificar vulnerabilidades e padrões nas botnets, motivando a seleção de características relevantes dos dados utilizados, de forma a melhorar a performance do algoritmo.

Na etapa seguinte, foi feito um levantamento das características canditadas a serem utilizadas pelo algoritmo da ferramenta. De posse dessas características, foi implementado um programa para realizar a extração automática dessas características da base de dados que utilizamos.

A próxima etapa será a implementação das técnicas de agrupamento nos dados tratados, por fim será feita uma análise, para identificar a técnica mais adequada e possíveis refinamentos na ferramenta.

1.5 Estrutura

Este trabalho é composto por seis capítulos, iniciando com esta introdução. No capítulo 2 é feito um estudo teórico sobre botnets para identificar o tipo de *malware* que desejamos identificar. Em seguida, no capítulo 3 é realizado um estudo sobre algoritmos de aprendizagem de máquina dando ênfase nos algoritmos de agrupamento, que será a estratégia usada neste trabalho. Em seguida, no capítulo 4 são explicados os procedimentos e a motivação que levaram à forma de tratamento dos dados. No capítulo 5, é mostrado o cronograma que planejamos seguir até a conclusão do trabalho. Ao final, no capítulo 6, estão as conclusões realizadas até o presente momento.

2 Botnets

As Botnets são redes formadas por máquinas infectadas com *malware*, permitindo que o atacante (*botmaster*) realize diversas atividades criminais remotamente, como roubo de informações, ataques de negação de serviço, envio de SPAM, etc.[1]

Com o crescimento e diversificação do uso da Internet, o meio cibernético se tornou mais relevante e mais atraente para a realização de ataques maliciosos. Isso motivou o crescimento do número de botnets existentes e aumentou o potencial de contaminação das mesmas, além disso, para evitar os mecanismos de detecção existentes, elas se tornaram cada vez mais sofisticadas.

Para que o detector se torne mais robusto, é preciso compreender o funcionamento das botnets e seus objetivos. Esse conhecimento é necessário para entender as configurações existentes nas botnets atuais, além de compreender como essas configurações podem evoluir. De posse desse conhecimento, espera-se que seja possível identificar características relevantes e intrínsecas ao funcionamento das botnets, mesmo quando o *botmaster* estiver tentando evitar os mecanismos de detecção.

2.1 Elementos das Botnets

As botnets apresentam alguns elementos estruturais tipicamente envolvidos, que estão presentes independente do protocolo ou arquitetura utilizada. A Figura 1 mostra a estrutura desses elementos e como eles se relacionam em uma botnet. Segue uma descrição para cada componente:

- Bots: São *malwares* instalados nos computadores das vítimas que podem realizar as ações maliciosas que o *botmaster* envia através do canal de comando e controle (C&C). Geralmente, o *malware* é inicializado quando o hospedeiro inicializa a máquina, porém isso pode ser configurado pelo *botmaster* para dificultar a detecção da atividade maliciosa.

- Hospedeiros (Zumbis): São as máquinas em que o bot foi instalado, ou seja, infectadas.[**??**]

- *Botmaster*: é o indivíduo que configura o bot, dissemina e controla a botnet.

- Canal de Comando e Controle (C&C): é o meio que o *botmaster* tem para se comunicar com a sua botnet. É a parte chave do funcionamento, pois é necessário para o envio dos comandos de atividade maliciosa aos hospedeiros. Dessa forma,

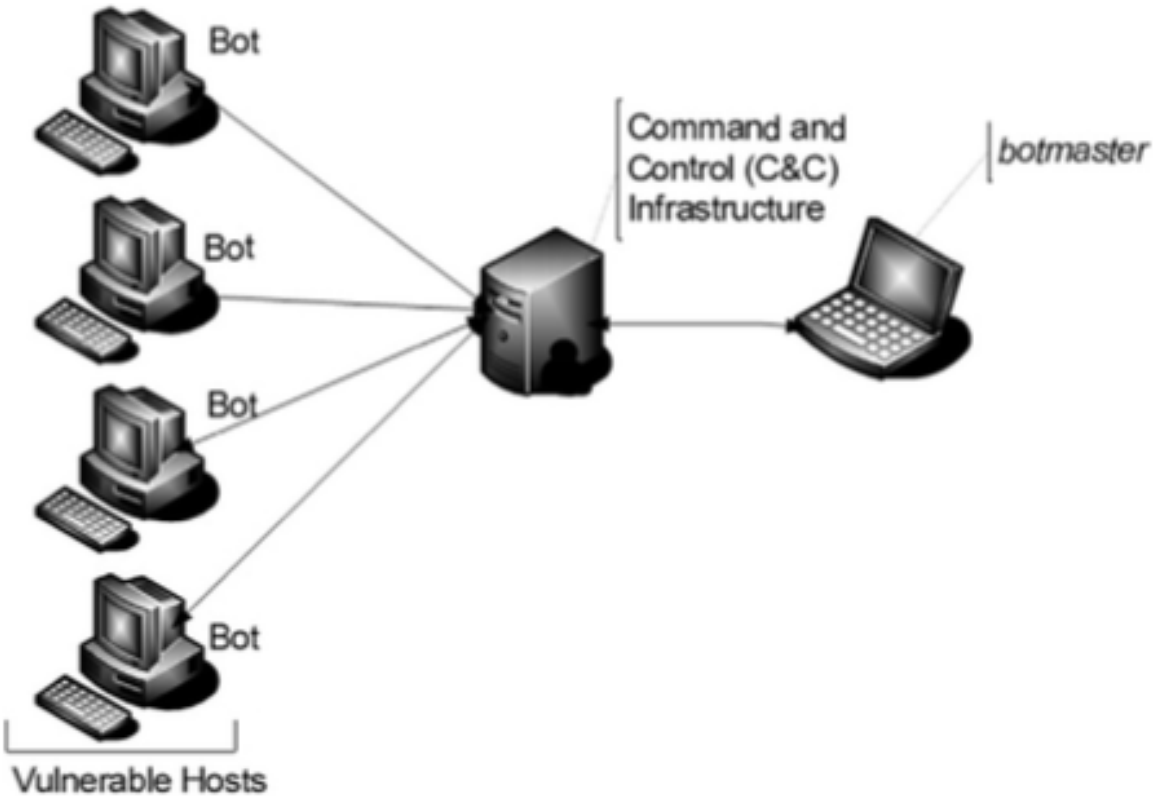

Figura 1 – Elementos das botnets [1]

grande parte das características da botnet, como robustez, facilidade de detecção/desativação, estabilidade, etc., são definidas pela forma que a infraestrutura de C&C está organizada.

2.2 Ameaças e Formas de Defesa

O crescimento do número de máquinas conectadas constantemente a enlaces de alta velocidade e rodando sistemas com vulnerabilidades consideráveis, criou um ambiente favorável à formação de botnets. Além disso, muitas vezes o bot é transparente ao responsável pela máquina infectada, ou seja, não atrapalha o funcionamento da máquina, fazendo com que a vítima não perceba a infecção e tente combatê-la. Esses fatores, aliados ao enorme potencial de causar danos, fazem com que as botnets sejam um dos maiores desafios de pesquisa em segurança no espaço cibernético atual. [4]

Existem características que tornam o host mais interessante ao *botmaster* como: altas taxas de transmissão, baixos níveis de segurança e monitoração, alta disponibilidade e localização distante (dificultando que as agências reguladoras detectem as atividades, já que os bots estarão espalhados por diversas nações). Esses fatores ajudam o bot a passar

desapercebido e a contribuir com maior capacidade de banda ao *botmaster*, facilitando ataques como os de negação de serviço.

Existem duas formas para combater um ataque realizado por botnets: reativamente ou preventivamente. A forma reativa é a mais comum e envolve detectar a existência da atividade maliciosa e reagir ao ataque tentando reduzir o tráfego malicioso para níveis aceitáveis. Uma desvantagem é que o ataque já vai ter sido inicializado quando for detectado, ou seja, já vai haver causado danos antes de ser solucionado. A forma preventiva busca evitar que a botnet possa realizar alguma atividade maliciosa, porém essa atividade não é simples, já que o atacante pode aprimorar seus bots, tornando-os mais sofisticados, exigindo grandes investimentos para manter os recursos de segurança atualizados.

O mecanismo que estamos desenvolvendo é da forma preventiva, já que o algoritmo busca encontrar padrões e identificar possíveis máquinas infectadas por botnets. Tudo isso na fase em que o *botmaster* ainda está infectando máquinas com o bot para a botnet. Ou seja, buscamos detectar as máquinas infectadas antes do ataque, como um ataque de negação de serviço, em si ser efetivado . Uma característica desejável para um detector é a detecção em tempo real, com o objetivo de minimizar os danos causados e o tempo de reação do *botmaster*. Porém, obter essa característica é um desafio, devido ao grande número de dados que devem ser tratados e analisados. Dessa forma, nosso projeto não fará detecção em tempo real, mas tentará se aproximar disso, utilizando a detecção dos dados coletados ao longo de um dia para detectar bots que atuaram nas últimas 24 horas.

2.3 Ciclo de Vida das Botnets

Na maioria dos casos, existe um ciclo com fases bem definidas de como uma botnet é criada e mantida, a Figura 2 mostra essas fases para cada novo hospedeiro que é contaminado.

Na primeira fase, chamada de injeção inicial, o atacante procura vulnerabilidades na máquina do futuro hospedeiro para explorá-las e infectá-lo com o *malware*, tornando-se um bot em potencial, isso pode ocorrer, por exemplo, através de um *download* indesejado ou através de um anexo em um e-mail. Após a infecção ser bem sucedida, ocorre a injeção secundária: o host infectado, através do *malware* inicial instalado, busca em uma rede os reais binários do *malware* do bot, os quais após baixados e executados concluirão a infecção e transformam o host em um bot real.[5].

Durante a fase de conexão, o bot estabelece conexão com o canal de C&C. Isso se repete sempre que o host é reiniciado, podendo ser considerada uma fase vulnerável já que é uma fase essencial, além de geralmente seguir um padrão. Após a efetivação da conexão, o bot se torna ativo na botnet, e passa a realizar os comandos enviados pelo *botmaster* através do canal de C&C, efetivando as atividades maliciosas solicitadas. A

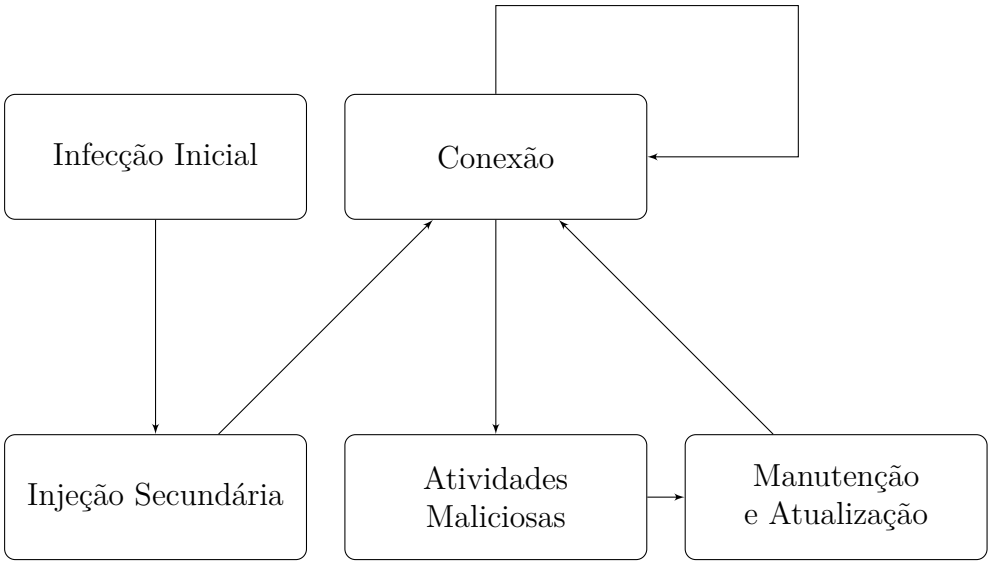

Figura 2 – Ciclo de Vida das Botnets

última fase é a de manutenção e atualização, e tem por objetivo manter a botnet ativa e atualizada, já que se o *botmaster* deseja que os bots possam evitar novas técnicas de detecção, adicionar novas funcionalidades ou até mesmo alterar o servidor de C&C, os binários do programa bot devem ser modificados.

2.4 Arquitetura das Botnets

Existem 4 tipos de arquiteturas para as botnets: centralizada, descentralizada, híbrida e aleatória.

Na arquitetura centralizada, mostrada na Figura 3 todos os bots se comunicam com um número pequeno de servidores de C&C. Embora ela ofereça vantagens ao *botmaster*, como baixa latência e facilidade de manutenção, ela também torna a botnet bastante vulnerável, permitindo que a botnet seja desligada após a identificação dos poucos pontos centrais de C&C. Esta arquitetura é muito utilizada por botnets que utilizam o protocolo IRC (*Internet Relay Chat*) para comunicação. Todavia, o tráfego desse protocolo é incomum e raramente utilizado, especialmente em ambientes corporativos, por esse motivo, o tráfego desse tipo de protocolo costuma ser bloqueado, inutilizando a botnet. Devido a isso, o uso do protocolo HTTP (*HyperText Transfer Protocol*) se popularizou já que ele é amplamente utilizado, disfarçando as comunicações das botnets.

A fragilidade da arquitetura centralizada, motivou o desenvolvimento da arquitetura descentralizada, na qual uma variedade de protocolos P2P (*Peer-to-peer*) é utilizada. A flexibilidade e robustez dessa arquitetura, permite que mesmo que muitos bots sejam desativados a botnet possa continuar funcionando, já que não existem pontos centralizados de C&C.

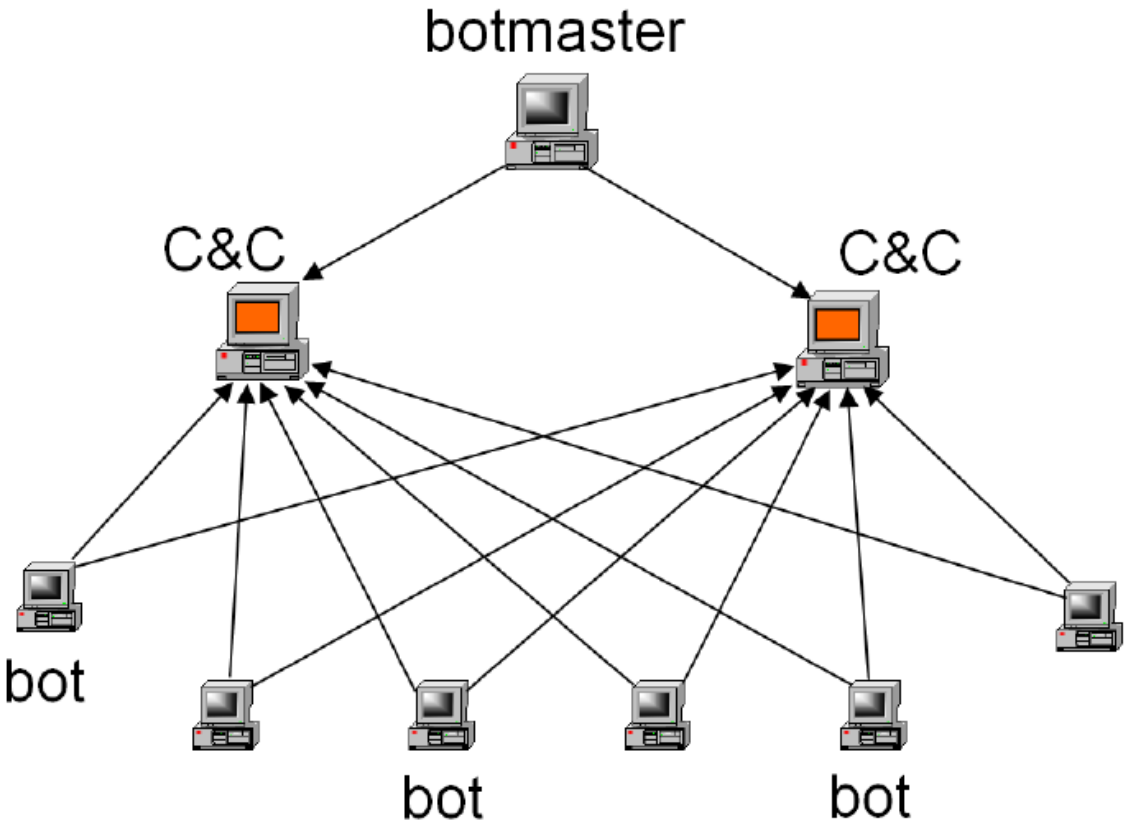

Figura 3 – Arquitetura Centralizada[6]

As arquitetura híbridas apresentam características de ambas as arquiteturas centralizadas e descentralizadas, como mostrado na Figura 4, na qual os bots são classificados em dois grupos: clientes e serventes. Os serventes exercem os papéis tanto de clientes quanto servidores, possuindo endereço de IP estático e público para serem acessíveis globalmente, sendo utilizados para repassar os comandos enviados pelo *botmaster*. Os demais bots, são denominados clientes pois não aceitam comunicações de entrada, dessa forma e podem apresentar IP dinâmico, privado ou protegidos por *firewall* para não serem roteados facilmente.

Por fim, a arquitetura aleatória é um modelo, até agora, apenas teórico. No qual o bot não se comunica ativamente com o *botmaster* ou com outros bot. Dessa forma, para realizar um ataque, o *botmaster* precisa vasculhar a rede em busca de um bot para enviar o comando e realizar as atividades maliciosas.

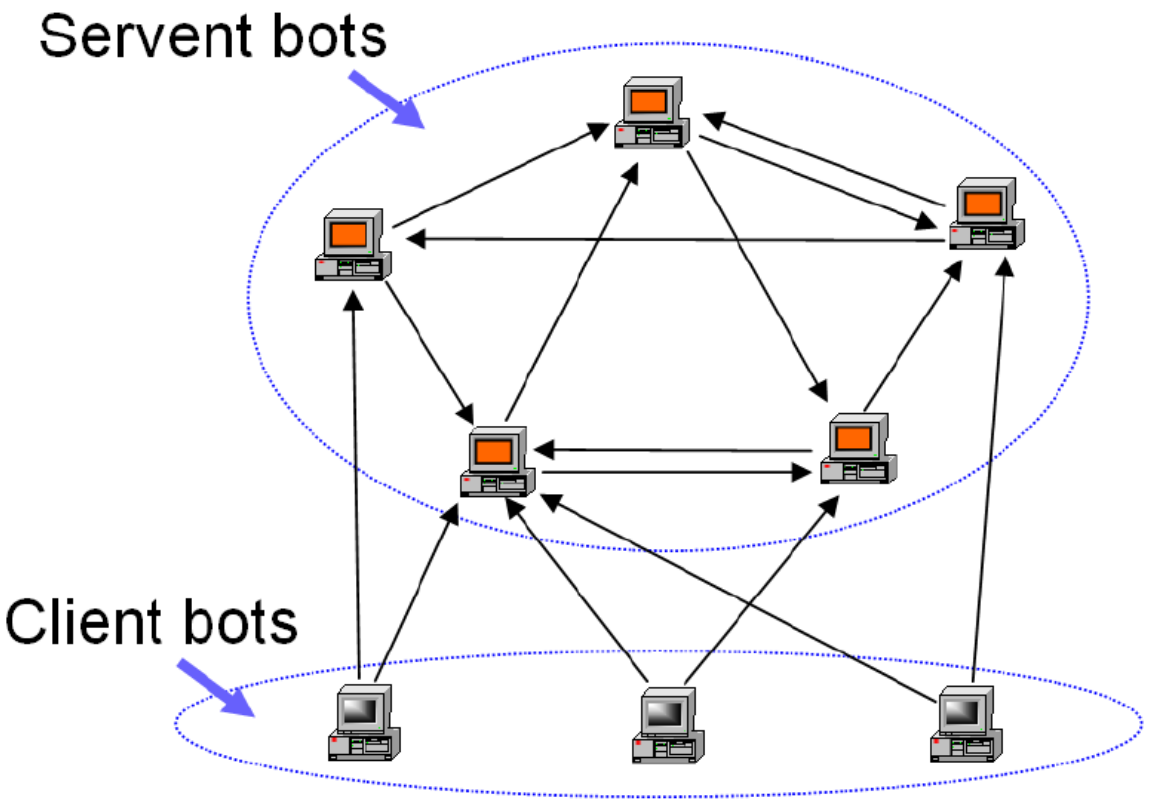

Figura 4 – Arquitetura Híbrida[6]

2.5 Detecção de Botnets

Existem duas categorias de técnicas para detecção de botnets: *honeynets* e sistemas de detecção de intrusos (IDS). As *honeynets* consistem na criação de redes com a intenção de que elas sejam comprometidas, permitindo que as informações sobre a botnet sejam captadas, por isso elas são consideradas mais efetivas para compreender as características de uma botnet do que a detecção propriamente dita.

A detecção por IDS, pode ser classificada entre duas técnicas: a baseada em assinaturas e a baseada em anomalias. A técnica baseada em assinaturas, consiste em extrair padrões da rede e comparar com um banco de dados onde se encontram os padrões que já foram vistos em botnets, ou seja, ela não permite que novas botnets sejam identificadas e envolve a posse de um banco de dados enorme com o maior número de informações existentes sobre as botnets previamente detectadas. Dessa forma, a técnica baseada em anomalias é a principal área de pesquisa para detecção de botnets, baseando-se em anomalias na rede, como alta latência, aumento no tráfego ou uso de portas incomuns para detectar a presença de bots na rede.

As técnicas baseadas em anomalias, podem ser baseadas no host, onde cada máquina possui uma ferramenta de monitoração instalada (o que não é muito escalável), e tem

seu comportamento analisado para verificar a existência de atividades suspeitas. Além disso, a análise pode ser baseada na rede, ativa (que possuem a grande desvantagem de aumentar o tráfego da rede ao injetar pacotes com a finalidade de examinar se um cliente é humano ou um bot) ou passivamente, sendo esta última a forma de detecção mais utilizada e pesquisada atualmente.

A monitoração passiva de uma rede consiste em analisar o tráfego da rede buscando por comunicações suspeitas que podem ter sido enviadas pelos bots ou canais de C&C. Essa monitoração é possível pois os bots de uma mesma botnet costumam apresentar padrões de comunicação, já que eles são pré-programados pelo mesmo *botmaster* para entrar contato com o servidor de C&C.

Para que a análise do tráfego seja viabilizada, são empregadas diversas técnicas como métodos estatísticos, mineração de tráfego, teoria de grafos, clustering, modelos estocásticos, redes neurais, entre outras.

A detecção de botnets é uma tarefa bastante desafiadora porque os *botmaster*s estão sempre aprimorando os bots, tornando os mais difíceis de serem detectados. Por exemplo, as primeiras detecções buscavam mensagens suspeitas nos conteúdos da mensagem, afim de evitar isso os *botmaster*s passaram a utilizar criptografia tornando essa técnica de detecção obsoleta. Outra dificuldade para algoritmos de clustering é que podem ser evitados usando técnicas de randomização nas comunicações e atribuição de tarefas diferentes para os bots.

3 Aprendizado de Máquina

Como Bishop [7] descreve, aprendizado de máquina é uma maneira de abordar um problema de computação. Nessa abordagem, a partir de um grande conjunto de dados, chamados como conjunto de treinamento, são inferidos um conjunto de parâmetros a serem utilizados em um modelo parametrizado.

3.1 Definições

Algumas breves definições serão apresentadas para fim de ambientar o leitor nos temas discutidos neste capítulo.

Conjunto de exemplos: Seja o conjunto de exemplos os dados conhecidos do problema.

Características: Sejam características um conjunto ordenado de valores que descrevem um exemplo, a ser modelada pelo algoritmo de Aprendizado.

Etiquetas: Seja etiqueta de exemplo, ou simplesmente etiqueta, a saída esperada do modelo para aquela instância.

Acurária: A proporção de acertos pelo tamanho do conjunto de exemplos.

Precisão: A proporção de exemplos inferidos corretamente como verdadeiros pela quantidade de inferidos como verdadeiro.

Abrangência: A proporção de exemplos inferidos corretamente como verdadeiros pela quantidade de acertos.

F1 score: A média harmônica entre a precisão e a abrangência

$$\frac{2 * \text{precisão} * \text{abrangência}}{\text{precisão} + \text{abrangência}}$$

Matriz X: Seja $\mathbf{X}$, exemplos de treinamento, uma matriz $m \times n$, o qual m é quantidade de instâncias e n é a quantidade de características. $\mathbf{X}$ é a representação matemática do Conjunto de exemplos.

Vetor Y: Seja $\mathbf{Y}$, conjunto de etiquetas, um vetor de tamanho m, a quantidade de instâncias, $\mathbf{Y}$ é o conjunto ordenado das respectivas saídas esperadas de cada linha da matriz $\mathbf{X}$, ou seja as etiquetas.

É importante destacar a relevância dos conceitos de precisão, abrangência e *F1 score* para problemas que possuem poucos exemplos positivos. Essas definições permitem avaliar esses tipo de dados, como discutiremos na sessão de Detecção de Anomalias.

3.2 Categorias de Problemas

Para ser mais específico, é possível classificar os problemas resolvidos pela aprendizado de máquina em cinco categorias[8].

Classificação: Decidir a classe de exemplo dadas às suas características, por exemplo decidir qual dígito foi escrito a apartir de uma imagem de dígito escrita a mão.

Regressão: Determinar um valor real para cada exemplo, por exemplo o risco de um paciente ter contraído câncer a partir de imagens e resultados de exames.

Ordenação: Ordenar os exemplos a partir de algum critério, por exemplo listar produtos por relevância a partir das palavras chaves da busca do usuário.

Agrupamento: Particionar os exemplos em regiões homogêneas, por exemplo identificar comunidades dentro de redes sociais massivas.

Redução de Dimensionalidade: Representar o conjunto de exemplos com um número reduzido de dimensões, por exemplo comprimindo imagens para processamento de imagens.

Neste projeto foi decidido encarar o problema como agrupamento, apesar de também poder se confundir com classificação, já que o objetivo é auxiliar a detecção de botnets.

3.3 Cenários dos Dados

Categoriza-se[8] sete cenários para os algoritmos de aprendizado, esse cenários são fortemente influenciados pelas condições dos dados de treinamento.

Aprendizado Supervisionado: O modelo tem acesso a dados com os resultados de saída já esperados, ou etiquetados, como lê-se na literatura. Os problemas mais comuns desse tipo de cenário são classificação, regressão e ordenação.

Aprendizado Não Supervisionado: Só se dispõe da conjunto de exemplos para treinamento sem etiquetas. Geralmente é mais utilizado para classificação, agrupamento e redução de dimensionalidade

Aprendizado Semi-Supervisionado: Neste cenário, é possível acessar uma conjunto de exemplos sem etiquetas e uma com etiquetas. Esse é o caso de problemas em que dados sem etiquetas são fáceis de serem adiquiridos, ao contrário dos dados etiquetados, pela dificuldade de etiquetar.

Inferência Transdutiva: Semelhante ao Aprendizado Semi-Supervisionado, nem todos os exemplos são etiquetados, mas o modelo só deve ser generalizado apenas para os exemplos conhecidos.

Aprendizado On-line: Neste cenário, iterasse no modelo a cada exemplo recebido em rodadas. No início da rodada, o modelo recebe um exemplo, inicialmente sem etiqueta, realiza uma predição, recebe a etiqueta e atualiza os parâmetros do modelo.

Aprendizado por Reforço: Neste cenário, os modelos são criados baseado num sistema de recompensa. O modelo é recompensado a cada decisão bem-feita ao interagir com um ambiente.

Aprendizado Ativo: O algoritmo é quem realiza requisições a uma entidade capaz de etiquetar exemplos para melhorar os parâmetros do modelo. O objetivo é conseguir gerar um modelo tão bom quando o modelo supervisionado, porém com menos exemplos.

E ainda há outros possíveis cenários ainda mais complexos e específicos. Esses cenários não foram catalogados neste trabalho porque a campo de Aprendizado de Máquina está ainda em constante fase de crescimento.

3.4 Detecção de Anomalia

O artigo, *Anomaly Detection: A Survey*[9], define Detecção de Anomalia como o problema da busca por padrões nos dados para apontar os dados que não seguem um padrão esperado. Essas anomalias podem acontecer em diversos contextos, ruídos nas medidas, defeitos em equipamentos, comportamentos inovadores ou até atividades fraudulentas, como fraude de cartão de crédito e ataques cibernéticos. Os algoritmos de Detecção de Anomalia podem ser utilizados para resolver problemas de Classificação e Agrupamento.

3.4.1 Etiquetas em Detecção de Anomalia

Geralmente a tarefa de etiquetar os dados de modo representativo e preciso é custosa. A forma mais comum de etiquetagem é feita manualmente por um especialista. Esse processo é geralmente custoso, pois são necessários muitos exemplos para encontrar uma anomalia, que por definição é um caso raro. Dentro desse contexto, existem 3 cenários possíveis na operação da Detecção de Anomalias: Supervisionado, Semi-Supervisionado, Não Supervisionado.

Os algoritmos de Detecção de Anomalia Supervisionados têm poucos exemplos de anomalias, por definição. Isso pode trazer problemas quanto a análise da acurácia. Para ilustrar, considere o caso extremo: Considerar todos normais. Isso resultaria em uma

acurácia alta, uma vez que os exemplos anômalos podem representar menos de 1% do conjunto de exemplos. Para se resolver isso, observa-se a precisão e a exaustividade que resultariam 0%. Como é preciso se chegar em um número para poder comparar performance, Witten[10] propõe que se avalie o *F1 score*, que seria a média harmônica das duas medidas.

Assume-se que os exemplos de treinamento são etiquetados como normais quando se trabalha no cenário Semi-Supervisionados desse modo é possível criar uma estimativa de como se comportam os exemplos normais e assumir que os que fogem do escopo é anomalia.

Por último, quando não se tem etiqueta para os exemplos opera-se num cenário Não Supervisionado, assume-se que os casos anômalos são menos frequentes. Note que se essa premissa não for satisfeita, o sistema está sujeito à alta quantidade falsos negativos.

3.4.2 Proposta de Modelo

Será exposto um modelo de Detecção de Anomalia baseado em modelo probabilístico. Suponha que os dados de um conjunto de exemplos se distribuam no plano determinado pelas características de acordo com o padrão normal. Dessa forma é possível associar cada exemplo $\mathbf{x}$ à uma probabilidade, $p(\mathbf{x})$. Sabe-se[7] que a distribuição normal multidimensional é

$$\mathcal{N}(\mathbf{x}|\mu, \mathbf{\Sigma}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)\right\}$$

O qual D é a dimenção de $\mathbf{x}$, μ é o vetor média de todas as caracteríticas da Matriz $\mathbf{X}$ e $\mathbf{\Sigma}$ é uma matriz covariância, $D \times D$, $|\mathbf{\Sigma}|$ é seu determinante.

$$\mathbf{\Sigma} = cov[\mathbf{x}] = \mathbb{E}\left[(\mathbf{x} - \mathbb{E}[\mathbf{x}])(x - \mathbb{E}[\mathbf{x}])^T\right]$$

O qual $\mathbb{E}$ é o operador valor esperado. Após a construção do modelo probabilístico, o algoritmo decide se um novo exemplo $\mathbf{x}$ é uma anomalia se

$$\mathcal{N}(\mathbf{x}|\mu, \mathbf{\Sigma}) < \varepsilon$$

O qual ε é um valor limiar escolhido previamente que resultou o melhor *F1 score*, se o modelo for aplicado em um cenário Supervisionado.

4 Preparação Dos Dados

4.1 Levantamento das Possíveis Features Relevantes

De posse das informações apresentadas nos capítulos anteriores, é possível definir o problema que esse trabalho se propõe a resolver e, em linhas gerais, como ele será abordado.

O objetivo desse trabalho é a partir dos registros de um servidor DNS, acusar quais máquinas na rede são suspeitos de pertencer a uma botnet e merecem atenção para uma investigação. A caracterização dos IPs será feita através da análise das características da iteração que seriam.

Afim de levantar essas características três comportamentos divergentes do uso comum foram observados: a escolha dos domínios, o comportamento de máquina, os domínios visitados em comum.

4.1.1 Escolha dos Domínios

Domínios gerados automaticamente é uma prática comum entre as redes de botnets. Essa prática gera nomes provavelmente não inteligíveis podendo inclusive trazer números. Além disso, por conveniência é possível que os domínios criados tendem a ter o mesmo número caractéres. Por outro lado, apenas 7.3% dos domínios dos um milhão primeiros domínios da Alexa contem número e que o tamanho do domínio de um usuário comum não segue nenhum padrão específico.

Não há garantia que essa feature é sempre efetiva, dado que o atacante pode gerar domínios de tamanho variável e evitar números ao gerar o domínio, por isso após a realização dos experiementos será possível confirmar ou não essa hipótese.

Para explorar essas propriedades foram propostas as seguintes caracteríticas

- Quantidade de consultas a domínios com alta quantidade de números
- Média do comprimento de domínios consultados
- Desvio Padrão dos comprimentos dos domínios consultados

4.1.2 Comportamento de Máquina

O tempo de reação de um humano a uma requisição sem sucesso não pode ser de décimos de segundo, por mera limitação de reflexo. Qualquer sinal de uso que aparente como uma máquina como essa deve ser considerada suspeita. Além disso, é possível que a

máquina acabe por visitar uma quantidade de domínios maior do que o normal. Esse tipo de feature necessita ser validada, pois o comportamento de máquina pode ser mascarado.

Para explorar essas propriedades foram propostas as seguintes caracteríticas

- Média do intervalo entre as consultas
- Desvio padrão dos intervalos entre consultas
- Quantidade total de consultas realizadas

4.1.3 Domínio Visitados em Comum

Do fato de que domínios suspeitos são acessados por poucas máquinas: As botnets costumam utilizar algoritmos de geração de domínios para tentar estabelecer uma comunicação com o C&C. Devido a isso, espera-se que os bots tentem acessar domínios que dificilmente serão procurados por máquinas normais, além do que, se a máquina infectada procura o centro de comando e controle, espera-se que muito dos domínios consultados sejam desse forma, pouco procurado por outras máquinas.

Para analisar essa propriedade, recomenda-se realizar um pré-processamento, que analisa para cada domínio consultado, por quantas máquinas diferentes ele foi consultado. Essa quantidade de máquinas que requisitou um domínio específico chamaremos de grau de requisição do domínio.

Dessa forma, acredita-se que as informações relativas ao grau dos domínios consultados pela máquina podem ser úteis para identificar um comportamento suspeito em uma máquina. Foram levantadas as seguintes característica:

- Grau de requisição mínimo entre os graus dos domínios consultados pela máquina
- Média dos graus de requisição dos domínios consultados pela máquina
- Desvio Padrão dos graus de requisição dos domínios consultados pela máquina

4.1.4 Experimentos

Não se tem garantia alguma de alteração no padrão dos tipos de consulta. Porém, esse dado é de fácil acesso e seu estudo pode evidenciar a exploração de alguma outra fragilidade nas requisições DNS.

- Quantidade de consultas para cada tipo de DNS
- Porcentagem de consultas para cada tipo de DNS

Após o levantamento de todas as máquinas da rede, é feita a construção do modelo probabilístico gaussiano e de acordo após a escolha da probabilidade mínima dos exemplos normais, o sistema estará pronto para decidir se uma nova máquina poderia pertencer ou não a uma botnet.

5 Cronograma

Embora o objetivo do trabalho seja o desenvolvimento de um projeto e não pesquisa, foi preciso começar por um intenso estudo do problema que queríamos resolver, ou seja das botnets. Isso é uma etapa importante de um projeto de aprendizagem de máquina, pois permite uma melhor identificação de quais *features* serão mais relevantes.

Em seguida, foi feito um estudo dos algoritmos de *clustering* existentes e os objetivos de cada técnica.

Após esses estudos, começa a implementação do sistema detector de botnets. A primeira etapa é desenvolver um tratamento automatizado dos dados obtidos pela coleta dos logs DNS, já que o objetivo final é de que esse tratamento seja feito diariamente. Depois serão implementados algoritmos de agrupamento que usarão os dados tratados para identificar padrões de botnets no log DNS coletado. Por fim, os resultados dessas técnicas serão testados e analisados e servirão de motivação para possíveis refinamentos nos algoritmos.

Durante essas tarefas, desenvolveremos também os relatórios e apresentações para as seguintes avaliações:

- Verificação Especial em Maio,
- Verificação Corrente em Julho,
- Verificação Final em Setembro.

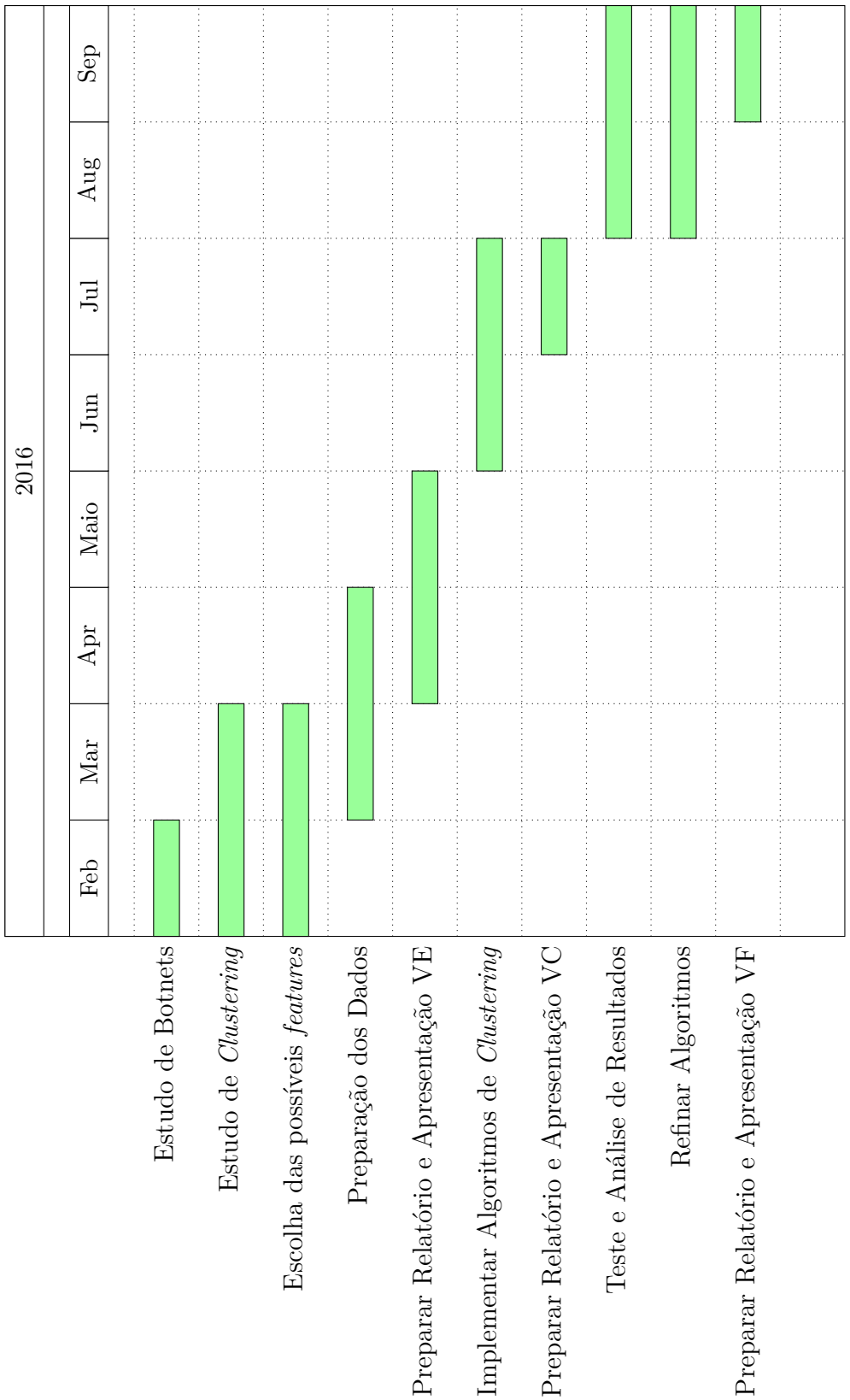

Figura 5 – Cronograma

6 Conclusão

Após os estudos apresentados, concluiu-se que pela raridade de ocorrência e tendência de anomalia nos comportamentos, o melhor tipo de modelo seria Detecção de Anomalia. Atualmente, há base de dados de IPs infectados etiquetados manualmente, porém é possível que se migre para o cenário Semi-Supervisionado, se for provado que há necessidade de mais dados. A aquisição de amostras é facilitada para este trabalho, pois dispõe-se um servidor de DNS no Instituto Militar de Engenharia que pode fornecer dados.

Além disso, não há garantias de que o melhor modelo de distribuição de probabilidade poderia ser o Gaussiano. Porém é o escolhido por ser o mais simples. Alternativo a ele, pode-se usar o modelo de Misturas de Gaussianas, caso existam regiões não contíguas de máquinas em operação normal.

A escolha dos modelos de distribuição de probabilidade, a validação das características levantas e a constatação da necessidade de mais amostras fazem parte dos trabalhos futuros.

Referências

1 SILVA, S. S. et al. Botnets: A survey. *Computer Networks*, Elsevier, v. 57, n. 2, p. 378–403, 2013.

2 JI, S. et al. Botnet detection and response architecture for offering secure internet services. In: IEEE. *Security Technology, 2008. SECTECH'08. International Conference on*. [S.l.], 2008. p. 101–104.

3 SILVA, S. S.; SALLES, R. M. Arquitetura de um sistema integrado de defesa cibernética para detecção de botnets. *Programa de Engenharia de Defesa*, Instituto Militar de Engenharia, 2012.

4 SOLTANI, S. et al. A survey on real world botnets and detection mechanisms. *International Journal of Information and Network Security*, IAES Institute of Advanced Engineering and Science, v. 3, n. 2, p. 116, 2014.

5 FEILY, M.; SHAHRESTANI, A.; RAMADASS, S. A survey of botnet and botnet detection. In: IEEE. *Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09. Third International Conference on*. [S.l.], 2009. p. 268–273.

6 WANG, P.; SPARKS, S.; ZOU, C. C. An advanced hybrid peer-to-peer botnet. *IEEE Transactions on Dependable and Secure Computing*, v. 7, n. 2, p. 113, 2010.

7 BISHOP, C. M. Pattern recognition. *Machine Learning*, 2006.

8 MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of machine learning*. [S.l.]: MIT press, 2012.

9 CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, ACM, v. 41, n. 3, p. 15, 2009.

10 WITTEN, I. H. F. et al. *Data mining: practical machine learning tools and techniques*. [S.l.], 2011.