MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
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
5 de maio de 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
5 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.

27f. : 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, 5 de maio de 2016:

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

Profa. Raquel Coelho Gomes Pinto
Convidada, D. Sc., do IME

Prof. Julio Cesar Duarte
Convidado, D. Sc., do IME

Rio de Janeiro
5 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 de spams 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 detectar máquinas que pertencem a botnets a partir de algoritmos orientados à Detecção de Anomalia.

Palavras-chave: botnets, 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 still sensible information, sending spams to common users and sending sucessives requests to get servers down. Because it is a illegal activity, the criminals do the comunication between the machines with divergents behaviours. Based on that premise, this project proposes to detect machines that are part of a botnet using Anomaly Detection oriented algorithm.

Keywords: botnets, 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*

Sumário

1 Introdução

1.1 Contextualização

1.2 Objetivo

O objetivo deste trabalho é desenvolver e analisar uma ferrementa para detectar possíveis hospedeiros de bots em uma botnet, utilizando algoritmos de agrupamento que utilizarão dados de consultas de DNS realizadas. Para esta análise, serão utilizados logs de consulta com os hospedeiros previamente mapeados.

1.3 Motivação

1.4 Justificativa

A justificativa desse trabalho é reduzir o trabalho manual de identificar possíveis botnets em uma rede, funcionando como um filtro que já informará com maior velocidade as máquinas suspeitas. Por isso, os algoritmos de agrupamento se destacam para auxiliar nesse processo, além de poder servir no futuro como um módulo de um sistema integrado de detecção de botnets[5].

1.5 Metodologia

1.6 Estrutura

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, e leve em conta as configurações existentes e até detecte possíveis novas configurações das botnets, é preciso compreender o funcionamento das botnets e seus objetivos, para que possamos identificar características constantes na botnet, mesmo quando o botmaster está tentando evitar os mecanismos de detecção.

2.1 Elementos das Botnets

Estruturalmente, as botnets são formadas pelos bots, que 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.

O canal de C&C é o meio que o botmaster tem para se comunicar com a sua botnet, e é a parte chave do funcionamento, pois é necessário para o envio dos comandos necessários para a atividade maliciosa aos hospedeiros. Dessa forma, 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 à enlaces de alta velocidade e rodando sistemas com vulnerabilidades consideráveis, criou um ambiente favorável à formação de botnets. Esse crescimento, aliado à alta efetividade e potencial de causar danos, fez com que as botnets se tornassem um dos maiores desafios de pesquisa em segurança no espaço cibernético atual. [2]

Existem características que tornam o host mais interessantes 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 o combate das botnets: reativamente ou preventivamente. A forma reativa é a mais comum e envolve detectar a existência da botnet e reagir ao ataque tentando reduzir o tráfico 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 reativa, já que o algoritmo encontrará padrões em botnets que já estão atuando. Porém, uma característica desejável para um detector reativo é a detecção em tempo real, com o objetivo de minimizar os danos causados e o tempo de reação do botmaster. Porém, essa característica é um desafio, devido ao grande número de dados que devem ser tratados e analisados. Dessa forma, nosso objetivo neste projeto será 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 1 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 infecta-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 tornam o host em um bot real.[3].

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 segue 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

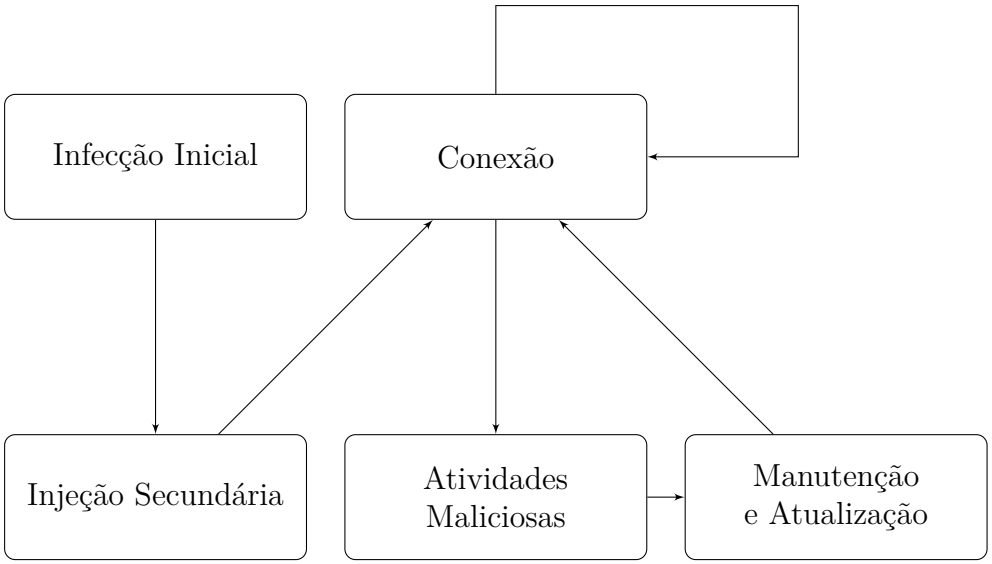

Figura 1 – Ciclo de Vida das Botnets

atividades maliciosas solicitadas. A 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 2 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 ela seja desligada após a identificação dos poucos pontos centrais de C&C. Ela é muito utilizada pelo protocolo IRC (*Internet Relay Chat*), porém pelo fato de tráfego desse protocolo ser incomum e raramente utilizado, ele costuma ser bloqueado, inutilizando a botnet. Por isso, o uso do protocolo HTTP (*HyperText Transfer Protocol*) se popularizou já que ele é muito 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 arquiterua, permite que mesmo que muitos bots sejam desativados a botnet possa continuar funcionando, já que não existem pontos centralizados de C&C.

As arquitetura híbridas apresentam características de ambas as arquiteturas centralizadas e descentralizadas, como mostrado na Figura 3, na qual os bots são classificados

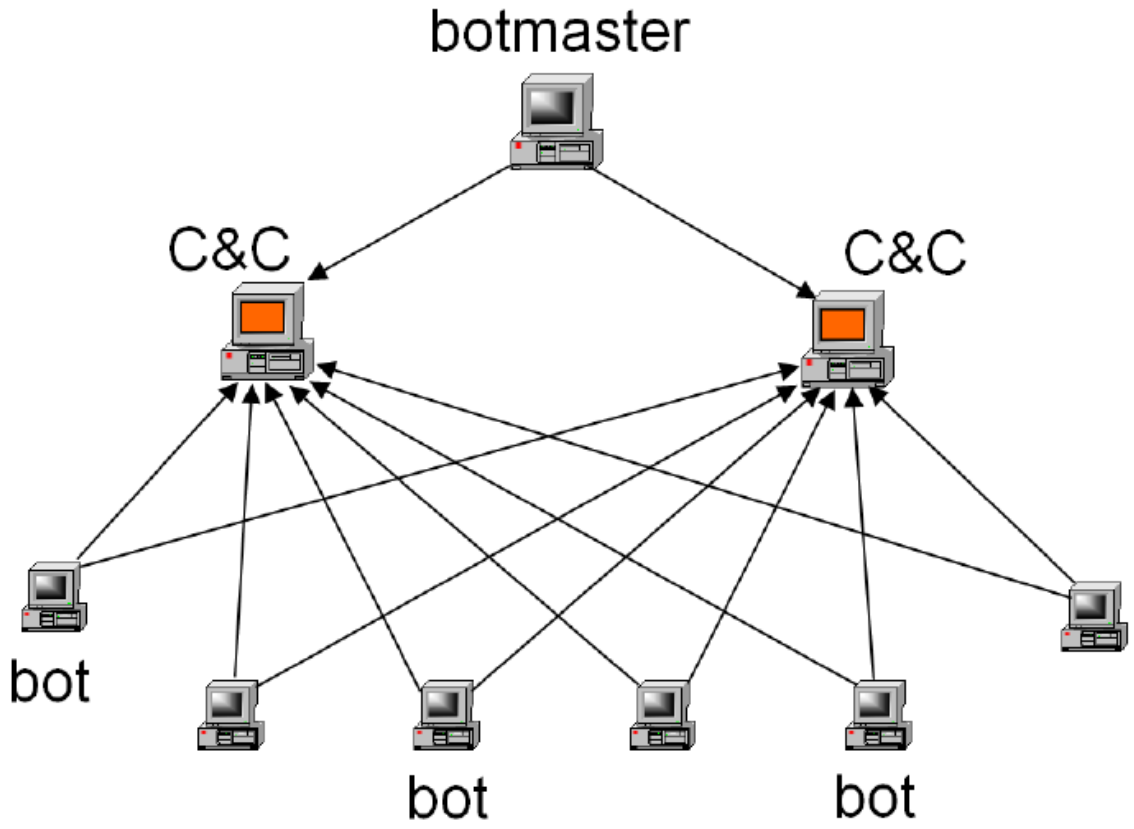

Figura 2 – Arquitetura Centralizada[4]

em dois grupos: clientes e servos. Os servos 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 denomidados clientes pois não aceitam comunicações de entrada, dessam 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 teórico, no qual o bot não se comunica ativamente com o botmaster ou com outros bot, para realizar um ataque o botmaster vasculha a rede em busca de um bot para enviar o comando e realizar as atividades maliciosas.

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

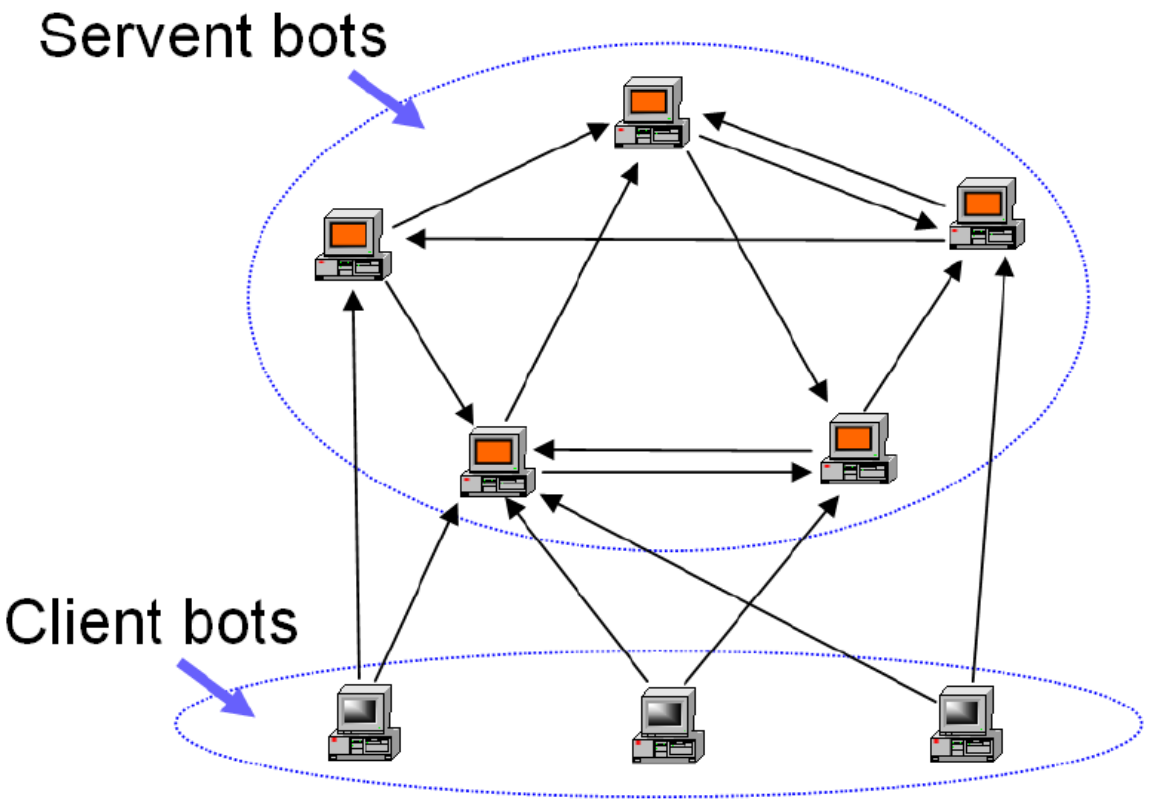

Figura 3 – Arquitetura Híbrida[4]

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 botmasters 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 botmasters 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 Aprendizagem de Máquina

Como Bishop [**??**] descreve, aprendizagem 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 que 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.

3.1.1 Amostra

Seja amostra o conjunto de exemplos que representam os dados conhecidos do problema.

3.1.2 Características

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

3.1.3 Etiquetas

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

3.1.4 Avaliações de Desempenho

Define-se acurácia como:

$$\frac{\#\text{Acertos}}{\#\text{Amostra}}$$

Define-se precisão como:

$$\frac{\text{VerdadeirosPositivos}}{\#\text{Positivos}}$$

Define-se *recall* como:

$$\frac{\text{VerdadeirosPositivos}}{\#\text{Acertos}}$$

Define-se *F1 score* como:

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

3.1.5 Matriz $\mathbf{X}$

Seja $\mathbf{X}$, amostra 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 da amostra.

3.1.6 Vetor $\mathbf{Y}$

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.

3.2 Categorias de Problemas

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

Classificação: Decidir a classe de exemplo dadas as 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.

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

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

Redução de Dimensionalidade: Representar a amostra com número reduzido de dimensões, por exemplo comprimindo imagens para processamento de imagens.

Neste projeto foi decidido encarar o problema como aglutinação, 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[**??**] sete cenários para os algoritmos de aprendizagem, esse cenários são fortemente influenciados pelas condições dos dados de treinamento.

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

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

Aprendizado Semi-Supervisionado: Neste cenário, é possível acessar uma amostra 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 outras possíveis cenários ainda mais complexos e específicos, não catalogados neste trabalho porque a ciência de Aprendizado de Máquina está ainda em constante fase de crescimento.

3.4 Exemplo: Regressão Supervisionada

Dado um conjunto de tamanho de imóveis e seus respectivos valores de compra em uma vizinhança hipotética estimar o valor de um apartamento sabendo apenas seu tamanho.

Seja o vetor ordenado $\mathbf{X} = [x_1, x_2, ..., x_m]$, $\#\mathbf{X} = m$, o tamanho em Metros quadrados de apartamentos em um bairro. Seja $\mathbf{Y} = [y_1, y_2, ..., y_m]$, o preço do i-ésimo apartamento indexado em $\mathbf{X}$. Se arranjarmos os valores graficamente obtem-se a Figura 1.

Suponha que de posse do tamanho x, não apresentado em $\mathbf{X}$, o usuário queira inferir um preço y. Para tal é necessário construir uma hipotese h_θ que melhor interpole os

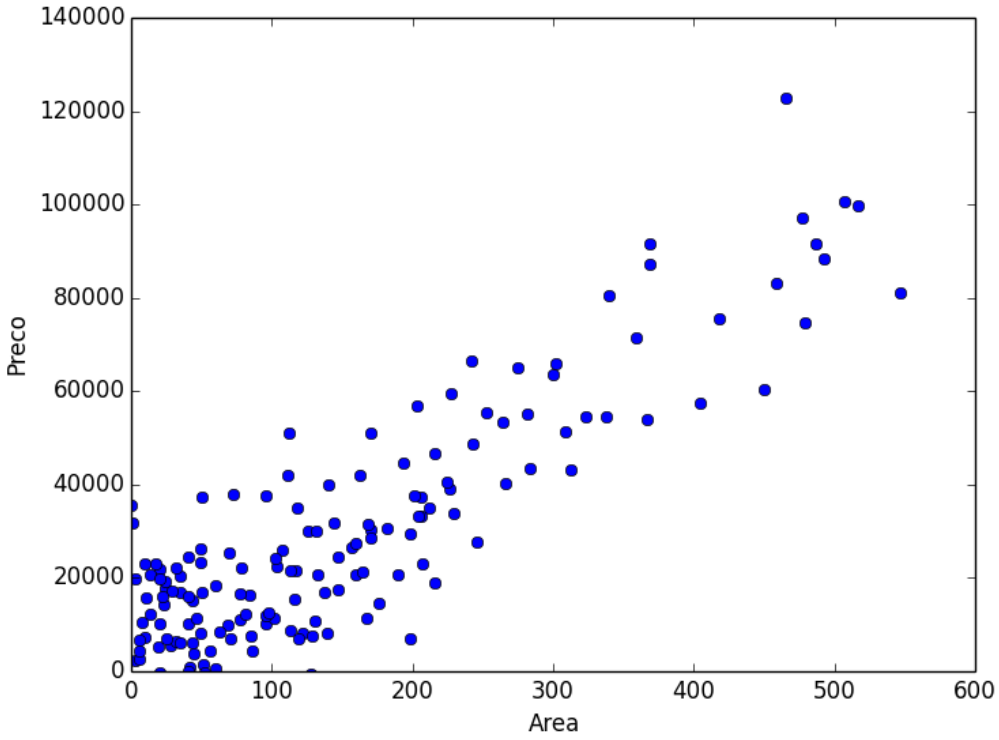

Figura 4 – A distribuição do preço das casas

valores de já conhecidos. Para poder avaliar se h_θ é ótimo, existe uma função custo, $custo(h_\theta, \mathbf{Y})$.

Para esse problema, considera-se:

$$h_\theta(x) = \theta_1 * x + \theta_0$$

Ou seja, um modelo linear. Além disso, para avaliar a proximidade do modelo e da amostra, aplica-se o erro quadrádico médio como função custo, isto é:

$$custo(h_\theta(X), \mathbf{Y}) = \frac{1}{m} * \sum_{i=1}^{m} (h(x_i) - y_i)^2$$

Tem-se dados, função de custo, precisa-se introduzir um método de otimização. Propõe-se, então, o uso Método do Gradiente, que é um método para achar o mínimo local dado um ponto. Como a função de custo em questão é quadrática, ou seja, é convexa. Existe α tal que atualizando iterativamente os parâmetros θ_0 e θ_1, seguindo a expressão

$$\theta_0 = \theta_0 - \alpha * \frac{\partial}{\partial \theta_0} custo(h_\theta(X), Y)$$

$$\theta_1 = \theta_1 - \alpha * \frac{\partial}{\partial \theta_1} custo(h_\theta(X), Y)$$

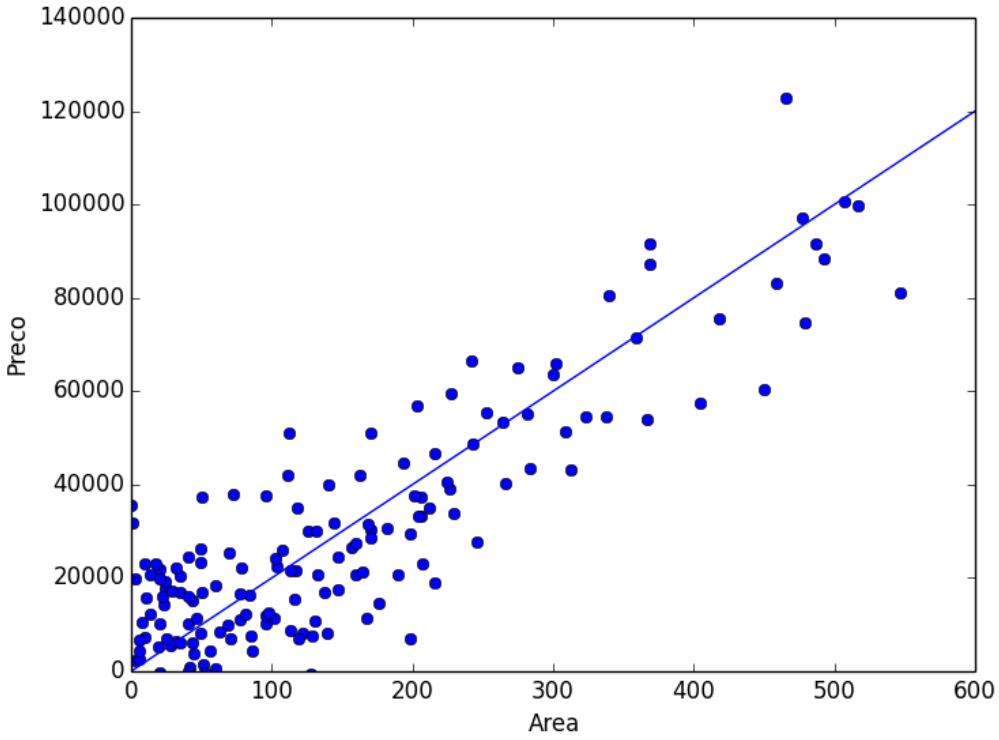

Figura 5 – O modelo acompanhado e a distribuição do preço das casas

Converge-se à solução ótima que minimiza o valor da função custo, sendo possível com algum erro inferir o valor do imóvel sabendo seu tamanho. Tal hipótese h_θ está representado na Figura 2

3.5 Detecção de Anomalia

O artigo, *Anomaly Detection: A Survey*[**??**], define Detecção de Anomalia o problema da busca por padrões nos dados que não seguem um padrão de um comportamento 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. Esse tipo de algoritmo pode ser utilizado para resolver problemas de Classificação e Aglutinação.

3.5.1 Etiquetas em Detecção de Anomalia

Geralmente a tarefa 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 convertem têm poucos exemplos de anomalias, por definição, que isso pode trazer problemas quanto a análise da acurácia. O caso extremo seria considerar todos normais e ter acurácia de alta, uma vez que os exemplos anomalos podem representar menos de 1% da amostra. Isso pode ser resolvido observando a precisão e a exaustividade que resultariam 0%. Como é preciso se chegar em um número para poder comparar performance, Witten[**??**] propõe que se avalie o *F1 score*, que seria a média harmônica das duas medidas.

Assume-se que as amostras de treinamento são exemplos 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 anoma-los são menos frequentes. Note que se essa premissa não for satisfeita, o sistema está sujeito à alta quantidade falsos negativos.

3.5.2 Proposta de Modelo

Será exposto um modelo de Detecção de Anomalia baseado em modelo probabilístico. Suponha que os dados de uma determinada amostra 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[**??**] 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 tal um novo exemplo $\mathbf{x}$ é uma anomalia se

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

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

4 O Problema

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 abordados. 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

- Quantidade de consultas a domínios com sufixos suspeitos;
- Quantidade de consultas a domínios com alta quantidade de números;
- Quantidade de domínios consultadas com string legível pequena;
- Quantidade de consultas a domínios com baixo grau no Alexa;
- Quantidade total de consultas realizadas;
- Média do comprimento de domínios consultados;
- Desvio Padrão dos comprimentos dos domínios consultados;
- Média do intervalo entre as consultas;
- Desvio padrão dos intervalos entre consultas;
- Total de consultas que resultaram em NXDOMAIN;
- Porcentagem de consultas que resultaram em NXDOMAIN;
- Quantidade de consultas para cada tipo de DNS;
- Porcentagem de consultas para cada tipo de DNS;
- Tamanho do menor clique que a máquina participa;
- Tamanho médio dos cliques em que a máquina participa.

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.

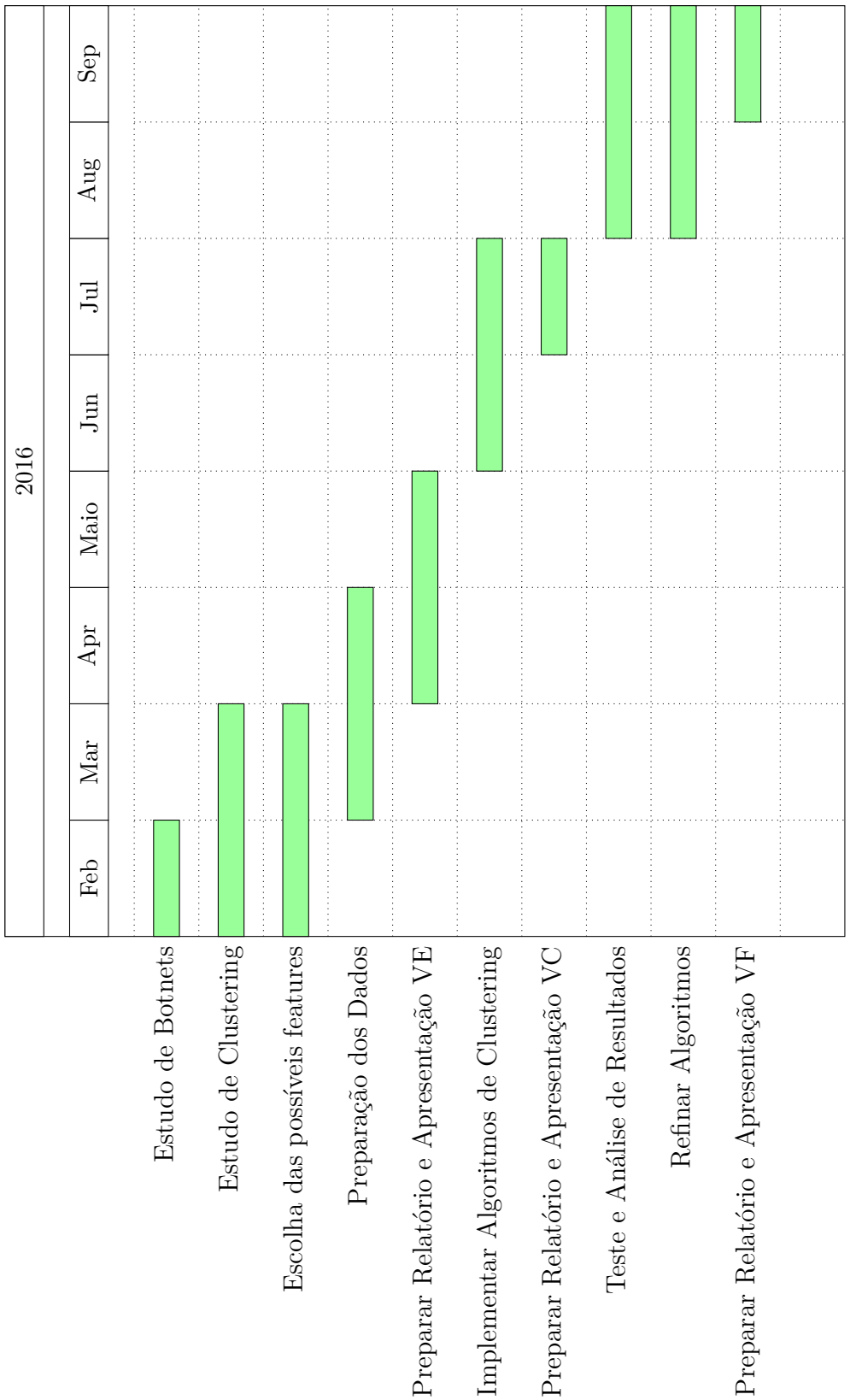

Figura 6 – Cronograma

6 Conclusão

Após os estudos apresentados, concluiu-se que pela raridade de ocorrência e tendência anomalia nos comportamentos, o melhor tipo de modelo seria Detecção de Anomalia. Atualmente há base de dados 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 que pode fornecer dados.

Além disso dos dados, 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 Mixturas de Gaussianas, caso existam regiões não contíguas de máquinas em operação normal.

A escolha dos modelos de distribuição de probabilidade, a validação dos 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 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.

3 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.

4 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.

5 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.