UNIVERSIDADE ESTADUAL DO CEARÁ

CENTRO DE CIÊNCIAS E TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

LUIZ GONZAGA MOTA BARBOSA

UMA ARQUITETURA PARA DETECÇÃO DE BOTNETS BASEADA NA ANÁLISE DO TRÁFEGO DNS DESCARTADO

FORTALEZA – CEARÁ

2017

LUIZ GONZAGA MOTA BARBOSA

UMA ARQUITETURA PARA DETECÇÃO DE BOTNETS BASEADA NA ANÁLISE DO TRÁFEGO DNS DESCARTADO

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Segurança em Redes

Orientador: Prof. Dr. Joaquim Celestino Junior

Co-Orientador: Prof. Dr. André Luiz Moura dos Santos

FORTALEZA – CEARÁ

2017

Dados Internacionais de Catalogação na Publicação

Universidade Estadual do Ceará

Sistema de Bibliotecas

Barbosa, Luiz Gonzaga Mota.
Uma arquitetura para detecção de botnets baseada na análise do tráfego DNS descartado [recurso eletrônico] / Luiz Gonzaga Mota Barbosa. – 2017.
1 CD-ROM: il.; 4 ¾ pol.

CD-ROM contendo o arquivo no formato PDF do trabalho acadêmico com 57 folhas, acondicionado em caixa de DVD Slim (19 x 14 cm x 7 mm).

Dissertação (mestrado acadêmico) – Universidade Estadual do Ceará, Centro de Ciências e Tecnologia, Mestrado Acadêmico em Ciência da Computação, Fortaleza, 2017.
Área de concentração: Segurança em Redes.
Orientação: Prof. Ph.D. Joaquim Celestino Júnior.
Coorientação: Prof. Dr. André Luiz Moura dos Santos.

1. Botnets. 2. Detecção. 3. Algoritmos Geradores de Domínio. I. Título.

LUIZ GONZAGA MOTA BARBOSA

UMA ARQUITETURA PARA DETECÇÃO DE BOTNETS BASEADA NA ANÁLISE DO TRÁFEGO DNS DESCARTADO

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Segurança em Redes

Aprovada em: 20 de Março de 2017

BANCA EXAMINADORA

Prof. Dr. Joaquim Celestino Júnior
(Orientador/UECE)

Prof. Dr. André Luiz Moura dos Santos
(Coorientador/UECE)

Prof. Dr. Marcial Porto Fernandez
(UECE)

Prof. Dr. Márcio Espíndola Freire Maia
(UFC)

AGRADECIMENTOS

Primeiramente a Deus, pela capacidade de buscar e alcançar meus desafios e por mais essa conquista.

À minha mãe que sempre me proporcionou uma excelente educação e condições de estudos.

À minha esposa, Danielle, que acompanha minha jornada há uma longa data e que esteve ao meu lado dando apoio e motivação. Por sua paciência nos momentos mais tensos e de estresse. E por todo o seu amor.

Ao prof. Dr. Joaquim Celestino que foi fundamental para a conclusão deste trabalho. Foi uma honra trabalhar mais diretamente com o senhor.

Ao prof Dr. André Santos por todas as oportunidades oferecidas, tanto no Insert quanto no início do mestrado. Uma honra ter trabalhado tantos anos com o senhor.

Aos demais membros da banca, prof. Dr. Marcial Fernández e prof. Dr. Márcio Maia, por terem aceitado o convite em avaliar meu trabalho, pelas críticas e sugestões repassadas.

A todos os professores do MACC por compartilhar seu conhecimento e experiências acadêmicas conosco.

Aos meus colegas de trabalho na DSEG-UFC, pela força nesse período e por todas as vezes em que me deram cobertura.

Ao Amarildo Rolim pelo apoio e pela flexibilidade para concluir minha pesquisa.

Ao Márcio André por todas as trocas de experiências em nossos trabalhos de dissertação. Foram de grande relevância para a resolução de questões ligadas a este trabalho.

A simplicidade é Divina. (Edsger W. Dijkstra)

RESUMO

Botnets são uma das principais ameaças na Internet e capazes de auxiliar na realização de atividades maliciosas. Ao infectar um hospedeiro, uma das primeiras ações de um bot é a comunicação com o servidor de comando e controle. Para este fim, bots mais recentes utilizam algoritmos que geram uma lista de nomes de domínios candidatos a servidores de comando e controle. Uma consequência deste comportamtento é o aumento de respostas negativas do protocolo DNS. Geralmente esse tráfego negativo é simplesmente descartado ou ignorado pelos administradores de rede. Pesquisas recentes relacionadas à detecção de botnets analisam o comportamento da rede em busca de evidências da presença de bots e a partir daí extrair característica a fim de modelar essas infecções e eliminá-las da rede. Com base no aumento no número de respostas negativas do protocolo DNS e acesso a amostras coletadas em um ambiente controlado, este trabalho apresenta uma metodologia capaz de gerar modelos de detecção para estes bots.

Palavras-chave: Botnets. Detecção. Algoritmos Geradores de Domínios.

ABSTRACT

Botnets are one of the main threats on the Internet and are able to assist in performing malicious activities. After a host infection, one of the first actions taken by a bot is the communicating with its command and control servers. To this purpose, most recent bot use algorithms to generate a list of candidate domain names for command and control servers. One consequence of this behavior is the increase on negative DNS answers. Usually, this traffic is discarded or ignored by network administrators. Recent research related to botnet detection make an analysis of the network behavior looking for evidencies of the presence of bots and then extract characteristics to model these infections and eliminate them from the network. Based on the increase in the number os negative DNS answers and having access to samples collected from a controled environment, this work propose a methodology capable of generating detection models to these bots.

Keywords: Botnets. Detection. Domain Generation Algorthms

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

2LD	Second Level Domain
3LD	Third Level Domain
C&C	Canal de Comando e Controle
CDN	Content Distribution Networks
DGA	Domain Generation Algorithm
DNS	Domain Name System
FFSN	Fast-Flux Service Networks
HTTP	HyperText Transfer Protocol
IRC	Internet Relay Chat
P2P	Peer-to-peer
PCAP	Packet Capture Library
RRDNS	Round-Robin DNS
TTL	Time-To-Live

SUMÁRIO

1 INTRODUÇÃO

Botnets tornaram-se uma das principais ameaças na Internet, sendo capazes de auxiliar na realização de diversas atividades maliciosas como ataques de negação de serviço, roubo de informações sensíveis, envio massivo de e-mails (spams) entre outras (CAVALLARO; KRUEGEL; VIGNA, 2009; GU *et al.*, 2008).

O impacto causado pela disseminação de botnets varia desde o consumo de recursos computacionais da vítima, a prejuízos financeiros (WUEEST, 2014; WUEEST, 2015). Por exemplo, em 2013, uma botnet foi utilizada para roubar mais de 250.000 dólares de instituições financeiras e seus clientes (SOPHOS, 2014). É evidente a necessidade de combater essa ameaça e diversos pesquisadores têm empenhado grande esforço no desenvolvimento de técnicas para tal fim.

Pesquisas recentes, relacionadas à detecção de botnets, analisam o comportamento da rede em busca de evidências da presença de bots e a partir daí extraem características como a quantidade de máquinas infectadas, a quantidade de servidores de comando e controle (C&C) identificados, o consumo de largura de banda e o protocolo utilizado a fim de modelar essas infecções e eliminá-las da rede (CAVALLARO; KRUEGEL; VIGNA, 2009; GU *et al.*, 2008; ANTONAKAKIS *et al.*, 2012; BILGE *et al.*, 2012).

1.1 MOTIVAÇÃO

Uma das primeiras ações executadas por um bot em um hospedeiro infectado é tentar localizar seu servidor de comando e controle. Para isso, bots mais recentes tem utilizado o protocolo DNS, desfrutando das mesmas vantagens que usuários legítimos.

Uma maneira de contactar seus servidores é carregar o domínio destes em seu código. Porém, para dificultar a detecção a partir de técnicas de engenharia reversa, desenvolvedores de bots passaram a utilizar Algoritmos de Geração de Domínios (*Domain Generation Algorithms - DGAs*) com a finalidade de criar uma lista de candidatos a servidores C&C para o estabelecimento de sua comunicação (STONE-GROSS *et al.*, 2009; BILGE *et al.*, 2011; ANTONAKAKIS *et al.*, 2012).

A utilização de DGAs ocasiona um aumento na quantidade respostas negativas do protocolo DNS na rede e esse tráfego é normalmente descartado. Entretanto, esse tráfego descartado pode ser utilizado para modelar a presença de infecções na rede e gerar perfis de comportamento dos bots presentes na rede.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Apresentar uma metodologia para detecção de botnets com base na análise do tráfego do protocolo DNS que seria descartado pela rede, em especial daquelas que utilizam Algoritmos de Geração de Domínios para a localização de seus servidores de comando e controle.

1.2.2 Objetivos Específicos

a) Analisar o Estado-da-Arte quanto às técnicas existentes;
b) Identificar um conjunto de características básicas para modelar o comportamento dos bots quanto ao protocolo DNS, especificamente domínios gerados por DGAs;
c) Utilizar técnicas de aprendizagem de máquina para a modelagem do comportamento malicioso;
d) Propor uma metodologia para a detecção de botnets com base nas características e técnicas estudadas.

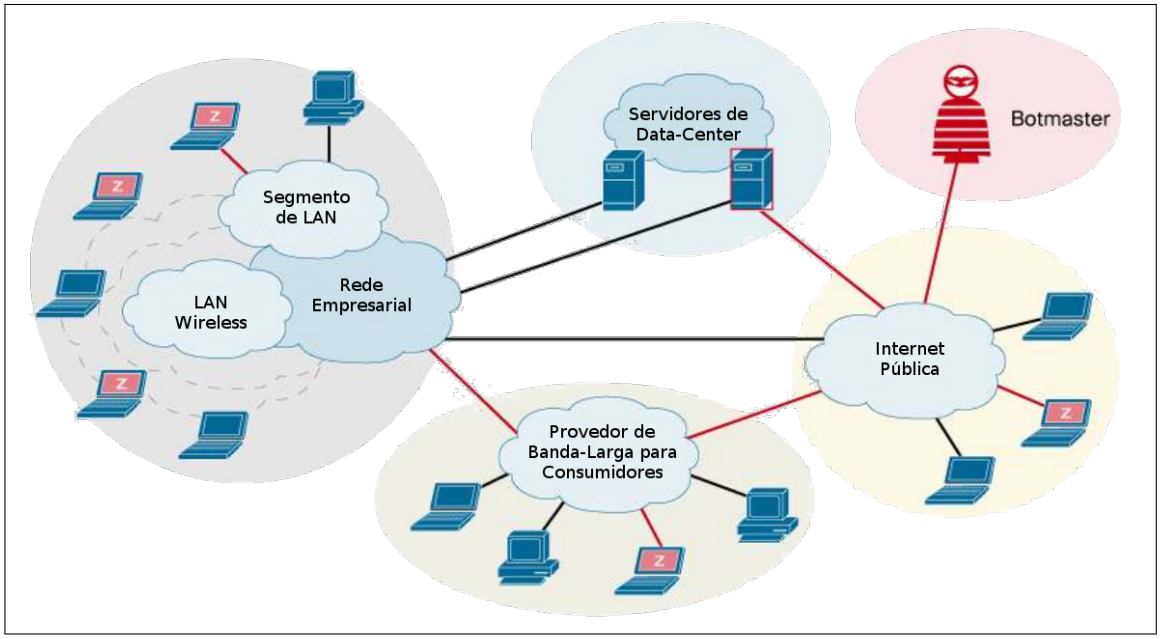
2 FUNDAMENTAÇÃO TEÓRICA

2.1 BOTNETS

Botnets tornaram-se uma das principais ameaças na Internet e, nos últimos anos, a explosão massiva dos dispositivos móveis fez com que a evolução de malwares para dispositivos móveis tenha acompanhado a evolução de malwares para computadores pessoais (GU *et al.*, 2008; CAVALLARO; KRUEGEL; VIGNA, 2009; NIGAM, 2014).

Botnets são grupos de máquinas comprometidas por malwares, conhecidas como *bots* ou zumbis, controladas remotamente por um indivíduo chamado *botmaster* através de um canal de comunicação de comando e controle (C&C) (CISCO, 2007; ANTONAKAKIS *et al.*, 2012). Um mesmo bot pode apresentar características de diferentes tipos de malwares tornando as botnets capazes de realizar diversas atividades maliciosas tais como: ataques de negação de serviço, roubo de informações sensíveis, envio massivo de e-mails (spams) entre outras (GU *et al.*, 2008; CAVALLARO; KRUEGEL; VIGNA, 2009; TIIRMAA-KLAAR *et al.*, 2013). Além disso, bots podem infectar diferentes tipos de máquinas vulneráveis espalhadas em uma mesma rede ou em redes distintas, como mostrado na Figura 1.

Figura 1 – Ilustração de uma Botnet.

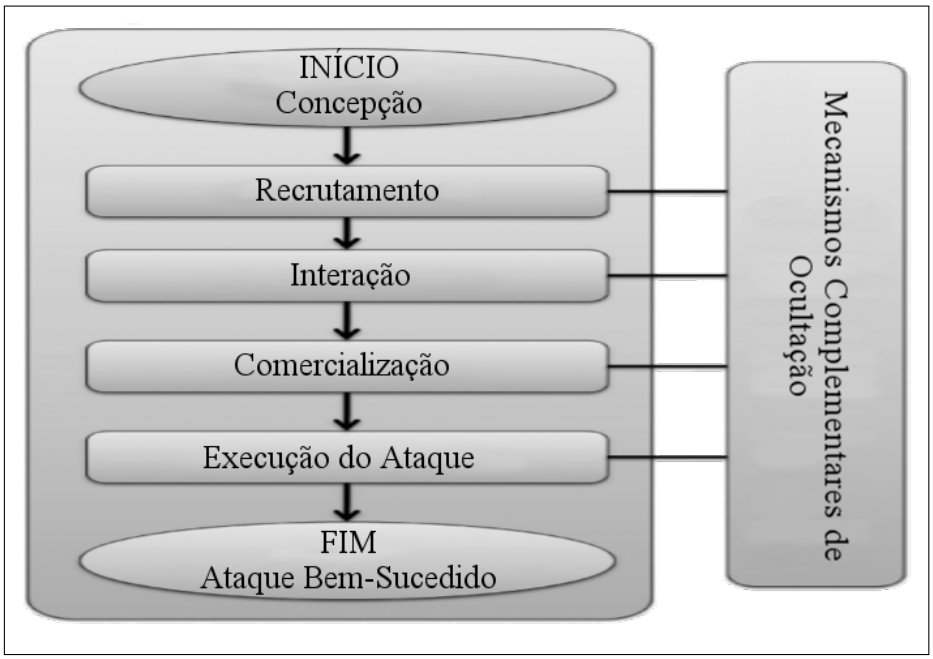


Fonte**:** (CISCO, 2007) - adaptado.

2.1.1 Ciclo de vida de uma botnet

Botnets seguem uma espécie de ciclo de vida linear desde sua criação até o final do ataque (RODRÍGUEZ-GóMEZ; MACIá-FERNáNDEZ; GARCÍA-TEODORO, 2013). A Figura 2 traz os estágios deste ciclo e a relação entre eles.

Figura 2 – Ciclo de vida de uma botnet.



Fonte**:** (RODRÍGUEZ-GóMEZ; MACIá-FERNáNDEZ; GARCÍA-TEODORO, 2013) - adaptado.

Os estágios desse ciclo são brevemente descritos a seguir:

- **Concepção:** nesse estágio são determinadas as principais características da botnet como arquitetura, protocolo utilizado, vulnerabilidades a serem exploradas, entre outras, levando em consideração as intenções do botmaster e seu principal objetivo com a mesma;
- **Recrutamento:** nessa fase os bots buscam por máquinas vulneráveis a fim de infectá-las e torná-las membros da botnet;
- **Interação:** ocorrem duas ações importantes: primeiro, o bot precisa se registrar na botnet (estabelecimento do canal de C&C) e receber atualizações/comandos do botmaster; segundo, é necessária a manutenção desse canal de comunicação e controle, tal que o botmaster mantenha contato com diversos bots;
- **Comercialização:** o desenvolvedor realiza uma espécie de publicidade de sua botnet destacando suas vantagens e capacidades a fim de atrair "clientes"dispostos a comprar ou alugar sua infraestrutura;

- **Execução do ataque:** nesse estágio o botmaster ordena a execução (lançamento) do ataque propriamente dito;
- **Ataque bem-sucedido:** o objetivo final de uma botnet é ter seu ataque bem-sucedido. Nessa fase, o botmaster pode tentar lançar outros ataques a partir dos mesmos bots e/ou novos bots recém recrutados.

Ao longo de todas as fases nesse ciclo-de-vida, os bots utilizam mecanismos complementares de ocultação. A função destes mecanismos consiste na tentativa de ocultar a existência do bot. Para isso, são utilizadas técnicas como: polimorfismo, mascaramento de IP (*IP spoofing*), criptografia, etc. Em (RODRÍGUEZ-GóMEZ; MACIá-FERNáNDEZ; GARCÍA-TEODORO, 2013) há uma tabela que ilustra quais mecanismos são mais utilizados e em quais fases do ciclo de vida estes são utilizados.

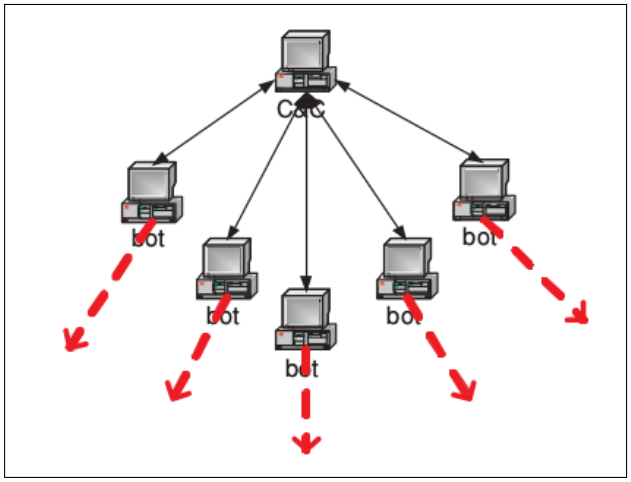
2.1.2 Arquitetura de uma botnet

A força de uma botnet está em possuir uma rede flexível de computadores que podem ser controlados remotamente (KARIM *et al.*, 2014), com isso, a escolha de uma arquitetura de comunicação é uma decisão importante do ponto de vista do botmaster, influenciando em sua capacidade de gerenciamento, resiliência e em alguns aspectos de seu funcionamento (TIIRMAA-KLAAR *et al.*, 2013).

A busca por mecanismos que proporcionem uma infraestrutura flexível e confiável para sua botnet é uma questão importante enfrentada por desenvolvedores de botnets (BILGE *et al.*, 2011). Sendo assim, diferentes arquiteturas foram utilizadas a fim de evitar contramedidas de bloqueio, mantendo a operabilidade da botnet pelo maior período possível (TIIRMAA-KLAAR *et al.*, 2013). Quanto a sua estrutura de C&C, uma botnet pode ser projetada como uma arquitetura centralizada, descentralizada ou híbrida.

Em uma arquitetura centralizada (Figura 3), os bots seguem uma infraestrutura cliente/servidor tradicional, onde um conjunto de bots está sob o controle de um servidor de C&C (KARIM *et al.*, 2014). Esta arquitetura, apesar de ser mais simples de gerenciar, apresenta a desvantagem de possuir um único ponto de falha (TIIRMAA-KLAAR *et al.*, 2013). Uma vez detectados seus servidores de C&C, estes podem ser eliminados/bloqueados da rede causando a derrubada da botnet (pelo menos na rede onde foi identificada), isso pode ser feito adicionando seus endereços a uma *blacklist*, ou realizando as ações necessárias para a recuperação da máquina infectada, por exemplo. Dentre os protocolos mais utilizados estão o IRC e o HTTP.

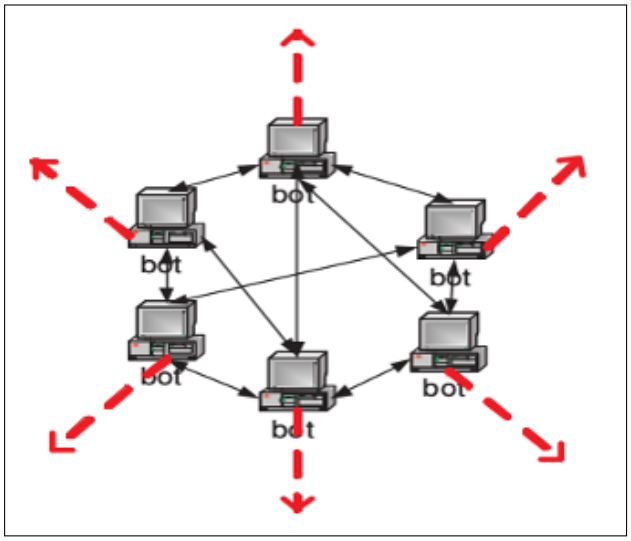
Figura 3 – Botnet com arquitetura centralizada.



Fonte: (GU *et al.*, 2008)

A fim de mitigar essa fraqueza, os desenvolvedores de botnets passaram a utilizar uma arquitetura descentralizada (KARIM *et al.*, 2014), basicamente baseada em P2P (Figura 4). Desta forma, torna-se mais difícil derrubar a botnet uma vez que qualquer bot pode agir como cliente ou servidor de acordo com a vontade do atacante. Ainda que alguns membros sejam identificados como servidores de C&C e estes sejam eliminados, outros membros da mesma botnet podem assumir seu lugar e passar a comandar esse "exército de zumbis". Apesar de sua maior flexibilidade, esta arquitetura necessita de mecanismos de gerenciamento mais complexos (TIIRMAA-KLAAR *et al.*, 2013) e o tempo de disseminação das mensagens é maior que em uma arquitetura centralizada (RODRÍGUEZ-GóMEZ; MACIá-FERNáNDEZ; GARCÍA-TEODORO, 2013).

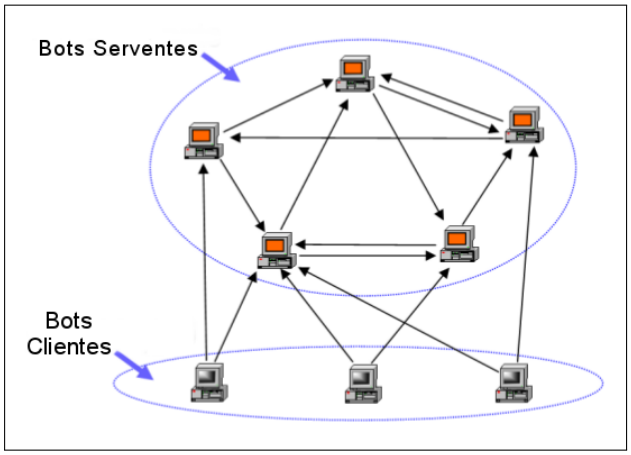
Figura 4 – Botnet com arquitetura descentralizada (P2P).



Fonte: (GU *et al.*, 2008)

Botnets mais complexas utilizam uma arquitetura híbrida (Figura 5) aproveitando-se das vantagens de ambas supracitadas. Bots nessa arquitetura dividem-se em duas categorias: *bots serventes* e *bots clientes* (WANG; SPARKS; ZOU, 2007). Resumidamente, bots serventes agem como clientes e servidores simultaneamente, escutando por conexões de entrada e possuindo endereços IP roteáveis (utilização de endereços estáticos), enquanto que bots clientes não escutam por conexões de entrada e possuem endereços IP dinâmincos.

Figura 5 – Botnet com arquitetura híbrida.



Fonte**:** (WANG; SPARKS; ZOU, 2007) - adaptado.

2.2 BOTNETS E O PROTOCOLO DNS

Devido a sua versatilidade, bots não sabem qual ação executar logo após a infecção, portanto, sua primeira ação é tentar estabelecer um canal de C&C com o botmaster a fim de registra-se na botnet e então passar receber comandos, atualizações e trocar dados (STONE-GROSS *et al.*, 2009; TIIRMAA-KLAAR *et al.*, 2013).

Os primeiros bots carregavam em seu executável a representação estática da localização de seu(s) servidor(es) de C&C (endereço IP, nome de domínio). A utilização dessa técnica tornava relativamente simples a derrubada de uma botnet por parte de equipes encarregadas de tal tarefa, desde que houvesse um exemplar do bot para ser analisado.

Do ponto de vista do botmaster, era necessária alguma técnica tal que fosse possível o estabelecimento da comunicação entre os membros da botnet e que os bots continuassem invisíveis e portáveis (CHOI *et al.*, 2007). Em resposta a essa necessidade, bots passaram a utilizar o protocolo DNS para localizar seus servidores de C&C e estabelecer este canal de comunicação entre estes (STONE-GROSS *et al.*, 2009; BILGE *et al.*, 2011; ANTONAKAKIS *et*

al., 2012).

O protocolo DNS consiste em um sistema distribuído hieráquico reponsável pelo mapeamento de endereços IP em nomes de domínios mais facilmente memorizados por seres humanos (KUROSE; ROSS, 2012). Basicamente, quando um domínio como *www.exemplo.com* é digitado em um navegador, por exemplo, o protocolo DNS é utilizado para resolver qual endereço IP corresponde ao servidor Web (www), neste caso, para o domínio *exemplo.com*. Essa busca ocorre de maneira recursiva entre servidores de níveis diferentes até que algum retorne o endereço solicitado. Quando a solicitação chega a um servidor, este busca pelo mapeamento correspondente em seus registros, ou indica qual outro servidor pode ser utilizado para o domínio em questão. Quando é encontrado, o endereço IP é retornado, caso contrário, é emitida uma resposta dizendo "domínio não existente"(NXDOMAIN). Na tentativa de reduzir o tempo de consulta, o protocolo DNS permite a utilização de caches que armazenam, por um período tempo, os domínios solicitados em um determinada rede.

O protocolo DNS, além de prover um mapeamento entre nomes de domínio e endereços IP, permite que um determinado hospedeiro possua mais de um nome (*host aliasing*) e que um mesmo domínio seja associado a mais de um endereço IP. Com isso, é possível que um determinado serviço seja distribuído entre servidores diferentes e, em caso de falha de algum destes servidores, outro poderá responder as requisições ao serviço.

Portanto, a utilização do protocolo DNS agilizou o processo de migração de botnet, mantendo assim a disponibilidade e a operabilidade da botnet ainda que um ou mais de seus servidores de C&C tenha sido descoberto e banido da rede (CHOI *et al.*, 2007; ANTONAKAKIS *et al.*, 2010).

A seguir são descritas duas técnicas utilizadas por botnets atualmente envolvento o protocolo DNS: Fast-Flux Service Networks e Algoritmos de Geração de Domínios (ANTONAKAKIS *et al.*, 2012).

2.2.1 Fast-Flux Service Networks

(HOLZ *et al.*, 2008) resumem Fast-Flux Service Networks (FFSN) como uma versão maliciosa do Round-Robin DNS (RRDNS) ou de uma Rede de Distribuição de Conteúdo (CDN) contendo algumas alterações.

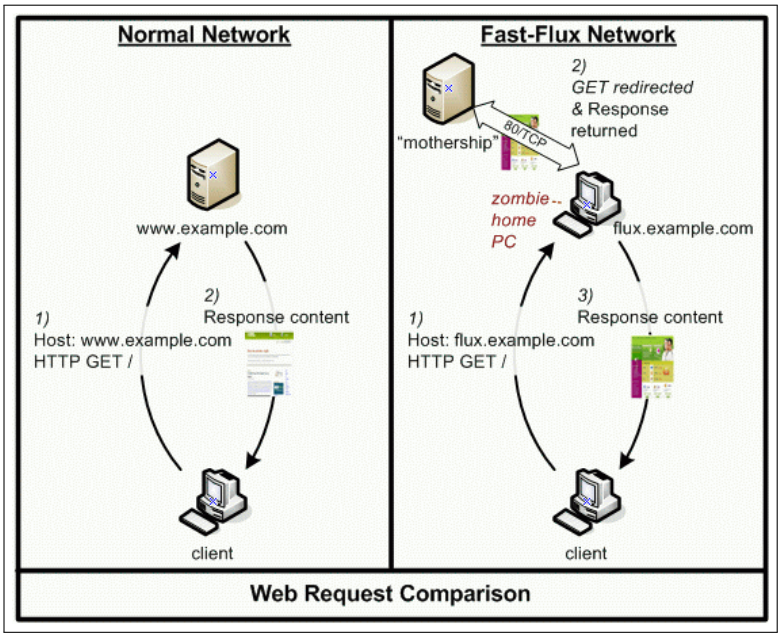
Basicamente, são utilizados valores de *Time-To-Live (TTL)* relativamente pequenos quando comparados aos valores utilizados para domínios não maliciosos. O TTL é utilizado por

servidores DNS para determinar por quanto tempo um determinado domínio deve ser mantido em cache.

Portanto, em uma FFSN, a lista de endereços IP presentes na resposta, os hospedeiros que fazem parte do "fluxo", é atualizada mais rapidamente (SALUSKY; DANFORD, 2007). Essa propriedade garante que um determinado domínio possa ser atribuído a um endereço IP diferente em um intervalo menor, dificultando a derrubada do serviço através de blacklists dos endereços descobertos como maliciosos por exemplo.

Além da propriedade anterior, esses endereços não correspondem aos servidores de C&C. Esses endereços correspondem a hospedeiros infectados, chamados de agentes-de-fluxo, que servem apenas como uma camada de redirecionamento (proxies) para os servidores de C&C, como pode ser observado na Figura 6.

Figura 6 – Comparação entre uma rede normal e uma FFSN.



Fonte**:** (SALUSKY; DANFORD, 2007).

2.2.2 Algoritmos de Geração de Domínios

Bots que utilizam um Algoritmo de Geração de Domínios (*Domain Generation Algorithms - DGA*) geram uma lista de nomes de domínio candidatos a servidor de C&C a cada tentativa de comunicação ou dado um intervalo determinado pelo botmaster durante a fase de concepção do bot. O primeiro host que responder a essas requisições será considerado um servidor de C&C válido.

A principal contribuição desta técnica para desenvolvedores de botnets está no aumento de sua resiliência. Caso algum domínio seja bloqueado, basta registrar um outro domínio que possa ser gerado pelo DGA, retomando o controle da botnet e mantendo sua operabilidade.

Pesquisadores já realizaram a engenharia reversa desses algoritmos a fim de prever os domínio gerados e tomar o controle da botnet. Essa abordagem técnica demonstra-se inviável por motivos como: o tempo necessário e a exigência de conhecimentos avançados em engenharia reversa; a capacidade de mutação dos bots atuais e; ser financeiramente inviável, devido à quantidade de domínios que seria necessária registrar para realizar o bloqueio de uma botnet (STONE-GROSS *et al.*, 2009; ANTONAKAKIS *et al.*, 2012).

Vale observar que não é necessário que todos os domínios gerados estejam válidos simultaneamente para que a botnet opere (STONE-GROSS *et al.*, 2009). Esse comportamento leva ao aumento da quantidade de repespostas negativas do protocolo DNS. (ANTONAKAKIS *et al.*, 2012) mostraram que a presença de bots utilizando DGAs em uma rede pode ser detectada baseando-se no aumento significativo da quantidade de respostas do tipo NXDOMAIN.

2.2.3 Comparação entre as duas técnicas

Primeiramente, enquanto FFSNs podem ser vistas como o mapeamento de um domínio a uma lista de endereços IP, DGAs fazem o oposto, mapeiam uma lista de domínios a um endereço IP. Daí seus nomes alternativos: *IP flux* e *domain flux*, respectivamente (STONE-GROSS *et al.*, 2009).

Observe que ambas possuem um ponto fraco semelhante: em FFSNs, o **domínio** que está sendo utilizado pode ser incluído em uma blacklist, ao passo que, em DGAs, o **endereço IP** retornado pode ser incluído em uma blacklist. Para diminuir este problema, uma botnet pode utilizar as duas técnicas em conjunto: DGAs para o domínio e, em seguida, FFSN para esconder o(s) servidor(es) de C&C e estabelecer sua comunicação.

Ao utilizar as duas técnicas em conjunto, uma botnet atinge um grau de resiliência bem maior uma vez que a lista de domínios gerados pelo DGA e o conjunto de hospedeiros infectados por uma FFSN podem possuir milhares de elementos.

2.3 DETECÇÃO DE BOTNETS

O principal desafio para os pesquisadores nesta área está na detecção de botnets que ainda não são conhecidas (TIIRMAA-KLAAR *et al.*, 2013), ou seja, aquelas que ainda não foram descobertas e modeladas. Sendo assim, as técnicas para detecção de botnets buscam ficar a par das carateríticas das botnets atuais através de diversas técnicas como análise de arquivos de *log*, monitoramento de hospedeiros, monitoramento de redes e instalação de armadilhas para os bots (*honeypots* e *spam-traps*).

Quando capturado, um bot pode ser analisado estática e dinamicamente a fim de extrair suas características comportamentais a nível de hospedeiro e/ou de rede (SIKORSKI; HONIG, 2012; TIIRMAA-KLAAR *et al.*, 2013). Porém, nem sempre é viável fazer a engenharia reversa de um exemplar de um bot, uma vez que eles podem ser programados para se modificarem a cada infecção ou ainda, sofrerem mutações como resposta a comandos do botmaster (STONE-GROSS *et al.*, 2009; ANTONAKAKIS *et al.*, 2012).

Recentemente, o processo de detecção de botnets concentrou-se na análise de comportamento dos bots nas redes (CAVALLARO; KRUEGEL; VIGNA, 2009; GU *et al.*, 2008; BILGE *et al.*, 2012). Em uma rede infectada, podem ser observados comportamentos baseados em: protocolo utilizado, quantidade de mensagens trocadas em um determinado período, variações geradas no tráfego da rede, acessos à sites marcados como maliciosos, entre outros.

2.3.1 Abordagens de detecção

(GU *et al.*, 2008) destacaram que independente da arquitetura utilizada, botnets compartilham uma característica: os bots pertencentes a uma mesma botnet, tendem a executar um mesmo comportamento, pré-determinado em seu código por seus criadores.

Na busca em capturar esses comportamentos, são seguidas duas abordagens (GU *et al.*, 2008; CAVALLARO; KRUEGEL; VIGNA, 2009):

- **Correlação vertical:** inspeciona a rede em busca de evidências da infecção de bots ou comunicação entre as máquinas e servidores de C&C. Algumas dessas técnicas estão diretamente relacionadas com a estrutura de uma determinada botnet ou necessitam que a rede já possua o ciclo de vida de uma determinada botnet. Além disso, essas técnicas precisam que ruídos na rede disparem alarmes para iniciar o processo de detecção;
- **Correlação horizontal:** busca correlacionar eventos de rede a fim de identificar hospedeiros envolvidos em atividades maliciosas similares. Apesar dessas técnicas serem

independentes de uma botnet ou outra, ainda dependem que hajam bots infectados na rede monitorada.

Ainda em (GU *et al.*, 2008), os autores capturam duas visões diferentes do comportamento dos bots, dividindo-o em dois planos: um plano de comunicação e um plano de ações ou atividades nas quais os bots estão envolvidos.

O processo de análise de comportamento pode ocorrer também seguindo uma abordagem passiva ou ativa (STONE-GROSS *et al.*, 2009). Em uma abordagem passiva, são estudados os efeitos secundários gerados pela atividade das máquinas infectadas. Já em uma abordagem ativa, os pesquisadores de fato buscam infiltrar-se na botnet através da utilização de um exemplar do malware ou um cliente simulando um bot.

Uma vez detectada a presença de uma ou mais botnets em uma rede, é feita a modelagem do comportamento dos bots identificados e em seguida esse modelo é aplicado à rede a fim de mitigar a ameaça. Para automatizar o processo de modelagem desses comportamentos, têm-se utilizado técnicas de aprendizagem de máquina (GU *et al.*, 2008; RIBEIRO; FILHO; MAIA, 2011; BILGE *et al.*, 2012; ZHAO *et al.*, 2013).

Para o passo seguinte, a modelagem do comportamento, é necessária a identificação de características capazes de representar o comportamento em questão. Outra decisão importante está ligada ao objetivo da detecção: detectar a presença de uma ou mais infecções na rede ou ir um pouco além, e identificar quem está infectado. As seções seguintes neste capítulo trazem uma breve discussão desses aspectos respectivamente.

2.3.2 Caraterização do comportamento

O comportamento anômalo de um bot pode ser explorado de diversas maneiras para a detecção. As características utilizadas podem ser classificadas como:

- Características do fluxo de rede;
- Características de payload;
- Características temporais.

A seguir é feita uma breve discussão sobre as principais características encontradas nos trabalhos acadêmicos. Deve-se levar em consideração que as características de cada classe podem ser combinadas de acordo com a intenção do pesquisador.

2.3.2.1 Características do fluxo de rede.

Essas características possibilitam uma identificação global dos padrões de comunicação dos bots quanto a: quais hospedeiros estão se comunicando e o protocolo utilizado por eles. Dentre essas características encontram-se:

- Endereço IP e número de porta na origem;
- Endereço IP e número de porta no destino;
- Protocolo utilizado;
- Número de pacotes trocados no fluxo em um dado período (segundos, minutos, horas);
- Tamanho dos pacotes;
- Quantidade de bytes por conexão;
- Número de re-estabelecimento de conexões;
- Heterogeneidade dos hospedeiros infectados.

Exemplos da utilização destas características podem ser encontrados em (PASSERINI *et al.*, 2008), (BILGE *et al.*, 2012) e (ZHAO *et al.*, 2013). Uma lista de características mais específicas pode ser encontrada em (GUNTUKU; NARANG; HOTA, 2013).

2.3.2.2 Características de payload.

Esse conjunto de características está diretamente ligado ao protocolo utilizado pelos bots, permitindo um melhor entendimento do comportamento dos bots. Exemplos destas características são:

- Tipos de mensagens trocadas;
- Assinaturas de bytes em comum;
- Assinaturas conhecidas como maliciosas;
- Estrutura das mensagens trocadas;
- Ordem das mensagens trocadas.

Uma botnet pode fazer o uso de protocolos diferentes em fases diferentes de seu ciclo de vida e até dependendo de sua arquitetura. Cabe aos pesquisadores a decisão de qual protocolo será modelado. Com isso, as metodologias devem levar em consideração a possibilidade de incorporar novos comportamentos, utilizado plugins por exemplo, ou ficarem limitadas ao protocolo selecionado para modelagem.

Exemplos da utilização destas características podem ser encontrados em (LEE *et al.*, 2008), (CAI; ZOU, 2012), (ANTONAKAKIS *et al.*, 2012) e (BARTHAKUR; DAHAL; GHOSE,

2012).

2.3.2.3 Características temporais.

Essas características são utilizadas a fim de identificar padrões temporais na comunicação entre os membros da botnet. Dentre as mais comuns estão:

- Intervalo entre o primeiro e o último pacote em uma conexão;
- Intervalo entre as conexões;
- Início das comunicações;
- Valores de TTL de pacotes.

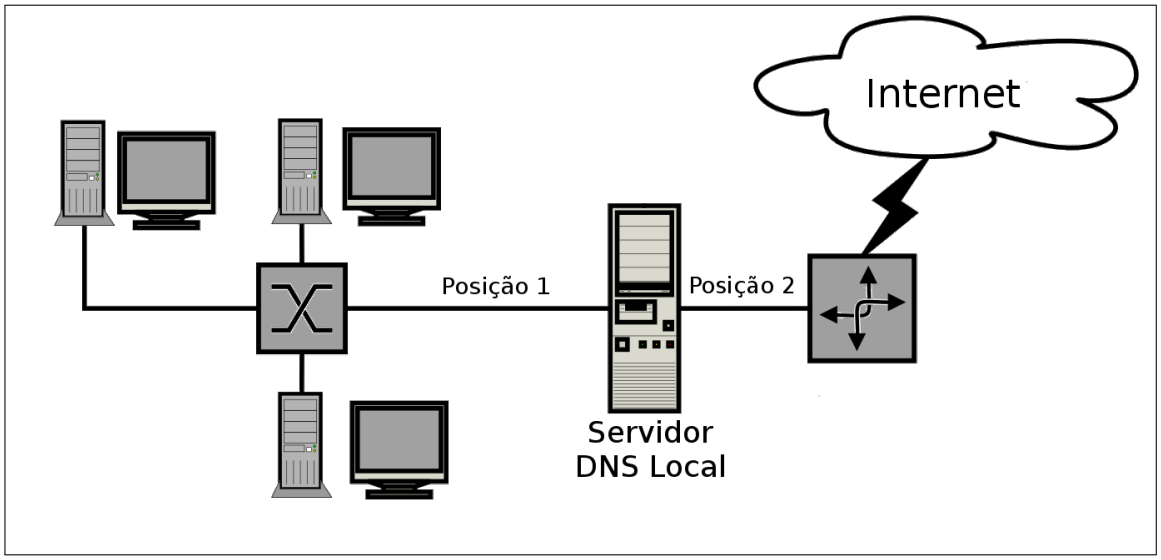
Exemplos da utilização destas características podem ser encontrados em (PASSERINI *et al.*, 2008), (BILGE *et al.*, 2011) e (ZHAO *et al.*, 2013).

2.3.3 Detecção de uma infecção *vs.* Identificação de hospedeiros infectados

Do ponto de vista de um administrador de rede, é importante detectar infecções em seu domínio e, principalmente, identificar quais máquinas estão infectadas, uma vez que torna mais fácil a tarefa de sanar a infecção naquele ambiente.

A capacidade de detectar quais máquinas estão infectadas em uma rede está relacionada ao posicionamento de sensores em relação a rede a ser monitorada e o restante do mundo. A Figura 7 auxilia na compreensão desse fenômeno em relação ao protocolo DNS.

Figura 7 – Exemplo de cenário de uma rede convencional.



Fonte**:** Elaborado pelo autor.

Um sensor localizado na Posição 1 tem a capacidade de identificar quais máquinas tentaram resolver os nomes de domínios maliciosos, pois as requisições possuem os endereços das máquinas que fizeram a requisição. Por outro lado, lembrando que o servidor DNS local age como um proxy para aquela rede quanto ao protocolo DNS, para um sensor situado na Posição 2, todas as requisições percebidas possuem o mesmo endereço de origem, o endereço do servidor DNS local.

3 TRABALHOS RELACIONADOS

Nesta seção, serão brevemente apresentados os trabalhos relacionados, destacando sua utilização dos elementos apresentados na fundamentação teórica deste trabalho.

3.1 BOTMINER: CLUSTERING ANALYSIS OF NETWORK TRAFFIC FOR PROTOCOL- AND STRUCTURE-INDEPENDENT BOTNET DETECTION

Em (GU *et al.*, 2008), os autores propõem uma *framework* de detecção generalizada baseada nas propriedades essenciais de botnets. O objetivo principal do BotMiner é detectar, em uma rede monitorada, grupos de máquinas comprometidas por uma botnet.

Através de uma análise baseada em duas visões diferentes do tráfego da rede, uma focada em informações de controle e a outra nas ações realizadas na rede, a proposta dos autores identifica *quem está falando com quem* e *quem está fazendo o que* na rede monitorada. BotMiner agrupa máquinas que apresentam padrões similares de comunicação e comportamento considerados suspeitos e, em seguida, faz uma combinação destes dois modelos a fim de tomar sua decisão se aquele conjunto de máquinas faz parte de uma botnet.

3.2 MINING THE NETWORK BEHAVIOR OF BOTS

(CAVALLARO; KRUEGEL; VIGNA, 2009) trazem um sistema que monitora a atividade de um bot em um ambiente controlado e extrai modelos que capturam as características de seu comportamento na rede. Os autores ressaltam que: (I) a proposta não depende de conhecimento prévio sobre a estrutura da botnet; (II) por não realizar sua inspeção nos *payloads* dos pacotes, é resiliente quanto ao uso de técnicas que encriptam a comunicação entre os bots; (III) por não necessitar da presença de ruídos na rede gerados pelos bots, é capaz de detectar infecções desde o primeiro hospedeiro infectado.

O sistema proposto agrupa fluxos semelhantes de acordo com a quantidade de bytes trocados, número de pacotes por fluxo e endereços IP de origem e destino. Em seguida, é realizada uma análise *intra-cluster* a fim de detectar comportamentos periódicos e quais os períodos em que estes ocorrem. Opcionalmente podem ser realizadas outras duas análises: uma análise de dependência *inter-clusters* que busca identificar rastros menos ruidosos da presença de um bot e comportamentos aperiódicos; e, uma correlação de *clusters inter-trace* para identificar similaridades entre clusters obtidos em diversas execuções do bot.

3.3 EXPOSURE: FINDING MALICIOUS DOMAINS USING PASSIVE DNS ANALYSIS

(BILGE *et al.*, 2011) destacam a importância do protocolo DNS para serviços não-maliciosos e maliciosos. Eles ainda afirmam que detectar domínios maliciosos, tão logo eles apareçam na rede, implica em uma significante medida para mitigar operações maliciosas. Sua principal hipótese é a de que, ao analisar grandes quantidades de dados do tráfego de uma rede, requisições DNS maliciosas e não-maliciosas exibirão padrões de comportamento diferentes e que poderão ser detectadas automaticamente.

Com base nessa hipótese, os autores apresentam EXPOSURE uma ferramenta que utiliza um conjunto de características genéricas para a definição do grau de maliciosidade de um dado domínio requisitado na rede monitorada a fim de reportar se este foi, ou não, utilizado em uma atividade maliciosa.

3.4 FROM THROW-AWAY TRAFFIC TO BOTS: DETECTING THE RISE OF DGA-BASED MALWARE

Em (ANTONAKAKIS *et al.*, 2012), os autores apresentam *Pleiades*: um sistema capaz de detectar bots que utilizam Algoritmos Geradores de Domínios (*Domain Generation Algorithm - DGA*) dentro de uma rede monitorada sem utilizar técnicas de engenharia reversa.

Pleiades parte da idéia que bots que utilizam DGAs geram um aumento na quantidade de respostas DNS negativas na rede. Com base nessa hipótese, são analisados os domínios resultantes das requisições mal sucedidas. Destes são extraídas carcterísticas utilizadas para a geração de modelos para sua detecção. A proposta também é capaz de identificar servidores de C&C das botnets detectadas.

3.5 BOTNET DETECTION USING PASSIVE DNS

Em (LUZ, 2014), o autor propõe a utilização de dados em (WEIMER, 2005) para a detecção de nomes de domínio relacionados à atividades maliciosas. Após a extração das características, com o auxílio de *whitelists* e *blacklists*, foi treinado um classificador capaz de distinguir domínios legítimos de domínios associados à atividades maliciosas.

Foi avaliado o impacto na performance da classificação utilizando as características léxicas dos domínios observados e as características relacionadas à rede. Essa avaliação mostrou que as características relacionadas à rede possuem uma performance menor que as características

léxicas para a classificação baseada no tráfego DNS.

3.6 PHOENIX: DGA-BASED BOTNET TRACKING AND INTELLIGENCE

Partindo da hipótese que botnets atuais apoiam-se na utilizaçao de DGAs, os autores apresentam um mecanismo para detecção de botnets baseada em seus DGAs. Em (SCHIAVONI *et al.*, 2014), Phoenix é este mecanismo capaz de, não apenas, separar domínios gerados por DGAs daqueles que não foram, mas também é capaz de encontrar grupos de domínios gerados por DGAs e que são representativos de suas respectivas botnets.

Para a caracterização dos domínios combinados atributos relacionados às cadeias de caracteres observadas e atributos relacionados ao endereço IP. Essas características são utilizadas para agrupar domínios de comportamento semelhante quanto aos endereços IP resolvidos. Com isso, novos domínios, ainda não observados, podem ser adicionados a um desses grupos que mais se assemelha, obtendo assim uma base da evolução da botnet monitorada.

3.7 DETECTING APT MALWARE INFECTIONS BASED ON MALICIOUS DNS AND TRAFFIC ANALYSIS

(ZHAO *et al.*, 2015) apresentam o IDnS: um sistema que utiliza técnicas de análise de domínios maliciosos para detecção de domínios ligados a servidores de comando e controle e, em seguida, analisa o tráfego do endereço IP suspeito utilizando tecnologias de detecção baseadas em assinaturas e anomalias.

Com base em características passivas e ativas, o sistema também possui um módulo responsável por atribuir um valor de reputação para os endereços IP a fim de julgar se o hospedeiro correspondente está infectado ou não.

3.8 BOTMETER: CHARTING DGA-BOTNET LANDSCAPES IN LARGE NETWORKS

(WANG *et al.*, 2016) apresenta Botmeter: uma ferramenta capaz de traçar um panorama da população de bots que utilizam DGAs em uma rede de grandes proporções. Os autores estabeleceram uma nova taxonomia baseada nas características comportamentais do fluxo de domínios. Essa taxonomia está adequada ao tráfego DNS agregado encontrado em servidores de níveis mais elevados na hierarquia so protocolo DNS.

A ferramenta também é capaz de diferenciar as famílias de DGAs de acordo com: a

maneira que as listas de candidatos a servidor de conmando e controle são geradas; a maneira que um domínio é selecionado para ser o servidos de C&C; e a configuração dos conjuntos de domínios requisitados pelo bot e aqueles que foram selecionados pelo botmaster.

3.9 DETECTING DGA MALWARE TRAFFIC THROUGH BEHAVIORAL MODELS

Em (ERQUIAGA; CATANIA; GARCíA, 2016), os autores analisam abordagens de detecção comportamental baseada em Cadeias de Markov para diferenciar o trágefo de DGAs e DNS legítimo.

Inicialmente, é feito um agrupamento baseado em uma tupla com informações de origem e destino da comunicação entre hospedeiros, esse agrupamento é chamado de conexão. Em seguida são extraídos o tamanho, duração e periodicidade de cada fluxo. Para cada fluxo é atribuído um estado, definido pelos autores, e todos os estados presentes em uma conexão é tomado como um modelo de comportamento. Para detecção, cada conexão terá seu modelo de Cadeia de Markov. Quando há um novo tráfego, este passa pelo crocesso de agrupamento e é testado em cada Cadeia de Markov criada a fim de identificar a qual conexão ele pertence.

4 PROPOSTA

A principal hipótese deste trabalho é que bots que utilizam DGAs geram um aumento na quantidade de respostas negativas do protocolo DNS, respostas do tipo NXDOMAIN. Com isso, é possível utilizar este tráfego para identificação de domínios maliciosos e, a partir daí, quais máquinas estão infectadas em uma rede monitorada (ANTONAKAKIS *et al.*, 2012).

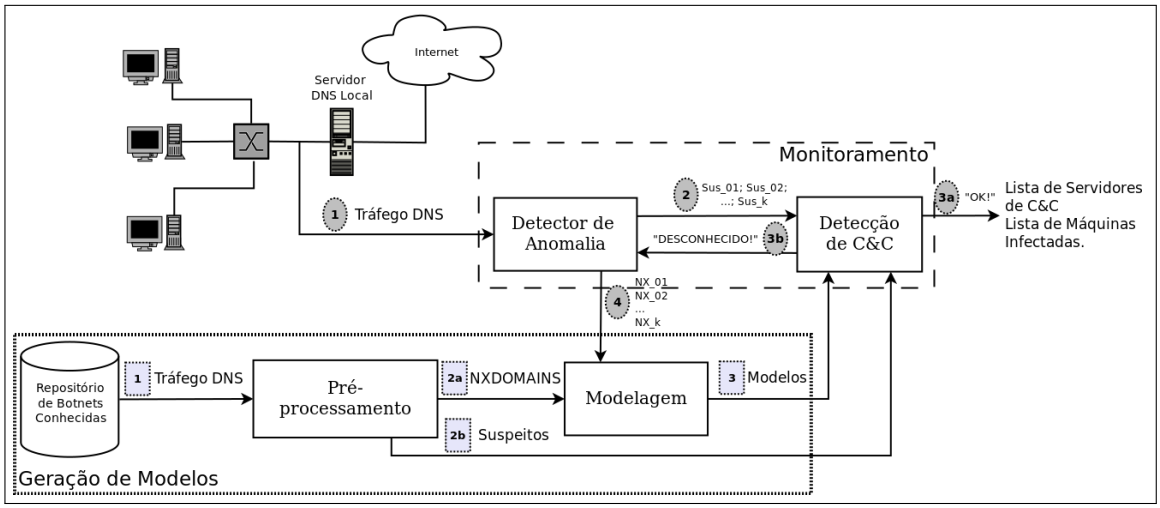
Com base nessa hipótese, a metodologia a seguir realiza uma correlação vertical baseada na análise passiva do comportamento de bots quanto ao protocolo DNS, em especial daqueles que utilizam Algoritmos de Geração de Domínios (DGAs) (subseção 2.2.2). Essa análise é realizada sobre o tráfego DNS que seria descartado, ou seja, concentra-se nos pacotes resposta DNS do tipo NXDOMAIN.

A metodologia visa, além de identificar a presença de uma infecção em uma rede, indicar quais botnets estão presentes naquele ambiente com base em um conhecimento de botnets previamente observadas.

4.1 VISÃO GERAL DA METODOLOGIA

A Figura 8 apresenta uma visão geral da metodologia.

Figura 8 – Visão geral e fluxo de informações da metodologia.



Fonte**:** Elaborado pelo autor.

A arquitetura está dividida em duas camadas: Monitoramento – composto por um módulo Detector de Anomalia e um módulo de Detecção de C&C; e Geração de Modelos, composto por uma base de dados composta por capturas de tráfego malicioso previamente

observadas, um módulo de Pré-processamento e um módulo de Modelagem.

A camada de Monitoramento é executada na dentro da rede monitorada, dessa forma é possível identificar não somente a existência de infecções, mas também, quais máquinas foram contaminadas a fim de que estas sejam sanadas. A camada de Geração de Modelos pode ser executada em um ambiente diferente do qual a camada anterior se encontra. Recomenda-se que este outro ambiente possua um poder computacional maior em relação ao Monitoramento tal que a geração dos modelos possa ocorrer o mais breve. O fluxo básico de funcionamento ocorre como descrito a seguir.

Primeiramente, o tráfego DNS é monitorado pelo Detector de Anomalia (Passo 1), este observa a rede monitorada a fim de identificar uma elevação nas taxas de tráfego NXDOMAIN, ao percerber tal comportamento anômalo, os domínios pertencentes a esse tráfego são agrupados e cada agrupamento é devidamente rotulado. As respostas de cada agrupamento são enviadas para a Detecção de C&C (Passo 2). Ao chegarem no módulo de Detecção de C&C, as respostas recebidas podem levar a um dos resultados a seguir:

- Ameaça detectada ("OK!") – retorna os endereços IP dos servidores de C&C e a lista de máquinas infectadas (Passo 3a);
- Ameaça desconhecida ("DESCONHECIDO") – padrão de comportamento ainda não modelado. O módulo envia um sinal para o módulo anterior (Passo 3b).

Seguindo o fluxo em caso de uma ameça ainda desconhecida, as respostas NXDOMAIN do agrupamento desconhecido são enviados para a camada de Geração de Modelos para que esta possa gerar um modelo para o novo comportamento (Passo 4).

Na camada de Geração de Modelos, o tráfego DNS obtido de um repositório de Botnets Conhecidas é enviado para uma fase de Pré-Processamento (Passo 1), onde será separado em respostas NXDOMAIN e respostas Suspeitas. O tráfego NXDOMAINS é enviado para a Modelagem (Passo 2a) e o tráfego Suspeito é enviado para a extração dos endereços IP dos servidores reportados (Passo 2b).

Quando a camada de Geração de Modelos recebe uma nova ameaça (Passo 4 da camada anterior), as requisições NXDOMAIN são enviadas direto para o módulo de Modelagem. Neste Módulo, são extraídas a características dos domínios requisitados e submetidos à técnicas de aprendizado de máquina para a geração dos modelos de detecção que serão enviados ao módulo de Detecção de C&C da camada de Monitoramento (Passo 3 da camada de Geração de Modelos).

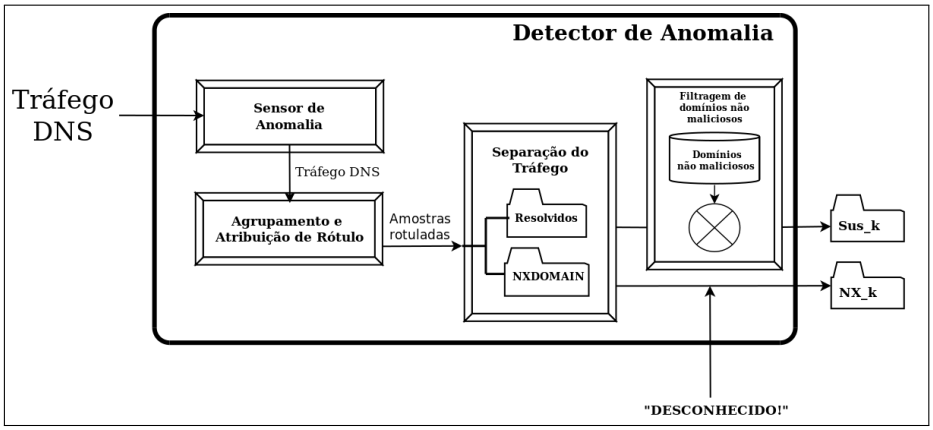
A seguir são dadas mais informações sobre os componentes da aquitetura proposta.

4.1.1 Detector de Anomalia

O módulo Detector de Anomalia (Figura 9) possui um sensor responsável por identificar alterações no tráfego NXDOMAIN da rede monitorada. Ao ser identificado uma alteração no padrão de comportamento na rede, o tráfego de máquinas com comportamento semelhante é agrupado e cada agrupamento recebe um rótulo para sua identificação. Em seguida, para cada agrupamento, são separadas as respostas positivas de servidores DNS (Resolvidos) e as respostas NXDOMAIN.

Com o auxílio de *whitelists*, os domínios do conjunto Resolvidos passam por um filtro para retirar domínios conhecidos como não maliciosos restando apenas um conjunto de domínios resolvidos suspeitos de pertencer a atividades maliciosas. Esse conjunto remanescente (Suspeitos) é repassado para o módulo de detecção. Caso receba um sinal de ameaça desconhecida para o conjunto repassado, o Detector de Anomalia envia o conjunto NXDOMAINS para o módulo de Modelagem para a atualização da base de botnets conhecidas e a geração do modelo de detecção para a ameaça recém detectada.

Figura 9 – Módulo Detector de Anomalia



Fonte**:** Elaborado pelo autor.

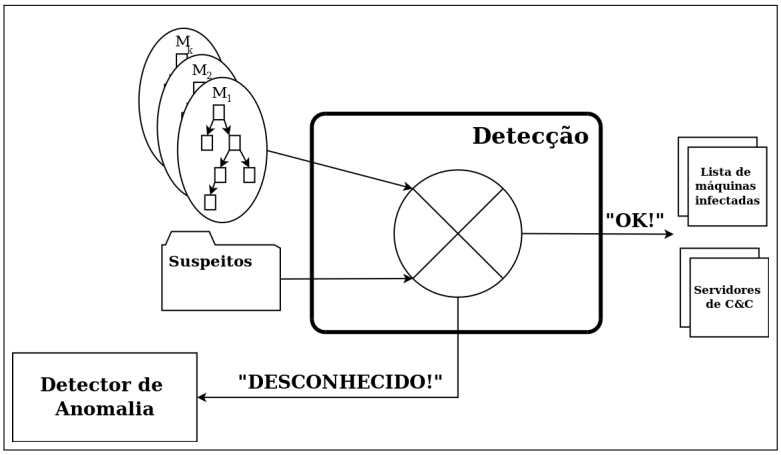
4.1.2 Detecção de servidores de C&C

Como visto na Figura 10, o módulo Detecção recebe os modelos gerados e o conjunto de domínios suspeitos que foram resolvidos por algum servidor DNS. Os elementos do conjunto Suspeitos serão submetidos aos modelos a fim de verificar se foram gerados por algum bot já conhecido.

Quando o resultado desta inspeção for afirmativo, os endereços IP presentes na res-

posta podem ser encaminhados para uma lista de servidores de C&C, e as máquinas que geraram tais requisições podem ser incluídas em uma lista de máquinas infectadas. Caso contrário, um sinal é enviado ao Detector de Anomalia informando que fora detectada um comportamento ainda desconhecido. Como expresso anteriormente, esse sinal dispara a atualização da base de comportamentos observados, bem como os modelos de detecção utilizados por este módulo.

Figura 10 – Estrutura interna do módulo Detecção.



Fonte**:** Elaborado pelo autor.

4.1.3 Base de Botnets Conhecidas

Essa base de dados consiste em capturas de redes infectadas e armazenadas em arquivos no formato *Packet Capture Library* (PCAP). Esse tipo de tráfego pode ser obtido através da execução dos bots em um ambiente controlado, em uma *honeynet*, por exemplo, ou através de bases de dados como a do projeto (STRATOSPHERE, 2015).

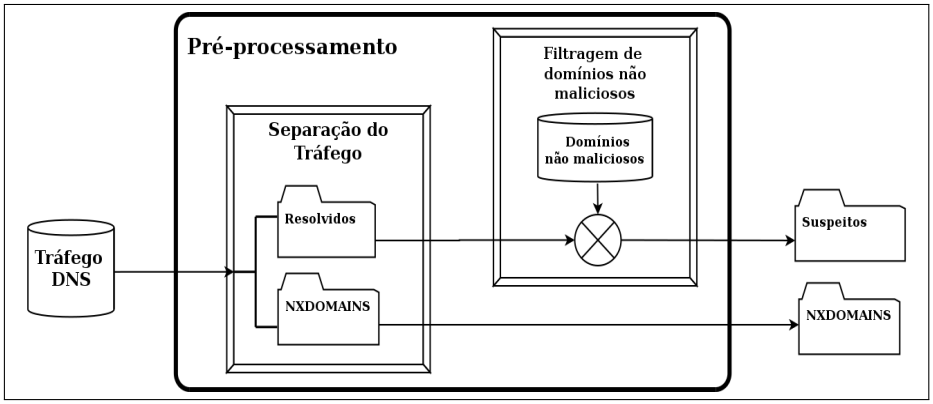
Cada captura deve possuir o tráfego de um bot apenas, garantindo assim, que cada captura seja rotulada pelo bot que gerou aquele tráfego.

4.1.4 Pré-processamento

A Figura 11 mostra uma visão geral do módulo Pré-processamento. Este módulo prepara os dados para a geração dos modelos através das seguintes ações:

- **Separação do tráfego:** O tráfego DNS é dividido em dois conjuntos de dados:
  - *Resolvidos*: tráfego DNS que obteve uma resposta positiva, ou seja, os nomes de domínios que foram devidamente resolvidos por algum servidor DNS;
  - *NXDOMAINS*: tráfego DNS que não foi resolvido, ou seja, as respostas do tipo

Figura 11 – Estrutura interna do módulo Pré-processamento.



Fonte**:** Elaborado pelo autor.

NXDOMAIN.

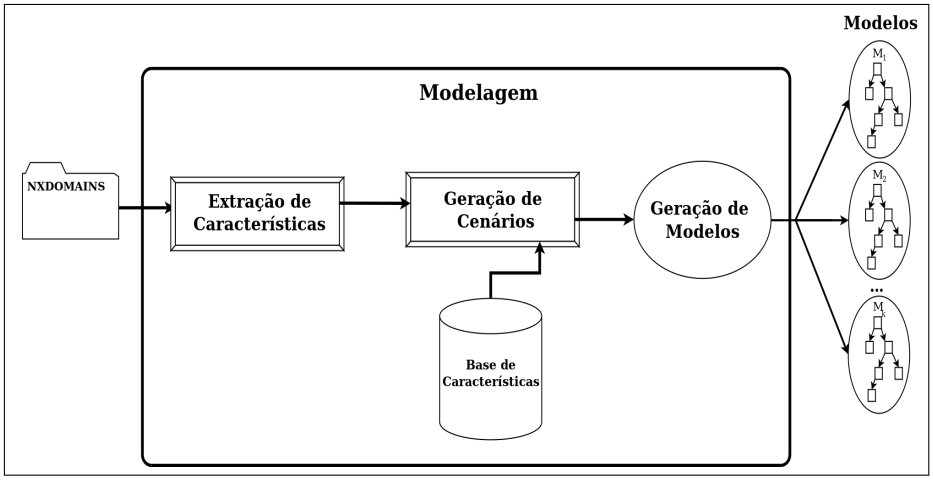
- **Filtragem de domínios não-maliciosos:** Cruza as informações do conjunto Resolvidos com uma lista de domínios conhecidos como não maliciosos a fim de retirá-los do conjunto, deixando somente os domínios que passarão a ser tratados como *Suspeitos*.

Portanto, a saída deste módulo são: o conjunto de domínios *Suspeitos* e o conjunto *NXDOMAINS*. Os Suspeitos são enviados, posteriormente ao módulo de Detecção de C&C para a extração dos endereços dos servidores de C&C e os NXDOMAINS serão enviados para a criação dos modelos de detecção.

4.1.5 Modelagem

A Figura 12 mostra uma visão geral do módulo Modelagem.

Figura 12 – Estrutura interna do módulo Modelagem.



Fonte**:** Elaborado pelo autor.

4.1.5.1 Extração de características

A primeira ação executada neste módulo é a extração de características do tráfego NXDOMAIN recebido.

Os domínios gerados por DGAs não têm a necessidade de ser algo compreensível por seres humanos, tratam-se, basicamente, de cadeias pseudo-aleatórias de caracteres alfa-numéricos como *aeaybikhawkftknjrtxd.org* e *g7ci7ek5ek1di.ad*, por exemplo.

Com base na estrutura dos domínios gerados e nos trabalhos citados no Capítulo 3, os atributos a seguir foram considerados fundamentais para capturar as características básicas dos domínios gerados por um DGA:

- Quantidade de níveis de domínio;
- Tamanho do 2LD;
- Número de caracteres distintos no 2LD;
- Número de dígitos no 2LD;
- Entropia do 2LD;
- Tamanho do 3LD;
- Número de caracteres distintos no 3LD;
- Número de dígitos no 3LD;
- Entropia do 3LD.

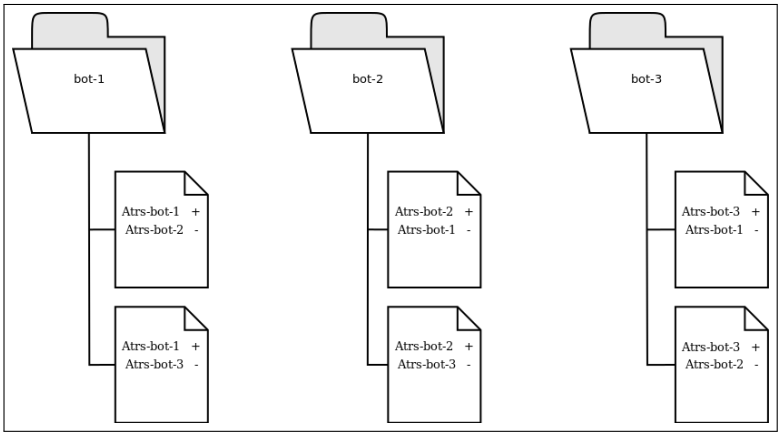
Entropia é um conceito apresentado por(SHANNON; WEAVER, 1949) que pode ser utilizado para representar o grau de aleatoriedade de uma informação. Quanto maior a entropia, maior a aleatoriedade da informação.

4.1.5.2 Geração de Cenários

Nesta fase, são criados cenários contendo informações sobre o bot em análise e os bots já conhecidos. Basicamente, são realizados cruzamentos entre as características da amostra em análise e as demais pertencentes a Base de Características, onde cada cruzamento possui amostras do bot em análise marcadas como positivas (para aquele cenário) e as demais marcadas como negativas. Um exemplo destes cenários, com três amostras na Base de Características, pode ser observado na Figura 13.

Os cenários são, então, repassados à fase de Geração de Modelos.

Figura 13 – Cenários.



Fonte**:** Elaborado pelo autor.

4.1.5.3 Geração dos Modelos

A ideia por trás dos cruzamentos nos cenários é transformar um problema de classificação multiclasse em vários problemas de classificação binária. Ou seja, um problema de classificação multiclasse, com M classes, é transformado em M problemas de classificação binária.

Com base na ideia acima, cada cenário é repassado para um algoritmo de aprendizagem de máquina, resultando em uma série de modelos de classificação binária para cada um dos bots apresentados.

Esta metodologia não se restringe a um algoritmo em particular. Portanto, diversos algoritmos podem ser utilizados para a resolução dos problemas de classificação baseados nos cenários descritos nesta proposta.

5 IMPLEMENTAÇÃO

5.1 CONJUNTO DE DADOS

A base de botnets conhecidas foi montada a partir de dados disponibilizados pelo projeto *Malware Capture Facility Project* (STRATOSPHERE, 2015). Este projeto possui um repositório responsável pela obtenção e manutenção de uma base de dados contendo o tráfego malicioso e não malicioso.

A Tabela 1 resume o quantitativo de respostas DNS encontradas nas amostras utilizadas aqui. As colunas prefixadas com NX denotam os números relacionados às respostas NXDOMAIN, e as colunas com prefixo RES denotam o quantitativo de respostas respondidas com sucesso por algum servidor DNS. Os sufixos UNIQ e TOT correspondem aos quantitativos de nomes de domínios únicos e totais, respectivamente. Informações detalhadas do tráfego das amostras pode ser obtido no site do projeto.

Tabela 1 – Resumo quantitativo de tráfego DNS nas amostras.

Amostra	NX_UNIQ	NX_TOT	RES_UNIQ	RES_TOT
b1	1995	185419	385	4577
b2	2934	352756	32	3418
b3	925	33698	1679	1037072
b4	1956	63775	5188	273635
b5	3568	194087	40	1294
b6	4324	211780	1475	348497
b7	1412	11448	852	1589507
b8	1503	44909	1333	1638905
b9	127	762693	46	165566

Fonte**:** Elaborado pelo autor.

As amostras coletadas pertencem a 4 famílias de botnets. Por família entende-se as variações existentes para um mesmo bot. Destas famílias, representadas entre parênteses: 3 utilizam DGAs – (b1-b5-b6), (b2-b3-b4) e (b7-b8); e 1 utiliza DGA e FFSN – (b9).

Pode-se observar ainda, na maioria das amostras, a quantidade de tráfego relacionado a respostas NXDOMAIN excede consideravelmente o tráfego legítimo quanto ao protocolo DNS.

5.2 CRIAÇÃO DE CENÁRIOS E GERAÇÃO DE MODELOS

Para cada amostra, foram extraídas as características básicas apresentadas na subseção 4.1.5.1. Em seguida foram criados os cenários descritos na subseção 4.1.5.2. A extração das características foi feita através de um script em linguagem Python 2.7 e criação dos cenários em um script em linguagem Bash.

Durante os experimentos, optou-se por não utilizar nenhuma técnica de tratamento de ruídos e também não foram realizadas análises baseadas em agrupamentos de domínios. Dessa forma, os dados foram trabalhados individualmente, da maneira que foram recebidos das capturas de rede.

Os cenários criados foram utilizados para a alimentação da ferramenta Weka (HALL *et al.*, 2009) para o a geração dos classificadores. Lembrando que o problema de classificação com M classes, foi transformado em M problemas de classificação binária, cada cenário resultará em um classificador binário para o seu bot dentre os demais.

Devido a sua alta versatilidade em problemas de classificação, os algortimos selecionados para os testes foram: árvores de decisão (QUINLAN, 1993) e naive-Bayes (JOHN; LANGLEY, 1995).

5.2.1 Naive-Bayes

Naive-Bayes é um método supervisionado de modelagem estatística baseado no teorema de Bayes. Dada uma classe A e uma amostra x:

$$P(A|x) = \frac{P(A)P(x|A)}{P(x)}$$

O termo *naive*, "ingênuo"em uma tradução livre, é dado ao método porque este ainda se apoia na hipótese de que os atributos x_j de uma amostra x são condicionalmente independentes, então a probabilidade de x pertencer a classe A é dada por:

$$P(x|A) = \prod_j P(x_j|C)$$

Apesar da visão simplista de que todos os atributos possam ser independentes em um cenário real, o Naive-Bayes mostrou-se eficiente quando testado sobre conjuntos de dados reais (WITTEN; FRANK; HALL, 2011).

Nos experimentos foi utilizada a implementação do classificador resumido acima.

5.2.2 Árvore de decisão

Árvore de decisão é um método de aprendizagem supervisionada considerado um dos mais simples e ainda mais bem sucedidos dentre os métodos indutivos (RUSSELL; NORVIG, 2010).

Uma árvore de decisão recebe como entrada um objeto ou situação descrito através de um conjunto de atributos e retorna uma decisão. Esta decisão é alcançada após uma sequência de testes realizados em cada nó interno da árvore. Cada nó interno corresponde ao teste do valor de um atributo e seus galhos são rotulados com os possíveis valores desse teste. Cada folha especifica o valor a ser retornado caso aquela folha seja alcançada.

A cada nó visitado durante o percurso na árvore, o conjunto de entrada é dividido de acordo com o valor que está sendo testado. Dessa forma, árvores de decisão podem ser vistas, também, como um particionamento recursivo do espaço de entrada tal que é definido um modelo local em cada região deste espaço (MURPHY, 2012).

Para a realização dos experimentos, foi utilizada a implementação J4.8, a qual corresponde ao algoritmo C4.5 (QUINLAN, 1993). Esse algoritmo é uma extensão ao algoritmo ID3, com a qual é possível: a utilização de atributos numéricos, contínuos e discretos; tratamento para valores de atributos ausentes nas amostras; e a realização de podas sobre a árvore após sua construção.

5.3 EXPERIMENTOS

Os experimentos com a ferramenta Weka foram realizados utilizando a técnica de validação cruzada (*cross-validation*) para a geração dos modelos.

Nessa técnica, o conjunto de amostras é dividido em k subconjuntos, dos quais k - 1 serão utilizados para treinar o algoritmo e o subconjunto remanescente para a validação. A cada repetição esses subconjuntos sofrem um rodízio no qual aquele que foi utilizado para validação será utilizado para treinamento na próxima iteração. Nos testes realizados nesse trabalho foi utilizado um valor $k = 10$, este valor já fora utilizado em trabalhos semelhantes, como em (BILGE *et al.*, 2011) e (ANTONAKAKIS *et al.*, 2012).

O ambiente de execução utilizou um processador Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz, 8 GB de memória RAM (6 GB dedicados à ferramenta WEKA) e sistema operacional ArchLinux (kernel 4.8.13-1).

6 ANÁLISE DOS RESULTADOS

As métricas utilizadas foram:

- **Acurácia:** expressa a proximidade entre o valor obtido no experimento e o valor real. Porcentagem de amostras positivas e negativas classificadas corretamente sobre o total das amostras positivas e negativas;
- **Taxa de Positivos Verdadeiros (TPR):** também conhecida como Recall ou taxa de detecção, ela expressa a quantidade de elementos que foram corretamente detectados. P($\hat{Y}$ = 1 | Y = 1), onde $\hat{Y}$ é a classe retornada pelo algoritmo e Y é a classe real do dado;
- **Taxa de Positivos Falsos (FPR):** expressa a quantidade de dados negativos que foram classificados como positivos. Ou seja, a quantidade de alarmes falsos na classificação. P($\hat{Y}$ = 1 | Y = 0);
- **Área sob a curva ROC (AUC):** expressa a relação entre TPR e FPR. Pode ser utilizada para expressar a qualidade da classificação (valores abaixo de 0,6 a classificação falhou; entre 0,6 e 0,7 classificação pobre; entre 0,7 e 0,8 classificação razoável; 0,8 e 0,9 classificação boa e; acima de 0,9 classificação excelente).

O Apêndice A possui as tabelas com os resultados obtidos nos experimentos. Os valores na parte central de cada tabela indicam os resultados de cada classificador binário para aquele cenário. Os valores na parte inferior das tabelas expressam o comportamento médio do cenário seguindo três visões:

- **Geral:** expressa o desempenho geral da classificação dentre todas as amostras analisadas, independente de qual família o bot pertence;
- **Família:** expressa o desempenho da classificação, levando-se em consideração o comportamento médio obtido com as demais famílias de bots;
- **Variação:** expressa o desempenho da classificação, sem considerar os membros da mesma família do cenário.

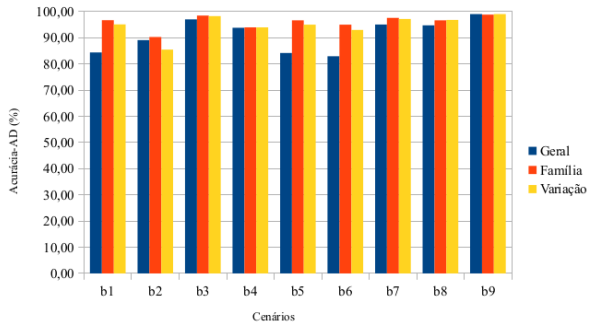
6.1 RESULTADOS PARA O ALGORITMO ÁRVORES DE DECISÃO

Como pode ser observado na Figura 14a, as características básicas sugeridas na metodologia proporcionaram uma acurácia acima de 80% em todos os cenários e independente de considerar a qual família o bot pertence. Para a classificação levando em consideração a família dos bots, a acurácia passa a níveis acima dos 90% todos os cenários.

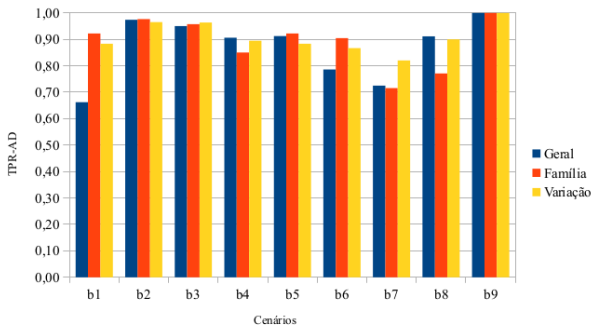
Através das Figuras 14b e 14c, observa-se mais detalhadamente os acertos na predi-

Figura 14 – Comportamento da metodologia com o algoritmo Árvores de Decisão.

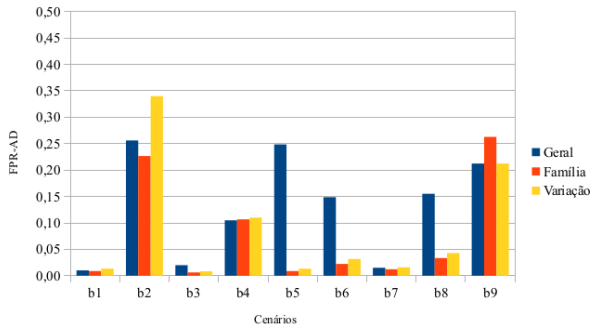
(a) Acurácia



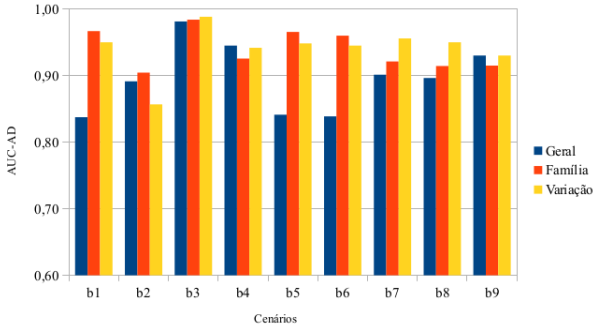
(b) Taxa de Positivos Verdadeiros



(c) Taxa de Positivos Falsos



(d) Área sob a curva ROC



Fonte**:** Elaborado pelo autor.

ção dos cenários. Ainda assim, a predição correta de elementos pertencentes a um dado cenário obteve um valor mínimo de 66%.

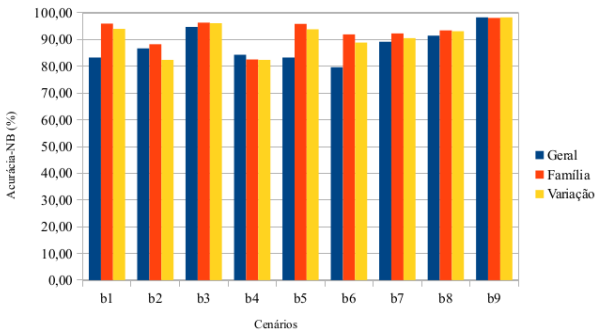
De acordo com a Figura 14d, todos os classificadores tiveram um desempenho, no mínimo, bom (entre 0,8 e 0,9) ainda que desconsidere a qual família o bot pertence. Quanto à classificação em relação à famílias diferentes, esta é considerada excelente (acima de 0,9) em todos os cenários observados.

6.2 RESULTADOS PARA O ALGORITMO NAIVE-BAYES

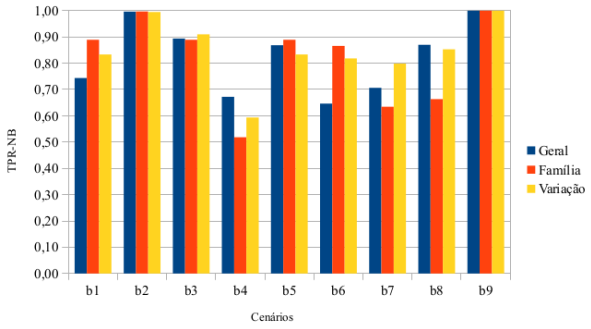
Como pode ser observado na Figura 15, o algoritmo Naive-Bayes apresentou resultados absolutos abaixo daqueles obtidos com Árvores de Decisão. Entretanto, o comportamento geral da metodologia para este algoritmo foi similar ao anterior: a metodologia mostrou-se mais eficaz na predição de bots pertencentes a famílias diferentes.

Figura 15 – Comportamento da metodologia com o algoritmo Naive-Bayes.

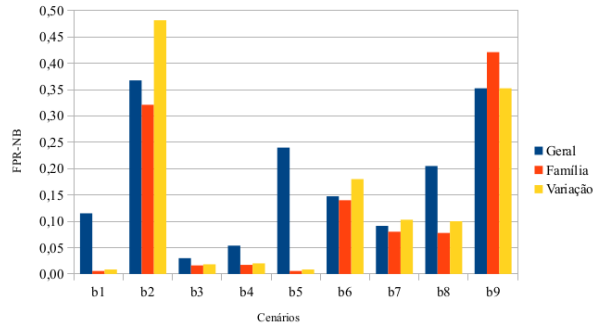
(a) Acurácia



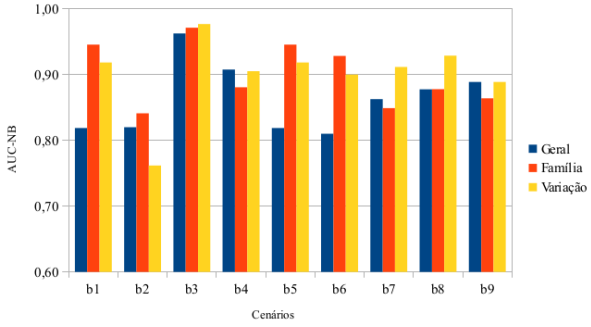
(b) Taxa de Positivos Verdadeiros



(c) Taxa de Positivos Falsos



(d) Área sob a curva ROC



Fonte**:** Elaborado pelo autor.

6.3 O IMPACTO DO PARENTESCO ENTRE BOTS

Durante os experimentos, para algumas famílias, observou-se uma dificuldade em diferenciar amostras de suas variações.

Como visto nas Tabelas 2a e 2c, a metodologia apresentou uma dificuldade ao diferenciar os domínios gerados pelas variações destas famílias. Entretanto, com a Tabela 2b nota-se que a metodologia consegue distinguir entre os elementos da família b2-b3-b4.

Esse impacto sobre a predição de elementos da mesma família pode estar relacionado ao grau de semelhança entre suas variações sob o conjunto de características utilizado para a classificação. Ou seja, quando a metodologia erra ao classificar um elemento da classe-A como sendo classe-B, onde A e B representam variações de uma família de bots, isso significa que o conjunto de carcaterísticas vigente não possui um elemento que expresse a diferença no comportamento observado para estas variações, neste caso, entre os domínios gerados por estes bots.

Além da expansão das carcaterísticas utilizadas, podem ser utilizadas técnicas de agrupamento dos domínios gerados pelos bots aumentando a robustez da metodologia. Esses

Tabela 2 – Resultados agrupados por famílias de bots para o algoritmo Árvores de Decisão.

(a) Família b1-b5-b6

b1-b5-b6	Acurácia-AD(%)	TPR-AD	FPR-AD	AUC-AD
b1-b5	51,15	0,00	0,00	0,50
b5-b1	51,15	1,00	1,00	0,50
b1-b6	53,32	0,00	0,00	0,50
b6-b1	53,32	1,00	1,00	0,50
b5-b6	52,27	1,00	0,91	0,54
b6-b5	52,26	0,09	0,00	0,54

(b) Família b2-b3-b4

b2-b3-b4	Acurácia-AD(%)	TPR-AD	FPR-AD	AUC-AD
b2-b3	100,00	1,00	0,00	1,00
b3-b2	100,00	1,00	0,00	1,00
b2-b4	99,81	1,00	0,01	0,99
b4-b2	99,81	0,99	0,00	0,99
b3-b4	86,61	0,82	0,11	0,92
b4-b3	86,61	0,89	0,18	0,92

(c) Família b7-b8

b7-b8	Acurácia-AD(%)	TPR-AD	FPR-AD	AUC-AD
b7-b8	80,04	0,06	0,01	0,52
b8-b7	80,04	0,99	0,94	0,52

Fonte**:** Elaborado pelo autor.

agrupamentos ocorrem baseados em fluxos, análise temporal ou refinamento na análise léxica dos nomes de domínio.

A necessidade de identificar outros atributos que capturem as nuances entre as demais variações de uma mesma botnet, evidencia uma das características de uma metodologia de correlação vertical: o alto grau de relacionamento com a estrutura das botnets observadas.

Portanto, se uma metodologia busca diferenciar tráfego malicioso de tráfego legítimo, esta deve considerar o impacto causado por variações de um bot e realizar uma análise mais profunda entre elas a fim de identificar atributos que os diferencie.

Outra hipótese para o impacto percebido neste trabalho, é que as diferenças entre as variações não estejam relacionadas ao comportamento dos bots para o protocolo DNS. Ou seja, essas variações podem utilizar o mesmo DGA em sua comunicação e suas diferenças correspondem a outros comportamentos do bot. A utilização do mesmo DGA entre variações de um bot justifica-se uma vez que todos os membros da família deverão contactar o centro de comando e controle daquela botnet.

Porém, nesta metodologia, como todas as classes referem-se a um bot e nenhuma a um tráfego não malicioso, o impacto gerado está apenas na predição de qual botnet foi detectada e não no processo de detecção da infecção em si. Ou seja, um domínio identificado como bot X, independente de pertencer a classe X ou não, continua sendo um domínio envolvido em atividade maliciosa.

7 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou uma arquiteura para a detecção de botnets que utilizam DGAs em sua comunicação, utilizando um conjunto de características consideradas fundamentais em relação aos nomes de domínio gerados pelos bots. Essas características apresentaram um nível de detecção razoável durante os experimentos.

Entretanto, esse conjunto de atributos mostrou-se insuficiente na distinção entre bots de uma mesma família, gerando um aumento no número de falsos positivos. Como a metodologia manipula apenas tráfego malicioso, esses falsos alarmes ainda são identificados como domínios gerados por bots.

Destaca-se que a metodologia apresentada utilizou apenas as informações contidas no tráfego gerado pelos bots que seria descartado na rede: as respostas NXDOMAIN do protocolo DNS. Sob esses dados não foi realizado nenhum tratamento de ruídos e a análise foi realizada sob cada domínio, sem a utilização de técnicas de agrupamento sob as amostras.

A metodologia apresenta os seguintes pontos positivos:

- Pelo fato de basear-se no comportamento DNS apresentado por bots, a metodologia aqui apresentada não depende da arquitetura da botnet (centralizada, P2P ou híbrida);
- Por não inspecionar a carga (*payload*) dos pacotes contendo os comandos e dados manipulados pela botnet, limitando-se apenas ao protocolo DNS, essa metodologia é imune à criptografia, polimorfismo e técnicas semelhantes.

Porém, o foco no comportamento relacionado à utilização do protocolo DNS, em especial aqueles que utilizam DGAs, também traz a principal desvantagem da proposta: bots que não utilizem essa técnica em sua comunicação não podem ser identificados por esta metodologia.

Como sugestões de trabalhos futuros, ficam:

- Identificar uma técnica de detecção de anomalia que seja adequada ao problema apresentado neste trabalho e a implementação da camada de Monitoramento e sua avaliação sob um ambiente real;
- A adição de um sistema de votação a fim de escolher, dentre modelos gerados por algoritmos diferetnes, aquele que mehor identifica um determinado bot;
- Utilização de técnicas de detecção de erros de digitação a fim de reduzir a quantidade de ruídos durante a fase de modelagem;
- A utilização de outras características, capazes de diferenciar melhor as variações que compõem a família de um determinado bot.

REFERÊNCIAS

ANTONAKAKIS, M.; PERDISCI, R.; DAGON, D.; LEE, W.; FEAMSTER, N. Building a dynamic reputation system for dns. In: USENIX CONFERENCE ON SECURITY, 19. 2010, Washington, DC, USA. **Anais…** Berkeley, CA, USA:USENIX Association, 2010. p. 273-289.

ANTONAKAKIS, M.; PERDISCI, R.; NADJI, Y.; VASILOGLOU, N.; ABU-NIMEH, S.; LEE, W.; DAGON, D. From throw-away traffic to bots: Detecting the rise of dga-based malware. In: USENIX CONFERENCE ON SECURITY, 21. 2012. Bellevue, WA, USA. **Anais…** Berkley, CA, USA:USENIX Association, 2012. p. 491-506.

BARTHAKUR, P.; DAHAL, M.; GHOSE, M. K. A framework for p2p botnet detection using svm. In: INTERNATIONAL CONFERENCE ON CYBER-ENABLED DISTRIBUTED COMPUTING AND KNOWLEDGE DISCOVER. Sanya, CN, 2012. **Anais...** Washington, DC, USA: IEEE Computer Society, 2012. p. 195-200.

BILGE, L.; BALZAROTTI, D.; ROBERTSON, W.; KIRDA, E.; KRUEGEL, C. Disclosure: Detecting botnet command and control servers through large-scale netflow analysis. In: ANNUAL COMPUTER SECURITY APPLICATIONS CONFERENCE, 21. 2012, Orlando, FL, USA. **Anais…** New York, NY, USA: ACM, 2012. p. 129-138.

BILGE, L.; KIRDA, E.; KRUEGEL, C.; BALDUZZI, M. EXPOSURE: Finding malicious domains using passive dns analysis. In: ANNUAL NETWORK & DISTRIBUTED SYSTEM SECURITY SYMPOSIUM,18. 2011, San Diego, CA, USA. **Anais eletrônicos…** [S.l.]:The Internet Society, 2011. Disponível em: <http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/bilg.pdf>. Acesso em: 31 dez. 2014.

CAI, T.; ZOU, F. Detecting http botnet with clustering network traffic. In: INTERNATIONAL CONFERENCE ON WIRELESS COMMUNICATIONS, NETWORKING AND MOBILE COMPUTING, 8. 2012, Shanghai, CN. **Anais…** [S.l.]:IEEE Inc., 2012. p. 1-7.

CAVALLARO, L.; KRUEGEL, C.; VIGNA, G. **Mining the Network Behavior of Bots.** Santa Barbara, CA, USA:[S.n.], 2009.

CHOI, H.; LEE, H.; LEE, H.; KIM, H. Botnet detection by monitoring group activities in dns traffic. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION TECHNOLOGY, 7. 2007, Fukushima, JP **Anais...** Washington, DC, USA: IEEE Computer Society, 2007. p.715-720.

CISCO. **Botnets:** The New Threat Landscape. [S.l.: s.n.], 2007.

ERQUIAGA, M. J.; CATANIA, C.; GARCíA, S. Detecting dga malware traffic through behavioral models. In: IEEE BIENNIAL CONGRESS OF ARGENTINA. 2016, Buenos Aires, AR. **Anais…** [S.l.]:IEEE, 2016. p. 1-6.

GU, G.; PERDISCI, R.; ZHANG, J.; LEE, W. Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In: CONFERENCE ON

SECURITY SYMPOSIUM. BERKELEY, 17. 2008, San Jose, CA, USA. **Anais…** Berkley, CA, USA: USENIX Association, 2008. p. 139–154.

GUNTUKU, S. C.; NARANG, P.; HOTA, C. Real-time peer-to-peer botnet detection framework based on bayesian regularized neural network. **CoRR**, 2013, abs/1307.7464. Não paginado. 2013. Disponível em: <http://arxiv.org/abs/1307.7464>. Acesso em: 28 ago. 2015.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The weka data mining software: An update. **SIGKDD Explor. Newsl.** New York, NY, USA, v. 11, n. 1, p. 10-18. nov. 2009.

HOLZ, T.; GORECKI, C.; RIECK, K.; FREILING, F. C. Measuring and detecting fast-flux service networks. In: NETWORK AND DISTRIBUTED SYSTEM SECURITY SYMPOSIUM, 15. 2008. San Diego, CA, USA. **Anais...** [S.l]:The Internet Society. 2008. Não paginado.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 11. 1995. Montreal, CA. **Anais…** San Mateo: Morgan Kaufmann, 1995. p. 338–345.

KARIM, A.; SALLEH, R. S.; SHIRAZ, M.; SHAH, S. A. A.; AWAN, I.; ANUAR, N. B. Botnet detection techniques: review, future trends, and issues. **Journal of Zhejiang University SCIENCE C (Computers & Electronics)**, China, v. 15, n. 11, p. 943–983, 2014.

KUROSE, J. F.; ROSS, K. W. **Computer Networking:** A Top-Down Approach. 6. Ed. [S.l.]: Pearson, 2012.

LEE, J. S.; JEONG, H.; PARK, J. H.; KIM, M.; NOH, B. N. The activity analysis of malicious http-based botnets using degree of periodic repeatability. In: INTERNATIONAL CONFERENCE ON SECURITY TECHNOLOGY, 2008. Hainan Island, CN. **Anais…** [S.l].:IEEE, 2008. p. 83–86.

LUZ, P. M. da. **Botnet Detection Using Passive DNS.** 2013/2014. 52 f. Dissertação (Mestrado em Ciência da Computação) — Radbound University of Nijmegen, 2014.

MURPHY, K. P. **Machine Learning:** A Probabilistic Perspective. Cambridge, Massachusetts:The MIT Press, 2012.

NIGAM, R. A timeline of mobile botnets. In: BOTNET FIGHTING CONFERENCE, 3. 2014. Nancy, France. **Anais…** [S.l.:s.n.]. 2014. Não paginado.

PASSERINI, E.; PALEARI, R.; MARTIGNONI, L.; BRUSCHI, D. Fluxor: Detecting and monitoring fast-flux service networks. In: INTERNATIONAL CONFERENCE ON DETECTION OF INTRUSIONS AND MALWARE, AND VULNERABILITY ASSESSMENT, 5. 2008. Paris. **Anais…** Berlin, Heidelberg: Springer-Verlag, 2008. p. 186–206.

QUINLAN, R. **C4.5:** Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

RIBEIRO, V. de A.; FILHO, R.; MAIA, J. Online traffic classification based on sub-flows. In: IFIP/IEEE INTERNATIONAL SYMPOSIUM ON INTEGRATED NETWORK MANAGEMENT. 2011. Dublin. **Anais…** Dublin: [s.n.], 2011. p. 415–421.

RODRÍGUEZ-GóMEZ, R. A.; MACIá-FERNáNDEZ, G.; GARCÍA-TEODORO, P. Survey and taxonomy of botnet research through life-cycle. **ACM Comput. Surv.**, ACM, New York, NY, USA, v. 45, n. 4, p. 45:1–45:33, ago. 2013.

RUSSELL, S.; NORVIG, P. **The Artificial Intelligence.** 3.ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010.

SALUSKY, W.; DANFORD, R. **Know Your Enemy: FastFlux Service Networks.** 2007. Disponível em: <http://www.honeynet.org/papers/ff/>. Acessso em: 05 jun. 2015.

SCHIAVONI, S.; MAGGI, F.; CAVALLARO, L.; ZANERO, S. Phoenix: Dga-based botnet tracking and intelligence. In: INTERNATIONAL CONFERENCE ON DETECTION OF INTRUSIONS AND MALWARE, AND VULNERABILITY ASSESSMENT, 11. 2014. Enham, UK. **Anais…** Egham, UK:Springer International Publishing, 2014. p. 192–211.

SHANNON, C. E.; WEAVER, W. **The Mathematical Theory of Communication.** Urbana/Champaign, IL, USA: University of Illinois Press, 1949.

SIKORSKI, M.; HONIG, **A. Practical Malware Analysis:** The Hands-On Guide to Dissecting Malicious Software. San Francisco:No Starch Press, 2012.

SOPHOS. **Security Threat Report 2014:** Smarter, Shadier, Stealthier Malware. Oxford, UK, 2014.

STONE-GROSS, B.; COVA, M.; CAVALLARO, L.; GILBERT, B.; SZYDLOWSKI, M.; KEMMERER, R.; KRUEGEL, C.; VIGNA, G. Your botnet is my botnet: Analysis of a botnet takeover. In: ACM Conference on Computer and Communications Security, 16. 2009. Chicago, IL, USA. **Anais…** New York, NY, USA: ACM, 2009. p. 635–647.

STRATOSPHERE, T. **Malware Capture Facility Project. 2015.** Disponível em: <https://stratosphereips.org/category/dataset.html>. Acesso em: 15 jan. 2016.

TIIRMAA-KLAAR, H.; GASSEN, J.; GERHARDS-PADILLA, E.; MARTINI, P. **Botnets.** [S.l.]: Springer Publishing Company. 2013.

WANG, P.; SPARKS, S.; ZOU, C. C. An advanced hybrid peer-to-peer botnet. In: CONFERENCE ON FIRST WORKSHOP ON HOT TOPICS IN UNDERSTANDING BOTNETS. 2007. Cambridge. **Anais…** Berkeley, CA, USA: USENIX Association, 2007. p. 2–2.

WANG, T.; HU, X.; JANG, J.; JI, S.; STOECKLIN, M.; TAYLOR, T. Botmeter: Charting dga-botnet landscapes in large networks. In: INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS, 36. 2016. Nara, JP. A**nais…** [S.l.: s.n.], 2016. p. 334–343.

WEIMER, F. Passive dns replication. In: FIRST CONFERENCE ON COMPUTER SECURITY INCIDENT, 17. 2005. Singapura. **Anais…** [S.l.: s.n.], 2005.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data Mining:** Practical Machine Learning Tools and Techniques. 3. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

WUEEST, C. **Security Response:** The Continued Rise of DDoS Attacks. California, USA, 2014.

WUEEST, C. **Security Response:** The State of Financial Trojans 2014. California, USA, 2015.

ZHAO, D.; TRAORE, I.; SAYED, B.; LU, W.; SAAD, S.; GHORBANI, A.; GARANT, D. Botnet detection based on traffic behavior analysis and flow intervals. **Comput. Secur.**, Elsevier Advanced Technology Publications, Oxford, UK, UK, v. 39, p. 2–16, nov. 2013.

ZHAO, G.; XU, K.; XU, L.; WU, B. Detecting apt malware infections based on malicious dns and traffic analysis. **IEEE Access**, [S.l.]. v. 3, p. 1132–1142, 2015.

APÊNDICE

APÊNDICE A – Tabelas de resultados dos experimentos

As Tabelas 3 a 11 possuem os resultados dos experimentos realizados. Cada tabela possui os valores obtidos nos experimentos para:

- **Acurácia:** expressa a proximidade entre o valor obtido no experimento e o valor real. Porcentagem de amostras positivas e negativas classificadas corretamente sobre o total das amostras positivas e negativas.
- **Taxa de Positivos Verdadeiros (TPR):** também conhecida como Recall ou taxa de detecção, ela expressa a quantidade de elementos que foram corretamente detectados. $P(\hat{Y} = 1 \mid Y = 1)$, onde $\hat{Y}$ é a classe retornada pelo algoritmo e Y é a classe real do dado;
- **Taxa de Positivos Falsos (FPR):** expressa a quantidade de dados negativos que foram classificados como positivos. Ou seja, a quantidade de alarmes falsos na classificação. $P(\hat{Y} = 1 \mid Y = 0)$;
- **Área sob a curva ROC (AUC):** expressa a relação entre TPR e FPR. Pode ser utilizada para expressar a qualidade da classificação (valores abaixo de 0,6 a classificação falhou; entre 0,6 e 0,7 classificação pobre; entre 0,7 e 0,8 classificação razoável; 0,8 e 0,9 classificação boa e; acima de 0,9 classificação excelente).

Como dito anteriormente, esses valores foram computados para os algoritmos de árvore de decisão (AD) e naive-Bayes (NB) e utilizando nível de significância igual a 5%.

Tabela 3 – Resultado cenário b1.

b1	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b1-b2	71,16	65,55	0,30	0,00	0,07	0,00	0,71	0,52
b1-b3	100,00	99,88	1,00	1,00	0,00	0,00	1,00	1,00
b1-b4	99,61	98,77	1,00	1,00	0,01	0,05	0,99	0,99
b1-b5	51,15	51,13	0,00	0,00	0,00	0,00	0,50	0,50
b1-b6	53,32	51,37	0,00	0,95	0,00	0,87	0,50	0,54
b1-b7	100,00	99,83	1,00	1,00	0,00	0,00	1,00	1,00
b1-b8	100,00	99,84	1,00	1,00	0,00	0,00	1,00	1,00
b1-b9	100,00	99,98	1,00	1,00	0,00	0,00	1,00	1,00
Geral	84,41	83,29	0,66	0,74	0,01	0,12	0,84	0,82
Família	96,75	95,96	0,92	0,89	0,01	0,01	0,97	0,95
Variação	95,13	93,98	0,88	0,83	0,01	0,01	0,95	0,92

Fonte**:** Elaborado pelo autor.

Tabela 4 – Resultado cenário b2.

b2	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b2-b1	71,16	65,55	0,93	1,00	0,70	1,00	0,71	0,52
b2-b3	100,00	99,98	1,00	1,00	0,00	0,00	1,00	1,00
b2-b4	99,81	99,22	1,00	1,00	0,01	0,05	0,99	0,99
b2-b5	70,43	64,51	0,93	1,00	0,70	1,00	0,70	0,52
b2-b6	71,45	64,69	0,93	0,97	0,64	0,89	0,73	0,53
b2-b7	100,00	99,79	1,00	1,00	0,00	0,00	1,00	1,00
b2-b8	100,00	99,95	1,00	1,00	0,00	0,00	1,00	1,00
b2-b9	100,00	100,00	1,00	1,00	0,00	0,00	1,00	1,00
Geral	89,11	86,71	0,97	1,00	0,26	0,37	0,89	0,82
Família	90,34	88,26	0,98	1,00	0,23	0,32	0,90	0,84
Variação	85,51	82,42	0,97	1,00	0,34	0,48	0,86	0,76

Fonte**:** Elaborado pelo autor.

Tabela 5 – Resultado cenário b3.

b3	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b3-b1	100,00	99,88	1,00	1,00	0,00	0,00	1,00	1,00
b3-b2	100,00	99,98	1,00	1,00	0,00	0,00	1,00	1,00
b3-b4	86,61	80,93	0,82	0,69	0,11	0,13	0,92	0,84
b3-b5	100,00	99,91	1,00	1,00	0,00	0,00	1,00	1,00
b3-b6	95,80	94,54	0,93	0,87	0,04	0,04	0,99	0,98
b3-b7	96,46	88,67	0,96	0,87	0,01	0,06	0,99	0,95
b3-b8	97,86	94,46	0,95	0,89	0,00	0,01	0,98	0,97
b3-b9	99,75	99,29	0,94	0,83	0,00	0,00	0,97	0,96
Geral	97,06	94,71	0,95	0,89	0,02	0,03	0,98	0,96
Família	98,50	96,32	0,96	0,89	0,01	0,02	0,98	0,97
Variação	98,31	96,13	0,96	0,91	0,01	0,02	0,99	0,98

Fonte**:** Elaborado pelo autor.

Tabela 6 – Resultado cenário b4.

b4	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b4-b1	99,61	98,76	0,99	0,95	0,00	0,00	0,99	0,99
b4-b2	99,81	99,22	0,99	0,95	0,00	0,00	0,99	0,99
b4-b3	86,61	80,94	0,89	0,87	0,18	0,31	0,92	0,84
b4-b5	99,64	98,81	0,99	0,95	0,00	0,00	0,99	0,99
b4-b6	94,65	83,55	0,96	0,44	0,06	0,05	0,98	0,95
b4-b7	86,92	51,34	0,95	0,44	0,58	0,06	0,91	0,82
b4-b8	85,43	67,00	0,77	0,45	0,02	0,01	0,89	0,85
b4-b9	97,78	94,84	0,71	0,33	0,00	0,00	0,89	0,83
Geral	93,81	84,31	0,91	0,67	0,11	0,05	0,95	0,91
Família	93,97	82,57	0,85	0,52	0,11	0,02	0,93	0,88
Variação	94,01	82,38	0,90	0,59	0,11	0,02	0,94	0,91

Fonte**:** Elaborado pelo autor.

Tabela 7 – Resultado cenário b5.

b5	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b5-b1	51,15	51,14	1,00	1,00	1,00	1,00	0,50	0,50
b5-b2	70,42	64,51	0,30	0,00	0,07	0,00	0,70	0,52
b5-b3	100,00	99,90	1,00	1,00	0,00	0,00	1,00	1,00
b5-b4	99,64	98,81	1,00	1,00	0,01	0,05	0,99	0,99
b5-b6	52,27	52,31	1,00	0,95	0,91	0,87	0,54	0,54
b5-b7	100,00	99,84	1,00	1,00	0,00	0,00	1,00	1,00
b5-b8	100,00	99,92	1,00	1,00	0,00	0,00	1,00	1,00
b5-b9	100,00	99,99	1,00	1,00	0,00	0,00	1,00	1,00
Geral	84,19	83,30	0,91	0,87	0,25	0,24	0,84	0,82
Família	96,67	95,87	0,92	0,89	0,01	0,01	0,97	0,95
Variação	95,01	93,83	0,88	0,83	0,01	0,01	0,95	0,92

Fonte**:** Elaborado pelo autor.

Tabela 8 – Resultado cenário b6.

b6	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b6-b1	53,32	51,37	1,00	0,13	1,00	0,05	0,50	0,54
b6-b2	71,44	64,69	0,36	0,11	0,07	0,03	0,73	0,53
b6-b3	95,81	94,54	0,96	0,96	0,06	0,13	0,99	0,98
b6-b4	94,64	83,55	0,94	0,95	0,04	0,56	0,98	0,95
b6-b5	52,26	52,32	0,09	0,13	0,00	0,05	0,54	0,54
b6-b7	98,13	95,29	0,98	0,97	0,02	0,35	0,99	0,97
b6-b8	98,36	96,27	0,98	0,96	0,00	0,01	0,99	0,98
b6-b9	99,56	99,06	0,98	0,96	0,00	0,00	0,99	0,99
Geral	82,94	79,64	0,79	0,65	0,15	0,15	0,84	0,81
Família	95,03	91,92	0,90	0,87	0,02	0,14	0,96	0,93
Variação	92,99	88,90	0,87	0,82	0,03	0,18	0,95	0,90

Fonte**:** Elaborado pelo autor.

Tabela 9 – Resultado cenário b7.

b7	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b7-b1	100,00	99,83	1,00	1,00	0,00	0,00	1,00	1,00
b7-b2	100,00	99,79	1,00	1,00	0,00	0,00	1,00	1,00
b7-b3	96,45	88,67	0,99	0,94	0,04	0,13	0,99	0,96
b7-b4	100,00	51,51	0,42	0,94	0,05	0,56	0,91	0,82
b7-b5	86,93	99,84	1,00	1,00	0,00	0,00	1,00	1,00
b7-b6	98,13	95,29	0,98	0,65	0,02	0,03	0,99	0,97
b7-b8	80,04	79,76	0,06	0,06	0,01	0,01	0,52	0,52
b7-b9	99,04	98,61	0,35	0,06	0,00	0,00	0,80	0,63
Geral	95,07	89,16	0,73	0,71	0,02	0,09	0,90	0,86
Família	97,63	92,31	0,72	0,63	0,01	0,08	0,92	0,85
Variação	97,22	90,51	0,82	0,80	0,02	0,10	0,96	0,91

Fonte**:** Elaborado pelo autor.

Tabela 10 – Resultado cenário b8.

b8	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b8-b1	100,00	99,84	1,00	1,00	0,00	0,00	1,00	1,00
b8-b2	100,00	99,95	1,00	1,00	0,00	0,00	1,00	1,00
b8-b3	97,86	94,46	1,00	0,99	0,05	0,11	0,98	0,97
b8-b4	85,43	66,99	0,98	0,99	0,23	0,55	0,89	0,85
b8-b5	100,00	99,92	1,00	1,00	0,00	0,00	1,00	1,00
b8-b6	98,36	96,27	1,00	0,99	0,02	0,04	0,99	0,98
b8-b7	80,04	79,77	0,99	0,99	0,94	0,94	0,52	0,52
b8-b9	96,19	94,44	0,32	0,00	0,00	0,00	0,79	0,70
Geral	94,74	91,46	0,91	0,87	0,16	0,21	0,90	0,88
Família	96,69	93,42	0,77	0,66	0,03	0,08	0,91	0,88
Variação	96,83	93,12	0,90	0,85	0,04	0,10	0,95	0,93

Fonte**:** Elaborado pelo autor.

Tabela 11 – Resultado cenário b9.

b9	Acurácia-AD (%)	Acurácia-NB (%)	TPR-AD	TPR-NB	FPR-AD	FPR-NB	AUC-AD	AUC-NB
b9-b1	100,00	99,98	1,00	1,00	0,00	0,00	1,00	1,00
b9-b2	100,00	100,00	1,00	1,00	0,00	0,00	1,00	1,00
b9-b3	99,75	99,29	1,00	1,00	0,06	0,17	0,97	0,96
b9-b4	97,78	94,84	1,00	1,00	0,29	0,67	0,89	0,83
b9-b5	100,00	99,99	1,00	1,00	0,00	0,00	1,00	1,00
b9-b6	99,56	99,06	1,00	1,00	0,02	0,04	0,99	0,99
b9-b7	99,04	98,61	1,00	1,00	0,65	0,94	0,80	0,63
b9-b8	96,19	94,44	1,00	1,00	0,68	1,00	0,79	0,70
Geral	99,04	98,28	1,00	1,00	0,21	0,35	0,93	0,89
Família	98,88	98,08	1,00	1,00	0,26	0,42	0,92	0,86
Variação	99,04	98,28	1,00	1,00	0,21	0,35	0,93	0,89

Fonte**:** Elaborado pelo autor.