Predizendo o Perigo: Alerta Antecipado Contra a Propagação de *Worms* Utilizando o Algoritmo das Células Dendríticas

Dilton D. de Oliveira[1], Ricardo J. P. de B. Salgueiro[1], Edward D. Moreno[1]

[1]Departamento de Computação - Universidade Federal de Sergipe (UFS)
Aracaju – SE – Brasil

{dilton.dantas,ricardo.salgueiro,edwdavid}@gmail.com

Abstract. *The self-reproduction and self-propagation capabilities of worms can make them spread rapidly and infecting thousands of vulnerable computers before any effective countermeasure can be taken either by users or network administrators. Therefore, a key point for the defense of this type of malware is its early detection. Danger Theory immune-inspired approaches have used the Dendritic Cell Algorithm (DCA) successfully in solving various problems related to computer security. Their accuracy and performance, especially in intrusion detection in hosts, has attracted the attention of researchers. This paper proposes an Early Warning System (EWS) against worm propagation, based on a cooperative approach of the DCA. The idea is to antecipate the process of detecting worm infection, through the exchange of alerts among the neighboring networks, in time to countermeasures be adopted. The gain in time obtained with the early warning was used in the Conficker worm propagation model and the results showed a reduction in the number of infected machines and, consequently, in the worm propagation.*

Resumo. *As capacidades de autoreprodução e autopropagação dos* worms *podem fazê-los se espalharem rapidamente e infectarem milhares de computadores vulneráveis, antes que qualquer contra-medida efetiva possa ser tomada por usuários ou administradores das redes. Por isso, um ponto chave para defesa desse tipo de* malware *está na sua detecção precoce. Abordagens imuno-inspiradas na Teoria do Perigo tem utilizado o Algoritmo das Células Dendríticas (DCA) com sucesso na solução de diversos problemas relacionados à segurança computacional. Sua precisão e desempenho, especialmente na detecção de intrusão em* hosts*, tem atraído a atenção dos pesquisadores. Este trabalho propõe um Sistema de Alerta Antecipado (EWS) contra propagação de* worms*, baseado em uma abordagem cooperativa do DCA. Trata-se de antecipar o processo de detecção de infecção por* worms*, através da troca de alertas entre redes vizinhas, em tempo de serem adotadas medidas de contenção. O ganho de tempo obtido com esse alerta precoce foi utilizado no modelo de propagação do* worm *Conficker e os resultados apontaram uma redução no número de máquinas infectadas e, consequentemente, na propagação deste* worm.

1. Introdução

Os ataques mais destrutivos contra a segurança das redes de computadores têm sido realizados por programas maliciosos (*malwares*) autopropagantes, conhecidos como *worms* (vermes) [WEAVER et al. 2003]. As características automáticas de sua ação

permitem que eles se reproduzam e se propaguem pelas redes sem nenhuma intervenção humana, de forma semelhante aos vermes biológicos que atacam o corpo humano.

As estratégias de defesa contra *worms* se dividem em detecção da intrusão e contenção da propagação. Diversas soluções têm sido propostas na literatura nestas duas frentes de combate. Na primeira, basicamente através de sistemas de detecção de intrusão (*Intrusion Detection Systems* - IDS) e na segunda através de Sistemas de Alerta Antecipado (*Early Warning Systems* - EWS)[GRASSO 2006].

Assim como a biologia inspirou o ataque por *worms*, inspirou também a sua defesa, através de sistemas imunes artificiais (*Artificial Immune Systems* – AIS) [CASTRO et al. 2002], ou seja, sistemas de defesa inspirados no comportamento do sistema imunológico humano. Esse conceito tem sido usado para criação de sistemas de detecção de intrusão imuno-inspirados.

A base de tal inspiração está nas teorias existentes sobre o funcionamento do sistema de defesa humano. A maioria delas está fundamentada na idéia da distinção entre o que é próprio e o que não é próprio do organismo para ativar uma resposta imune. Por outro lado, a Teoria do Perigo [MATZINGER 1994], se baseia na idéia de que uma resposta imune é ativada apenas quando ocorre um dano à célula, ou seja, quando a célula está sob uma situação de perigo. A detecção de tal situação de perigo é feita por um grupo especial de células, chamadas Células Dendríticas, através do cruzamento de diversas informações moleculares (sinais e antígenos).

A funcionalidade das células dendríticas inspirou a criação de um algoritmo que tem tido bons resultados quando aplicado em sistemas de detecção de intrusão, o Algoritmo das Células Dendríticas – DCA [GREENSMITH 2007]. Este algoritmo, correlaciona informações de um *host* na forma de sinais e antígenos, identifica contextos anômalos, seu grau de severidade e o seu causador.

Este trabalho utilizou o DCA para correlacionar não só informações do próprio *host*, mas também informações (alertas) enviadas por *hosts* vizinhos em uma rede, de maneira colaborativa, a fim de antecipar a identificação de um contexto anômalo e permitir que máquinas ainda não infectadas possam tomar medidas de precaução. Tal estratégia foi aplicada em um EWS, usando alertas emitidos por *hosts* infectados com o *worm* Conficker [LAWTON 2009] como um dos sinais de entrada no DCA e os resultados mostraram que é possível predizer sua infecção e conter sua ação.

Além desta sessão introdutória, este trabalho apresenta na sessão 2 a Teoria do Perigo; na sessão 3 o Algoritmo das Células Dendríticas; na sessão 4 os Sistemas de Alerta Antecipado e seus requisitos; na sessão 5 o modelo de propagação dos *worms*; na sessão 6 é detalhado o EWS proposto; na sessão 7 são descritos os experimentos realizados e resultados obtidos; e na sessão 8, as conclusões e trabalhos futuros.

2. Teoria do Perigo

A Teoria do Perigo (*Danger Theory* – DT), proposta por Matzinger (1994), tenta explicar a natureza e o funcionamento das respostas imunes do corpo humano de uma maneira diferente da visão mais clássica e difundida da Seleção Negativa, que se baseia da discriminação entre o que é próprio e o que não é próprio do organismo. A DT afirma que a resposta imune é uma reação aos estímulos que o corpo reconhece como

danoso, e não uma simples reação ao que não é próprio dele. Dessa forma, células imunes e estranhas podem existir juntas, contrariando a visão tradicional.

A hipótese da DT é que as células que morrem por naturalmente ou que sofrem danos enviam um sinal de alarme que se propaga cobrindo uma área em torno da célula, criando uma "zona de perigo". Algumas células, chamadas de Células Apresentadores de Antígenos (*Antigen Presenting Cells* – APC) recebem esse sinal, ficam estimuladas e estimulam as células do sistema imune adaptativo para que entrem em ação.

As Células Dendríticas (DC) são um grupo de APCs responsáveis por instruir o sistema imunológico a responder apropriadamente às ameaças percebidas, através da combinação de uma infinidade de informações moleculares e interpretando essas informações para as células T do sistema imune adaptativo. Elas combinam a evidência de um dano celular com o antígeno suspeito coletado. São detectores naturais de anomalias, cuja forma de agir inspirou a criação de um algoritmo que tem sido usado por AISs para detecção de intrusão baseados em anomalias [AL-HAMMADI et al., 2008; GREENSMITH, 2007; GREENSMITH et al., 2006].

3. Algoritmo das Células Dendríticas (DCA)

A criação de um modelo abstrato para o comportamento das DCs foi o primeiro passo para o desenvolvimento de um sistema de detecção de intrusão inspirado na Teoria do Perigo. O Algoritmo das Células Dendríticas [GREENSMITH, 2007] é um algoritmo de base populacional, projetado para lidar com tarefas de detecção baseadas em anomalias. É inspirado por funções das células dendríticas naturais do sistema imune inato, que fazem parte da primeira linha de defesa do corpo contra invasores.

O objetivo do DCA é correlacionar diversos fluxos de dados coletados em uma Célula Dendrítica Artificial (*Artificial Dendrític Cell* – aDC) na forma de sinais (evidência de danos) e antígenos (responsáveis pelos danos), e rotular grupos de antígenos idênticos como 'normais' ou 'anômalos'. Os principais sinais recebidos são:

- **Sinais PAMP (*PAMP Signals* – PS)** - *Pathogen Associated Molecular Pattern* (Padrões Moleculares Associados ao Patógeno) – são indicadores confidentes de uma situação anormal, ou seja, o aumento na concentração destes sinais conduz ao aumento de CSM, migrando a célula para o seu estado maduro, a fim de ativar a resposta imune.

- **Sinais de Perigo (*Danger Signals* - DS)** - o aumento da sua concentração eleva o CSM, causando a maturação da célula para o estado maduro e induzindo à formação de um contexto de perigo com menos potência e menor confidência que os sinais PAMP.

- **Sinais Seguros (*Safe Signals* – SS)** – é interpretado como uma ocorrência normal do sistema, aumentando o sinal de saída que a torna semimadura e diminuindo o valor do sinal que a torna madura (sinais PAMP e de perigo) para prevenir falsos alarmes.

Desse modo, dependendo da concentração de sinais recebidos pela aDC, esta pode se tornar madura (quando a aDC possui maior concentração de sinais de perigo ou PAMP), para ativar a resposta imune; ou semimadura (quando a aDC possui maior concentração de sinais seguros), para suprimí-la.

A potência dos sinais, ou seja, seu grau de influência nos resultados, é representada no algoritmo através da distribuição de pesos, que podem assumir valores positivos ou negativos. Por exemplo, atribui-se o sinal negativo ao peso do sinal SS,

para dar um efeito supressor nos demais sinais, a fim de prevenir falsos positivos. O processamento dos sinais de entrada é feito através da Eq. 1 e gera três sinais de saída: CSM (moléculas coestimulatórias), citocinas semi-maduras e citocinas maduras.

$$O_j = \sum_{i=0}^{n} (W_{ij} * S_i), \forall_j \quad (1)$$

onde O_j é o sinal de saída de índice j; i o índice da categoria do sinal de entrada; n é o número de categorias de sinais de saída menos um; W_{ij} é o peso da categoria de sinal de entrada i para o cálculo do sinal de saída O_j; e S_i o número de sinais de entrada de categoria i. Quando a soma dos valores de CSM calculados por cada aDC ultrapassa um limiar de migração, a aDC muda seu estado e se torna madura.

O diferencial do DCA é que ele fornece também informações que representam o quão anômalo é um grupo de antígenos, e não apenas se um item de dados é anômalo ou não. Isto é conseguido através da geração de um valor de coeficiente de anomalia, denominado *Mature Context Antigen Value* – MCAV. Este coeficiente corresponde a um valor entre 0 e 1, onde quanto mais próximo de 1 maior o grau de anomalia.

A versão determinística do DCA (dDCA) requer um menor número de parâmetros e menos recursos computacionais. O Algoritmo 1 apresenta o pseudo-código do dDCA que foi utilizado neste trabalho.

Algoritmo 1: Pseudo-código do Algoritimo das Células Dendríticas Determinístico (dDCA) (GREENSMITH et al., 2008).

```
entrada : antigenos e sinais
saída : tipos de antígenos e Kα

define tamPopulação de DCs;
inicializa DCs;
enquanto há dados de entrada
    escolha entrada
        caso antigeno
            antigenoContador++;
            celulaIndice = agContador %= tamPopulacao;
            DC->antigeno de indice cellIndex++;
            atualiza o perfil do antigeno da DC;
        fim
        caso sinal
            calcula csm e k;
            para cada DC
                DC.tempo -= csm;
                DC.k += k;
                se DC.tempo <= 0 então
                    armazena antigeno, DC.k;
                    renova DC;
                    atualiza MCAV;
                fim
            fim
        fim
    fim
fim
para cada tipo de antigeno
    calcula métrica de anomalia Kα;
fim
```

A variável antigeno corresponde aos antígenos que estão sendo analisados, a variável sinal corresponde aos sinais de entrada, DC é um conjunto de células dendríticas, csm e k são os dois sinais de saída derivados da transformação dos sinais de entrada, calculados através das Eq. 2 e 3; MCAV o grau de anomalia do antígeno, calculado através da Eq. 4, conforme proposto em Greensmith e Aickelin (2008).

$$csm = SS + DS \quad (2)$$

$$k = DS - (2*SS) \quad (3)$$

$$MCAV_{\alpha} = \frac{M}{Ag} \quad (4)$$

onde $MCAV_{\alpha}$ é o MCAV para o antígeno do tipo α, M é o número de antígenos maduros do tipo α e Ag é o número total de antígenos do tipo α. Essa métrica retorna um valor entre zero e um, com a probabilidade do antígeno ser anômalo aumentando quando tende a um. A versão determinística apresenta ainda a métrica K_{α}, calculada através da Eq. 5, que usa as magnitudes dos valores de k, gera valores reais de anomalias e permite polarizar processos normais e anômalos.

$$K_{\alpha} = \frac{\sum k_i}{\sum \alpha_i}, \forall_i \quad (5)$$

onde α_i é o número de antígenos do tipo α amostrados pela DC i, e k_i é o seu valor de k cumulativo. Quanto maior o valor do K_{α} maior a probabilidade do antígeno α representar uma intrusão.

Diferentemente das demais abordagens disponíveis na literatura, o dDCA foi implementado e utilizado neste trabalho com o objetivo de antecipar a identificação de ataques lançados por máquina infectadas com *worms*.

4. Sistemas de Alerta Antecipado (EWS)

Um Sistema de Alerta Antecipado (EWS - *Early Warning System*) é utilizado comumente no alerta de fenômenos naturais ameaçadores que são emitidos tão antecipadamente que as potenciais vítimas têm a possibilidade de reagir, de forma que os danos pessoais possam ser evitados ou reduzidos [BASTKE et al., 2009]. Compreendem a integração de quatro elementos: conhecimento do risco, monitoramento e predição, disseminação da informação e resposta [GRASSO, 2006]. Estes sistemas têm sido utilizados principalmente para minimização de danos ocasionados por fenômenos naturais, como na predição de terremotos, *tsunamis*, atividades vulcânicas e, mais recentemente, também na predição de ataques de segurança a infraestruturas de TIC [ENGELBERTH et al., 2010; KOSSAKOWSKI et al., 2006], como os ataques de *worms* nas redes ou na Internet.

4.1 Modelos de EWS

Um EWS é composto basicamente por uma rede de sensores posicionados em áreas representativas de uma ou mais redes de computadores, que coletam amostras de dados referentes a ataques de segurança e as enviam para uma central de monitoramento, chamada *Early Warning Central* – EWC (Central de Alerta Antecipado). A Figura 5.1 ilustra alguns modelos de EWSs [THEILMANN, 2010].

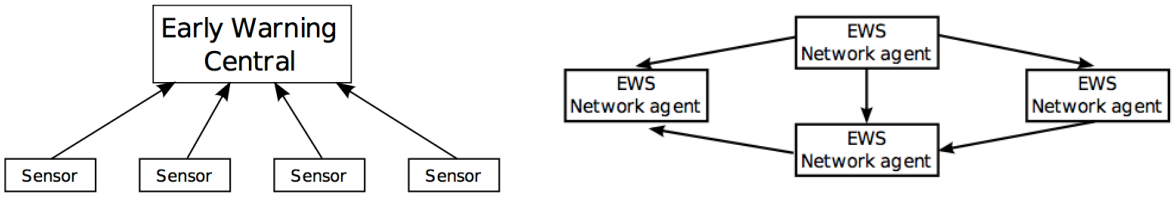

Figura 4.1: Modelos de EWSs [THEILMANN, 2010].

Como é possível observar na Figura 4.1 (a), um EWS é organizado basicamente de forma hierárquica. A desvantagem dessa organização é que quando se tenta desenvolver uma rede de sensores muito grande não é simples trabalhar em um ambiente que exige distribuição administrativa e cooperativa, ou seja, onde os nodos de rede e os dados coletados estão sobre diferentes domínios administrativos. E em redes diferentes os administradores têm objetivos, motivações e políticas diferentes.

Este trabalho, porém, utiliza uma forma não hierárquica, distribuída e colaborativa de organização, a abordagem Herold [THEILMANN, 2010], que utiliza EWS *Network Agents* (Agentes da Rede EWS) distribuídos, ao invés de uma EWC. Por esta abordagem os EWS *Network Agents* compartilham entre si conhecimento do ambiente, objetivos e funcionalidades, conforme ilustra a Figura 4.1 (b). Nela, cada EWS *Network Agents* pode requisitar e oferecer dados para outros agentes. Isso permite que os agentes contribuam com seus dados para múltiplos EWSs, em uma abordagem cooperativa de monitoramento da rede, sendo mais adequada ao mecanismo de troca de mensagens de alerta.

4.2 Requisitos de um EWS

Para cumprir sua missão um sistema de alerta antecipado precisa atender à alguns requisitos básicos, como: pelo menos dois sistemas de informação distintos no cyberspaço (SI_1 e SI_2); pelo menos duas instâncias de tempo (T_1 e T_2; onde $T_1 < T_2$ = "*early*"); disponibilização de informação útil para prevenção de danos ("*warning*"); e antecipação, ou seja, a informação deve estar disponível de forma antecipada (uma informação enviada de SI_1 deve chegar em SI_2 em um tempo menor que T_2) [ZOU et al., 2006]. Esses requisitos foram utilizados para validar o EWS aqui proposto.

5. Propagação de *Worms*

A fim de avaliar o EWS na contenção da propagação de *worms* é necessário analisar o modelo de propagação desse tipo de *malware*.

A Figura 5.1 ilustra o ciclo de ação de um *worm*. A primeira ação de um *worm* após a infecção de um *host* é buscar novos alvos. Isso é feito através de varreduras (*scans*) buscando por *hosts* vulneráveis em espaços de endereçamento IP aleatórios. Ao encontrar novos alvos, ele lança o ataque pela exploração da vulnerabilidade dos sistemas. Obtendo sucesso, ele inicia o processamento da infecção, realizando uma série de alterações nesses sistemas e se autorreplicando, ou seja, enviando uma cópia de si mesmo para o alvo. Em seguida, ele reinicia o ciclo buscando outros novos alvos.

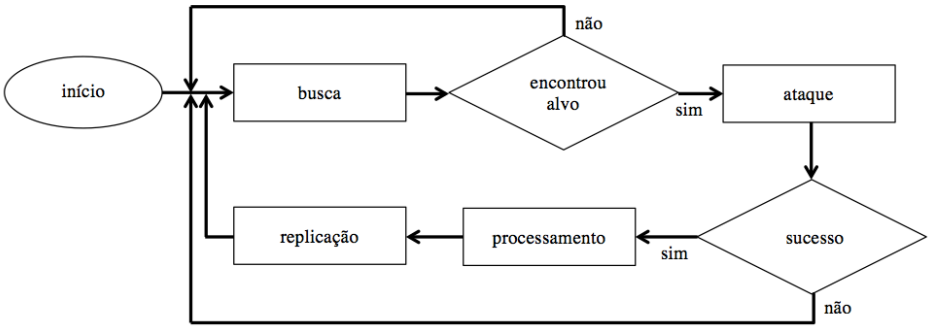

Figura 5.1: Ciclo de ação de um *worm*.

Outra característica comum dos *worms* é que o seu estágio inicial de propagação se inicia de forma lenta, com uma baixa taxa de infecção. No estágio seguinte, à medida que novos alvos são descobertos, a taxa de infecção aumenta consideravelmente. Depois, tende a estabilizar à medida que as vulnerabilidades vão sendo corrigidas e o universo de alvos potenciais diminui. A Figura 5.2 ilustra esse comportamento.

Desse modo, todas as ações de detecção e contenção do *worm* devem ser realizadas ainda em seu estágio inicial de propagação, ou seja, quando as primeiras máquinas são infectadas. Se considerarmos que nesse estágio as máquinas infectadas formam uma zona de perigo e que o envio de alertas a partir delas para seus *hosts* vizinhos vulneráveis indica a aproximação desta zona, podemos basear a detecção na identificação desta situação de perigo, a fim de predizer uma iminente infecção.

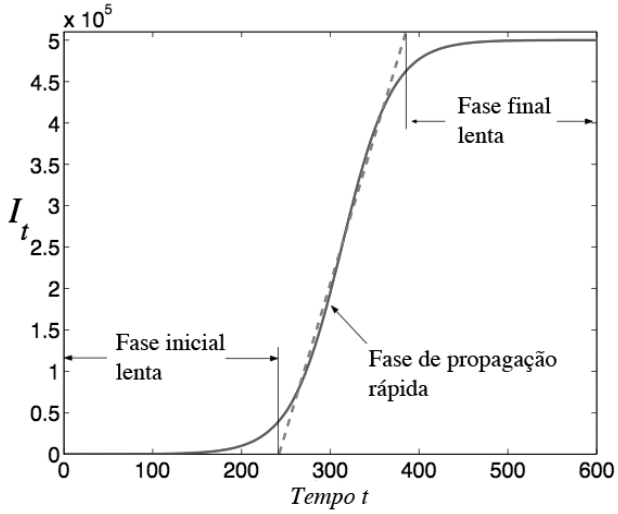

Figura 5.2: Modelo de propagação dos worms (Adaptado de: ZOU 2003).

Denotando o tempo médio que um *host h* vulnerável leva para detectar sua infecção por *Ti(h)* e o tempo médio que ele leva para receber um alerta por *Ta(h)*, para satisfazer os requisitos de um EWS, descritos na sessão 4, é preciso garantir:

$$Ta(h) < Ti(h) \quad (6)$$

Para esta análise foi utilizado o modelo analítico de propagação de *worms* ativos (*Analytical Active Worm Propagation* – AAWP) apresentado em [CHEN et al. 2003]. Este modelo é adequado para análise do *worm* Conficker, por exemplo, pois caracteriza a propagação de *worms* que enviam varreduras aleatórias. Ele se vale do modelo de tempo discreto e de aproximação determinística para descrever esta propagação.

A ação de propagação de um *worm* ocorre conforme a Figura 5.1. Para defesa de uma máquina são aplicadas correções (patches) da vulnerabilidade de segurança que permite sua infecção. Quando uma máquina é corrigida ela se torna invulnerável.

A aceleração da propagação é conseguida através da ideia de "lista alvo" ("*hitlist*") , ou seja, antes de lançar um *worm* o atacante reúne uma lista de máquinas vulneráveis em potencial. Ao infectar a primeira máquina o *worm* inicia sua varredura na lista de máquinas, infectando-as. Esgotada esta lista ele cria uma nova lista alvo.

O modelo AAWP assume que o *worm* lança sua varredura para várias máquinas ao mesmo tempo e que uma máquina já infectada não é re-infectada. Assume também que ele leva um instante de tempo para completar sua infecção em uma máquina, simplificando o modelo sem afetar os resultados.

No AAWP, o número *n* de máquinas infectadas em um instante *i+1* é dado por:

$$n_{i+1} = (1-d-p)n_i + [(1-p)^i N - n_i][1-(1-\frac{1}{2^{32}})^{sn_i}] \quad (7)$$

onde $i \geq 0$, N é o número de máquinas vulneráveis, s é o número médio de máquinas varridas por uma máquina infectada por unidade de tempo, d é a taxa em que a infecção é detectada em uma máquina e eliminada sem correção, e p é a taxa em que uma máquina infectada ou vulnerável se torna invulnerável.

6. O EWS Proposto

Como visto antes, este trabalho propôe um EWS que utiliza um modelo não hierárquico, com a função de monitoramento distribuída entre vários *EWS Network Agents*, que trocam informações entre si em forma de mensagens de alerta. O processo de análise dessas informações em cada *EWS Network Agent* é realizado por uma unidade de detecção de intrusão composta por instâncias do DCA, que coletam informações do *host* local e os alertas recebidos de nodos vizinhos para identificar um contexto de perigo.

Também como já visto, o primeiro sinal de infecção por um *worm* está na sua primeira ação no *host* infectado: a varredura por novos alvos. Desse modo, os dados para detecção de infecção foram mapeados em três instâncias de sinais coletados durante essa ação. Por outro lado, neste EWS, o dDCA foi implementado com uma instância adicional de sinal PAMP, para representar os dados de uma possível infecção, coletados durante a recepção de alertas. Estes sinais foram mapeados da seguinte forma:

• **Sinal de PAMP 1 (PS1)**: número de pacotes ICMP com mensagens de erro "*destination unreachable*" recebidas por segundo, já que na busca por novos alvos uma máquina infectada por *worm* lança uma varredura para endereços IP aleatórios. Como muitos desses IPs não possuem máquinas ativas, mensagens desse tipo são retornadas para a máquina infectada. Isso significa que o aumento no número de pacotes ICMP com "destination unreachable" representa uma constatação de uma situação anormal;

• **Sinal de Perigo (DS)**: número de pacotes TCP em relação ao número total de pacotes enviados pelo *host* infectado. A varredura por novos alvos é realizada através do envio de uma grande quantidade de pacotes TCP SYN, que normalmente ocorrem em pequeno número em relação ao total de pacotes enviados por um *host* não infectado. Isso significa que quando o número de pacotes TCP aumenta em relação ao número total de pacotes enviados por um *host*, há um indício de uma situação de anormalidade;

• **Sinal Seguro (SS)**: número de pacotes enviados com tamanho normal, pois, para que a varredura do *worm* seja rápida, ele lança pacotes de tamanho menor que os geralmente utilizados nas redes, ou seja, um aumento nesse número representa um situação normal.

• **Sinal de PAMP 2 (PS2)**: número de alertas recebidos de *hosts* vizinhos infectados. A recepção de um sinal de alerta indica a chegada da zona de perigo em torno da máquina vulnerável, ou seja, o aumento desse número representa uma situação de anormalidade.

7. Experimentos

Para mostrar a viabilidade desta proposta, primeiro foi utilizado o DCA para detectar uma infecção por *worm* e validar sua eficácia neste tipo de problema. Em seguida, os dados foram confrontados com o modelo formal de propagação dos *worms*, considerando os requisitos de um EWS para predição do ataque.

Um ambiente de simulação, utilizando a ferramenta NeSSi2 – *Network Security Simulator* – [SCHMIDT 2010], foi montado para extrair dados de ataques do *worm*. Esta ferramenta permite criar redes IP compostas por diversos componentes (clientes, roteadores, etc), trocar pacotes entre si, e simular diversos ataques de segurança, como infecção por *worms*, *botnets*, negação de serviço (DoS), etc. Do NeSSi2 foram extraídos os conjuntos de dados para serem mapeados em sinais de entrada no DCA.

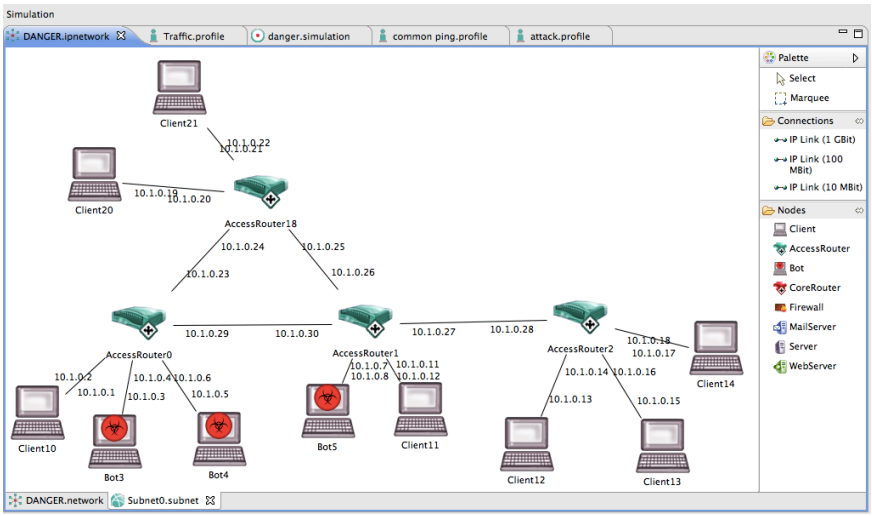

Figura 7.1: Interface gráfica do NeSSi2 com a topologia utilizada.

A topologia de rede utilizada e o ambiente de simulação são apresentados na Figura 7.1. Nesta topologia, cada roteador de acesso (*AccessRouter*) corresponde a um EWS *Network Agent*, e os clientes (*Clients*) e *bots* correspondem aos sensores coletores dos dados que serão enviados para os roteadores de acesso.

7.1. Aplicação do DCA

Nos experimentos foi utilizada uma implementação em C da versão determinística do DCA (dDCA), mais especificamente uma variação dessa versão que usa segmentação [GU et al. 2009]. Nela, ao invés da análise ser realizada somente após o processamento de todos os dados da célula, ela é dividida em segmentos (subconjuntos e dados) para ser executada a cada iteração, juntamente com o processamento dos dados. Isso deixa o dDCA capaz de realizar a análise em tempo real, já que um sistema de detecção de intrusão efetivo e totalmente funcional deve identificar intrusões o mais rápido possível.

O DCA foi executado em um computador com processador Intel Core 2 Duo de 2.13 GHz, com 4 GB de memória DDR3 e sistema operacional Mac OS versão 10.8.2, utilizando o compilador C do Xcode. Foi utilizada na fase de pré-processamento um conjunto de dados coletados de uma máquina infectada por *worm* no NeSSi2, composto por *logs* de ataques de SYN *scan* que a mesma passou a realizar após sua infecção.

A captura dos sinais foi feita a cada segundo. Os tipos de antígenos correspondem aos números dos processos (PID) coletados à partir de chamadas de sistema durante a infecção, sendo eles o processo Nmap (utilizado para *scans*) e o SSH (para acesso remoto), utilizado como uma aplicação comum executando em paralelo.

Os sinais e os parâmetros do dDCA foram atribuídos da mesma forma que em [GU et al. 2009], com o valor dos sinais normalizados entre 0 e 100, o tamanho da população de DC em 100, o limiar de migração igual a 12 vezes um número entre 1 e 100 e tamanho do segmento também igual a 100. Os pesos utilizados também foram os mesmos, conforme a tabela 1.

Tabela 1. Pesos dos sinais usados nos experimentos.

	PS	**DS**	**SS**
CSM	4	2	6
k	8	4	-12

Os resultados de K_α para cada antígeno α coletado num conjunto de dados com cerca de 3.000 instâncias de antígenos, estão ilustrados na tabela 2.

Tabela 2. Resultados de *K* para cada antígeno α

Processos	K_α									
Nmap	242.0	242.0	234.0	441.0	312.0	328.0	356.0	272.0	352.0	312.0
SSH	-241.0	-242.0	-149.0	-327.0	-214.0	-192.0	-233.0	-126.0	-326.0	-128.0

Os valores positivos de K_α se referem à identificação de perigo, aqui associado ao Nmap, enquanto os valores negativos não apresentam perigo, associados ao SSH. Estes resultados ilustram a detecção de uma infecção por *worm* realizada pelo dDCA.

O aumento no número de alertas em um *host* vulnerável faz o dDCA identificar um contexto de perigo (anomalia) antes dele ser infectado, possibilitando alguma ação impeditiva contra a ameaça. Isto pode ser verificado no gráfico da Figura 7.2, que ilustra as medições de tempo realizadas em dez execuções do dDCA com um grau de confiança de 95%. A curva de tracejo mais fino ilustra o tempo de detecção medido com o primeiro sinal de PAMP (infecção) e a curva de tracejo mais espesso ilustra o tempo de detecção medido usando adicionalmente o segundo sinal de PAMP (alerta).

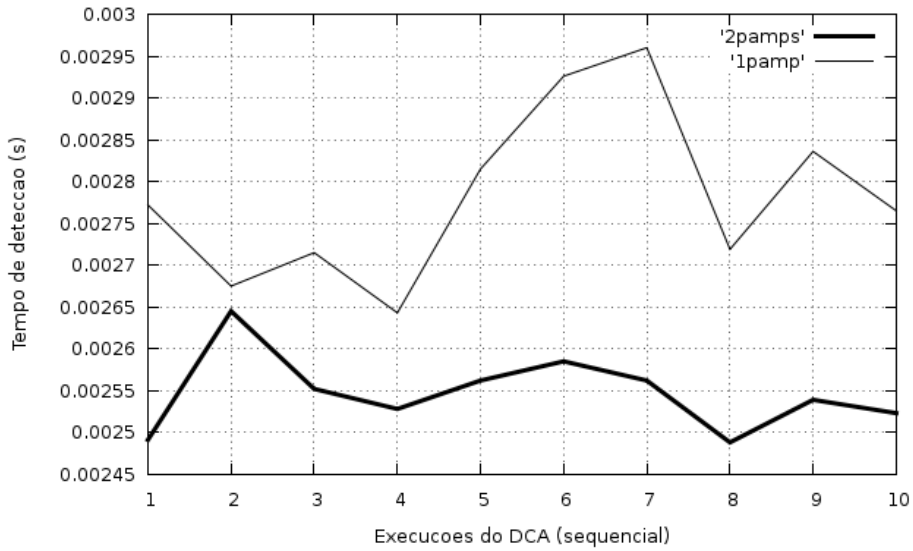

Figura 7.2: Tempos de detecção de uma anomalia com os sinais de PAMP 1 e PAMP 2.

Com base nos valores médios observa-se um ganho de aproximadamente 8,45% nos tempos de detecção quando utilizado adicionalmente o sinal de PAMP 2 (alertas). Isso é explicado porque o recebimento dos alertas antecede a detecção, ou seja, o aumento destes sinais ocorre antes do aumento dos sinais PAMP 1 (infecção). Desse modo, a recepção de alertas antecipa a formação de um contexto de perigo.

7.2. Alerta Antecipado do *Worm* Conficker

O *worm* Conficker é interessante por possuir alta capacidade de infecção e rápida propagação em larga escala. Detectado inicialmente em 2008, também chamado de Downup, Downadup ou Kido, tornou-se um dos *worms* mais ativos [LAWTON, 2009], e atualmente continua se propagando. Sua infecção é conseguida pela exploração de uma vulnerabilidade conhecida do sistema operacional Microsoft Windows.

Para confirmar a Eq. 6 como verdadeira, considere que no EWS o recebimento de alertas em uma máquina implica na correção da vulnerabilidade. Toma-se, então, o ganho percentual médio de 8,45% no tempo de detecção do *worm*, conseguido com a utilização do dDCA como o sinal de PAMP 2, referente ao recebimento de alertas por máquinas vulneráveis e aplica-se ao valor da taxa de correção p.

A Figura 7.3 apresenta os resultados obtidos com a utilização da Eq. 7, para os valores de $p = 0$, $p = 0{,}0005$ (utilizados no trabalho de Chen (2003)) e $p = 0{,}00054225$ (valor com o ganho médio de 8,45%), respectivamente. Observa-se que quando $p = 0$, ou seja, quando não há aplicação de correção, o número de *hosts* infectados se mantém muito grande. Quando a taxa $p = 0{,}0005$ de correção é utilizada o número de *hosts* infectados tende a diminuir. E para a taxa $p = 0{,}00054225$ observa-se que o número de *hosts* infectados diminui ainda mais. Os demais valores considerados no referido trabalho foram mantidos por serem compatíveis com os dados de propagação do Conficker, conforme dados reais coletados em [THE CAIDA UCSD Network Telescope] em Novembro de 2008, durante os seus primeiros dias de propagação. Foi considerado um número N de 1.000.000 de máquinas vulneráveis, uma taxa s de 100 varreduras por segundo e uma taxa d de máquinas infectadas eliminadas sem correção de 0.001 por segundo.

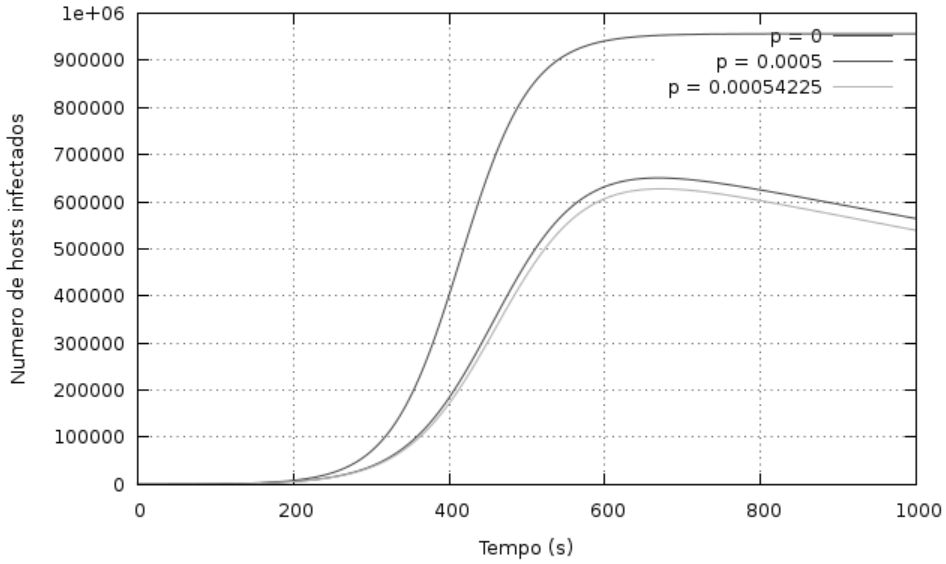

Figura 7.3: *Hosts* infectados por segundo para $p = 0$, $p = 0{,}0005$ e $p = 0{,}00054225$.

Também sob as mesmas condições empregadas na análise dos resultados exibidos na figura 7.3, na Figura 7.4 estão ilustrados os resultados para uma taxa de correção $p = 0$, para $p = 0{,}001$, também utilizado no trabalho de Chen (2003), e para $p = 0{,}0010845$, que representa o valor com o ganho médio de 8.45%, devido ao uso do sinal PAMP adicional de alerta antecipado.

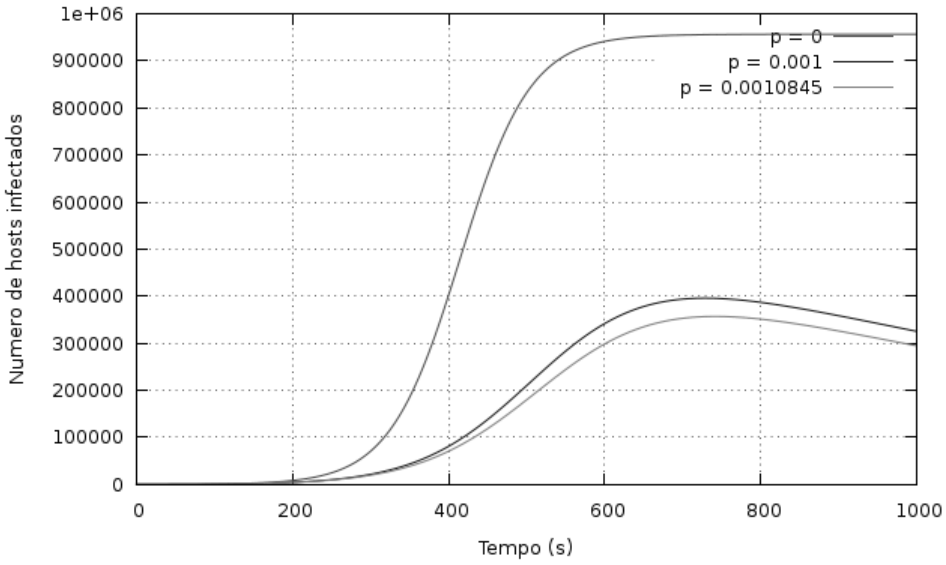

Figura 7.4: *Hosts* infectados por segundo para $p = 0$, $p = 0{,}001$ e $p = 0{,}0010845$.

Como pode ser observado nos gráficos das Figuras 7.3 e 7.4, para os valores de $p = 0{,}00054225$ e $p = 0{,}0010845$, respectivamente, correspondendo ao incremento de 8,45% na taxa de correção, conseguido com o recebimento de alertas, ocorre uma diminuição no número de *hosts* infectados, principalmente comparado ao caso em que nenhuma correção é aplicada ($p = 0$). Isto significa que a utilização de alertas antecipados consegue a contenção da propagação do *worm* mais rapidamente.

É importante observar ainda que *p* se refere à taxa de correção em um universo de máquinas infectadas ou vulneráveis. No entanto, as correções ocasionadas pelo recebimento de alertas só são efetivas em máquinas vulneráveis, ou seja, máquinas ainda não infectadas, para que haja antecipação à infecção. Isso significa que quanto menor o número de máquinas infectadas maior será a efetividade do EWS, reforçando a necessidade da contenção da propagação ocorrer o mais antecipadamente possível.

A análise aponta ainda que para valores muito grandes de tempo as curvas tendem a 0, ou seja, o ponto em que ocorre a contenção definitiva do *worm* (seu estágio de inatividade) é bastante lenta. Como o uso de alertas antecipados faz a curva chegar a 0 antes, o EWS também acelera o processo que torna o *worm* inativo.

8. Conclusões e Trabalhos Futuros

Este trabalho propôs um Sistema de Alerta Antecipado contra a propagação de *worms*, imuno-inspirado na Teoria do Perigo, que utiliza o seu Algoritmo das Células Dendríticas. Os resultados mostraram que o dDCA pode ser usado colaborativamente, utilizando sinais de alerta recebidos de *hosts* vizinhos mapeados em sinais PAMP, para antecipar a formação de um contexto de perigo e predizer uma infecção do *worm* Conficker a tempo de realizar ações de proteção e, então, conter a sua propagação. O Algoritmo das Células Dendríticas, portanto, se mostrou viável no alerta antecipado contra ataques de segurança e uma importante recurso para correlação de eventos.

Ficam para trabalhos futuros, identificar formas de envio rápido de mensagens de alerta entre *hosts*, a detecção de *worms* que utilizam o envio de *scans* de modo não aleatório, a implementação do dDCA segmentado em uma solução de tempo real e uma comparação com o desempenho de outras soluções de EWS tradicionais.

Referências

Aickelin, U.; Greensmith, J.; and Twycross, J. (2007). Immune system approaches to intrusion detection - A review. *Natural Computing*. Vol. 6, p. 413-466, December.

Al-Hammadi, Y.; Aickelin, U.; and Greensmith, J. (2008). DCA for Bot Detection. *In Proceedings of the IEEE World Congress on Computational Intelligence (WCCI2008)*, p. 1807-1816, Hong Kong.

Bastke, S.; Deml, M.; Schmidt, S. (2009). Internet Early Warning Systems – Overview and Arquitecture. *In proceedins of First European Workshop of Internet Early Warning and Network Intelligence (EWNI 2010)*. Hambug, Germany.

Castro, L.; Timmis, J.., A. Whitbrook, and U. Aickelin (2002), Artificial Immune Systems - A New Computational Aproach. Springer-Verlag, London,UK, September.

Chen, Z; Gao, L.; Kwiat, K. (2003). Modeling the Spread of Active *Worms*. *In Proceedings of IEEE INFO- COMM*, volume 3, pages 1890 – 1900.

Engelberth, M.; Freiling, F. C.; Gobel, J.; Gorecki. C, Holz, T.; Hund, R.; Trinius, P.; Willems, C. (2010). The InMAS Approach. *In Proceedings of the 1st Workshop on Early Warning and Network Intelligence (EWNI)*, Hamburg, Germany, February.

Grasso, V. F. (2006). Early Warning Systems: State-of-Art Analysis and Future Directions. *Draft report of United Nations Environment Programme (UNEP)*.

Greensmith, J. (2007). *The Dentritic Cell Algorithm*. Tese (Doutorado), School of Computer Science, University of Nottingham.

Greensmith, J.; Aickelin, U. (2007). Dentritic Cell for SYN Scan Detection. *In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007)*, p. 49-50.

Greensmith, J.; Aickelin, U. (2008). The Deterministic Dendritic Cell Algorithm. *In Proceedings of the 7th international conference on Artificial Immune Systems (ICARIS '08)*, Springer-Verlag, Berlin, Heidelberg, 291-302.

Greensmith, J.; Twycross, J.; Aickelin, U. (2006). The Dentritic Cell Algorithm for Anomaly Detection. *In Proceedings of the IEEE Congress on Evolutionary Computation (CEC2006)*, p.664-671. Vancouver, Canada.

Gu, F.; Greensmith, J.; and Aickelin, U. (2009). Integrating Real-Time Analysis With The Dendritic Cell Algorithm Through Segmentation. *In Genetic and Evolutionary Computation Conference (GECCO)*.

Kossakowski, K.; Sander, J.; Grobauer, B.; Mehlau, J. I. (2006). Carmentis: A co-operative approach towards situation awareness and early warning for the Internet. *In Proceedings of IMF'06*, volume 97 of LNI, pages 55–66. GI.

Lawton, G. (2009). On the Trail of the Conficker *Worm*. Computer, vol.42, no.6, p. 19-22, June.

Matzinger, P. (1994). Tolerance, Danger, And The Extended Famil. *Annu. Rev. Immunology,* v. 12, p. 991–1045, April.

Schmidt, S.; Bye, R.; Chinnow, J.; Bsufka, K.; Camtepe, A.; Albayrak, S. (2010). Application-level Simulation for Network Security". SIMULATION, Vol. 86 No. 5-6, p. 311-330.

Silva, G. S. (2009). "Detecção de Intrusão em Redes de Computadores: Um algoritmo imunoinspirado baseado na teoria do perigo e células dendríticas". Dissertação (Mestrado). Universidade Federal de Minas Gerais, UFMG, Belo Horizonte, MG.

The CAIDA UCSD Network Telescope Two Days in November 2008 Dataset - < November 12 andd 19, 2008 >, http://www.caida.org/data/passive/telescope-2days-2008_dataset.xml. - Acesso em 27/11/2012.

Theilmann, A. (2010). Beyond centralism: The Herold Approach to Sensor Networks and Early Warning Systems. *In proceedins of First European Workshop of Internet Early Warning and Network Intelligence (EWNI 2010)*. Hambug, Germany.

Uchôa, J. Q. (2009). "Algoritmos Imunoinspirados Aplicados em Segurança Computacional". Tese (Doutorado). Universidade Federal de Minas Gerais, UFMG, Belo Horizonte, MG.

Weaver, N.; Paxson, V.; Staniford, S. (2003). A taxonomy of computer *worms*. *In Proceedings of the 2003 ACM Workshop on Rapid Malcode*. Washington D.C., USA. ACM press, p. 11-18.

Zou, C. C.; Gao, L.; Gong, W.; Towsley, D. (2003). Monitoring and early warning for internet *worms*. *In Proceedings of the 10th ACM conference on Computer and communications security*. Washington D.C., USA.