GTS Grupo de Trabalho em Segurança

Implementação Combinada entre Algoritmos Imunológicos para Detecção de Intrusão

Oscar Aleixo Costa Rocha (LabCOM-UFMG)
Alessandro Vivas Andrade (UFVJM/LabCOM-UFMG)
Luciano de Errico (LabCOM-UFMG)

30 de Junho de 2007

Linfócito T (azul), identificando a assinatura de uma molécula atraves de uma célula dendrítica (verde) no processo de sinapse.
BERKELEY LAB - http://www.lbl.gov/Science-Articles/Archive/PBD-immune-system.html

Agenda

- Introdução e Motivação
- Sistemas de Detecção de Intrusão (IDS)
- Sistema Imunológico Humano
- Sistemas Imunológicos Artificiais (SIA)
- Implementação
- Simulações e Resultados

Introdução

Os Sistemas Imunológicos Artificiais (SIA) são sistemas computacionais que utilizam como inspiração os sistemas imunológicos biológicos. O sistema proposto neste trabalho consiste de um Sistema de Detecção de Intrusão (IDS), atuando diretamente no kernel do sistema operacional Linux e utilizando algoritmos imunológicos de seleção negativa e seleção clonal de forma cooperativa.

Motivação

- Aumento do interesse pelo tema Segurança Computacional;
- A prevenção de intrusão é uma das formas de garantir os principais aspectos de segurança;
- Crescimento do interesse em sistemas computacionais bio inspirados;
- Semelhança entre o Sistema Imunológico e os Sistemas de Segurança Computacionais.

IDS

Sistemas de Detecção de Intrusão tem por finalidade a detecção e análise de varreduras, ataques e invasões em sistemas computacionais.

- Baseados em Assinaturas ou Eventos

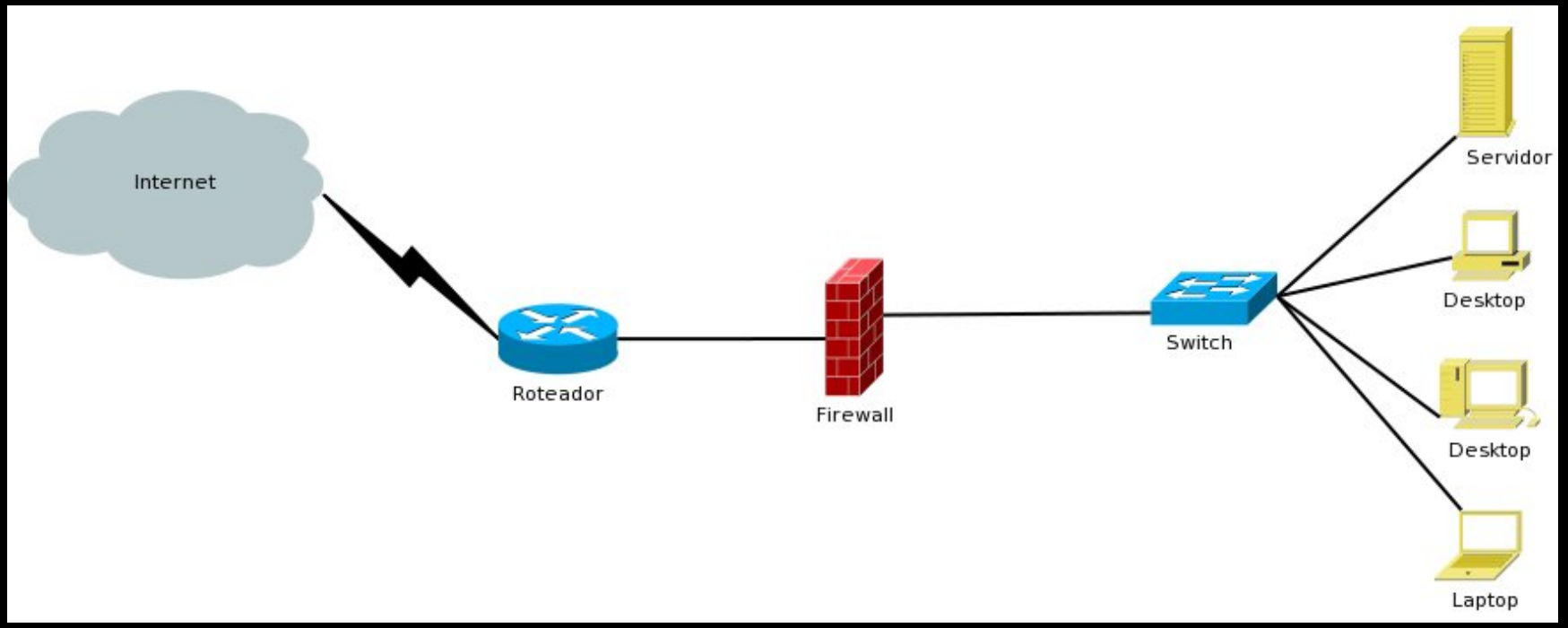
  As Assinaturas presentes em todo ataque são Caracteristicas Únicas.

- Detectam Ataques e Varreduras

  Firewalls atuam sob ataques, já IDS atuam sobre varreduras e ataques.

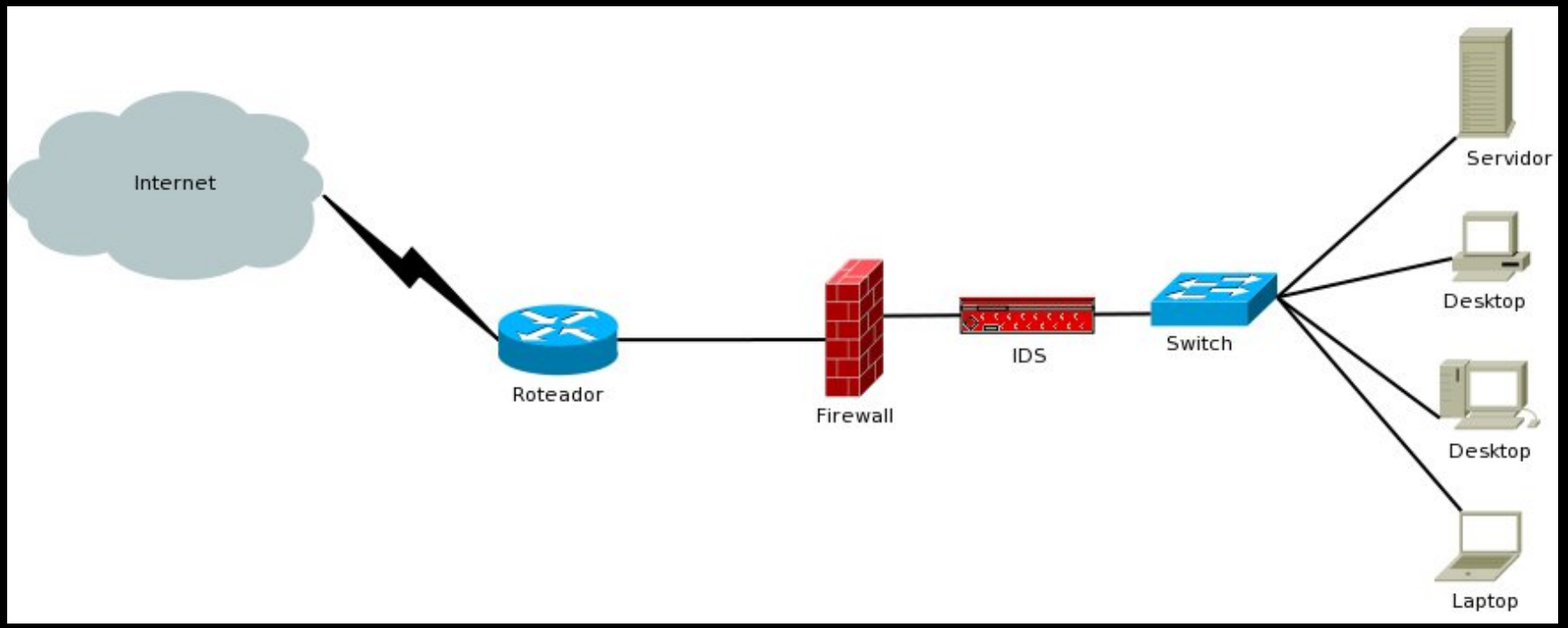
IDS - Tipos

- HIDS *(Host-based Intrusion Detection System)*



IDS - Tipos

- NIDS *(Network-based Intrusion Detection System)*



IDS - Tipos

- IDS Ativo ou IPS *(Intrusion Prevention System)*

  Respondem às tentativas de invasão:

  (Fecha a conexão, reprograma o firewall, bloqueia IP, ...)

- IDS Passivo

  Alertam sobre as tentativas de invasão:

  (Logs, e-mail, sms, alerta, ...)

Sistema Imunológico

Responsável pela proteção do organismo contra agentes infecciosos.

Características:

- *Especificidade:* Existe uma resposta para cada ataque.
- *Diversidade:* Capacidade de reconhecer milhares de microorganismos.
- *Sensibilidade:* Alta sensíbilidade a substâncias estranhas.
- *Memória:* Capacidade de aprendizagem e reconhecimento.

Sistema Imunológico

Dividido em **Inato** e **Adaptativo**.

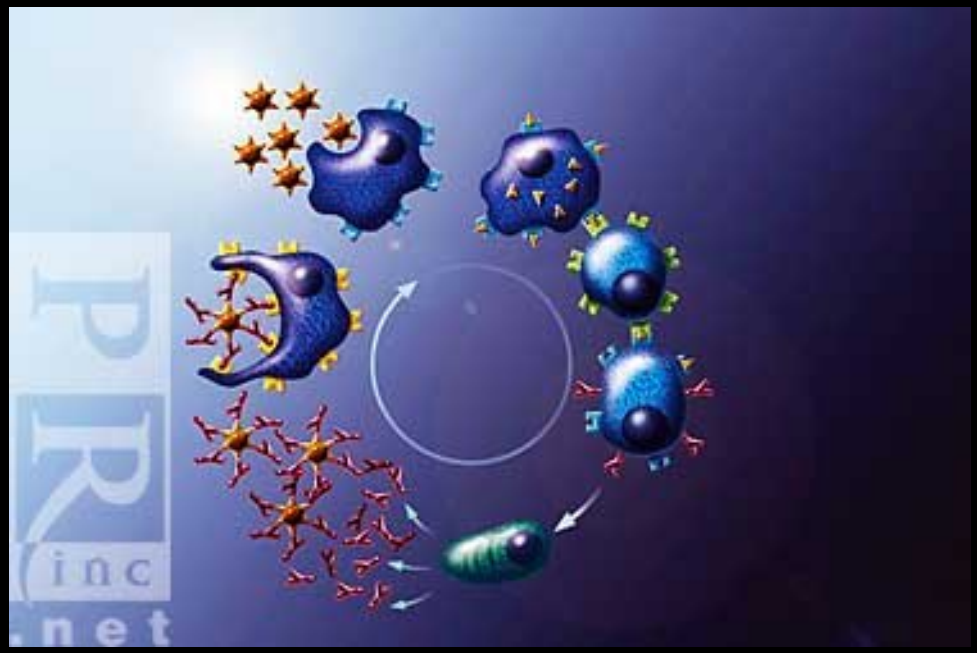


Imagem 1: Macrófagos (azul) atacando as bactérias (verde). Imagem 2: Ciclo da resposta imunológica adaptativa.
Photo Researches: http://www.photoresearchers.com/

Sistema Imunológico

- O Sistema Imunológico Inato é a primeira linha de defesa do Sistema Imunológico e permanece inalterada desde o nascimento do organismo [Hofmeyer, 2000].
- O Sistema Imunológico Adaptativo consiste na elaboração de uma resposta imunológica mais eficaz, já que esta elaboração é realizada a partir de partes de Antígenos dos Patógenos.

SIA

- Os Sistemas Imunológicos Artificiais utilizam como inspiração o Sistema Imunológico Humano.

  Processos biológicos -> Algoritmos Imunológicos

- São amplamente utilizados em computação.

  Data mining, otimização, robótica, reconhecimento de padrões e aprendizagem de máquinas, etc...

SIA - Seleção Negativa

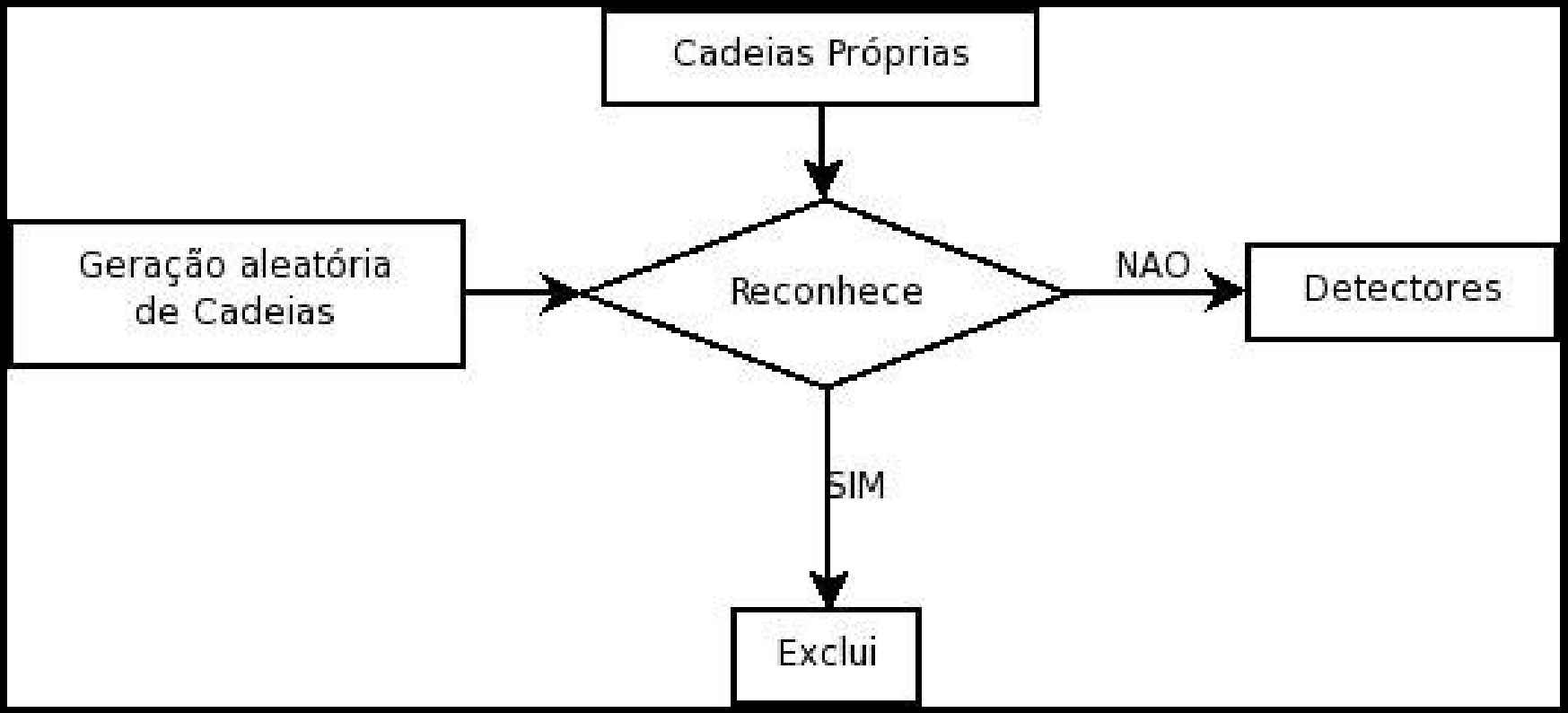
O algoritmo de Seleção Negativa visa resolver o problema de destinguir o *próprio (falso positivo)* do *não próprio (falso negativo)*, utilizando como inspiração a criação dos Linfócitos T [Forrest et al, 1994; 1997].

Áreas de aplicação:

- detecção de vírus;
- reconhecimento de tráfego positivo ou negativo.

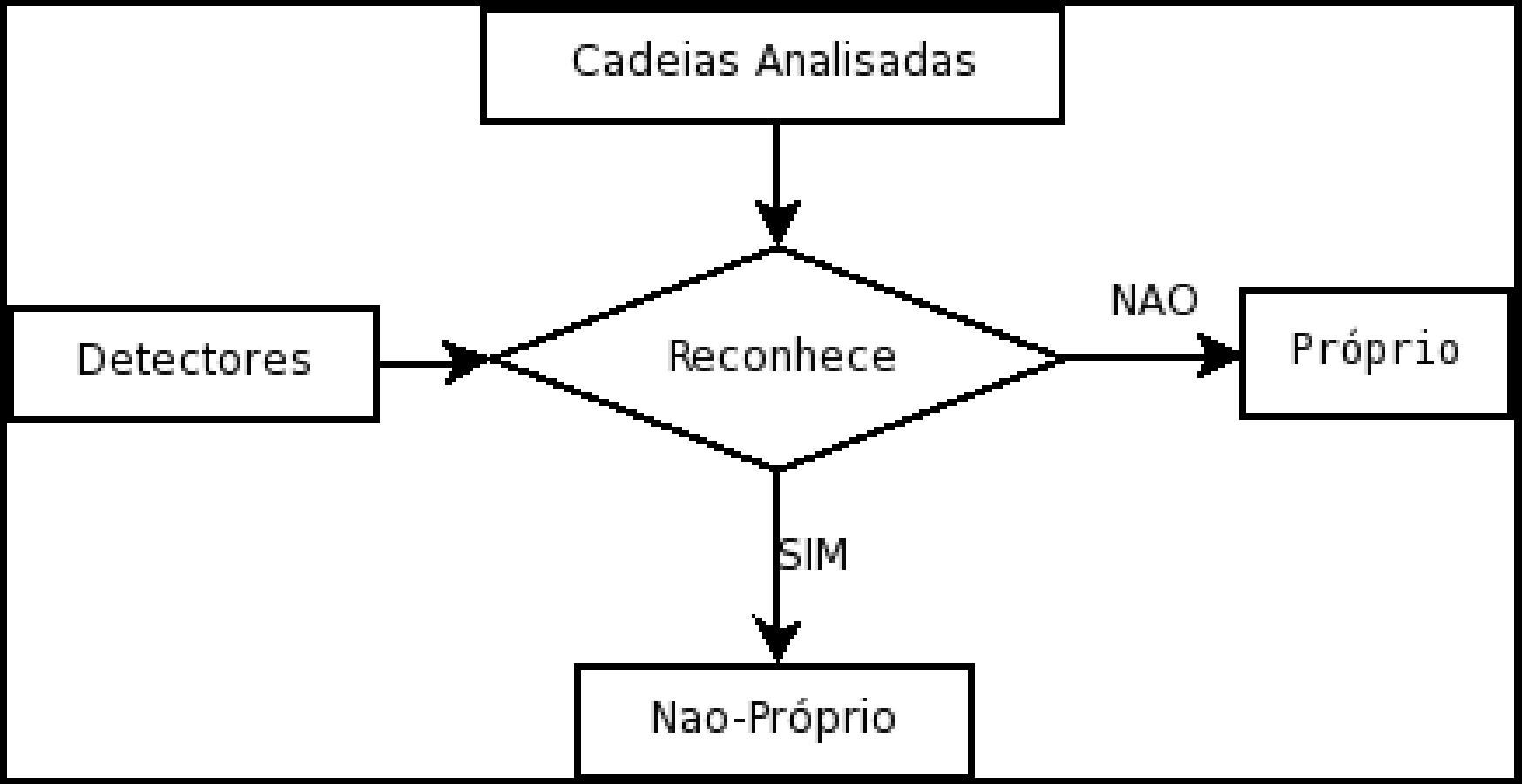
SIA - Seleção Negativa – 1ª Fase

A primeira fase do algoritmo de Seleção Negativa é onde são definidos os detectores incapazes de identificarem cadeias próprias.



SIA - Seleção Negativa – 2ª Fase

A segunda fase do algoritmo de Seleção Negativa consiste na detecção das cadeias não próprias pelos detectores gerados pela primeira fase.



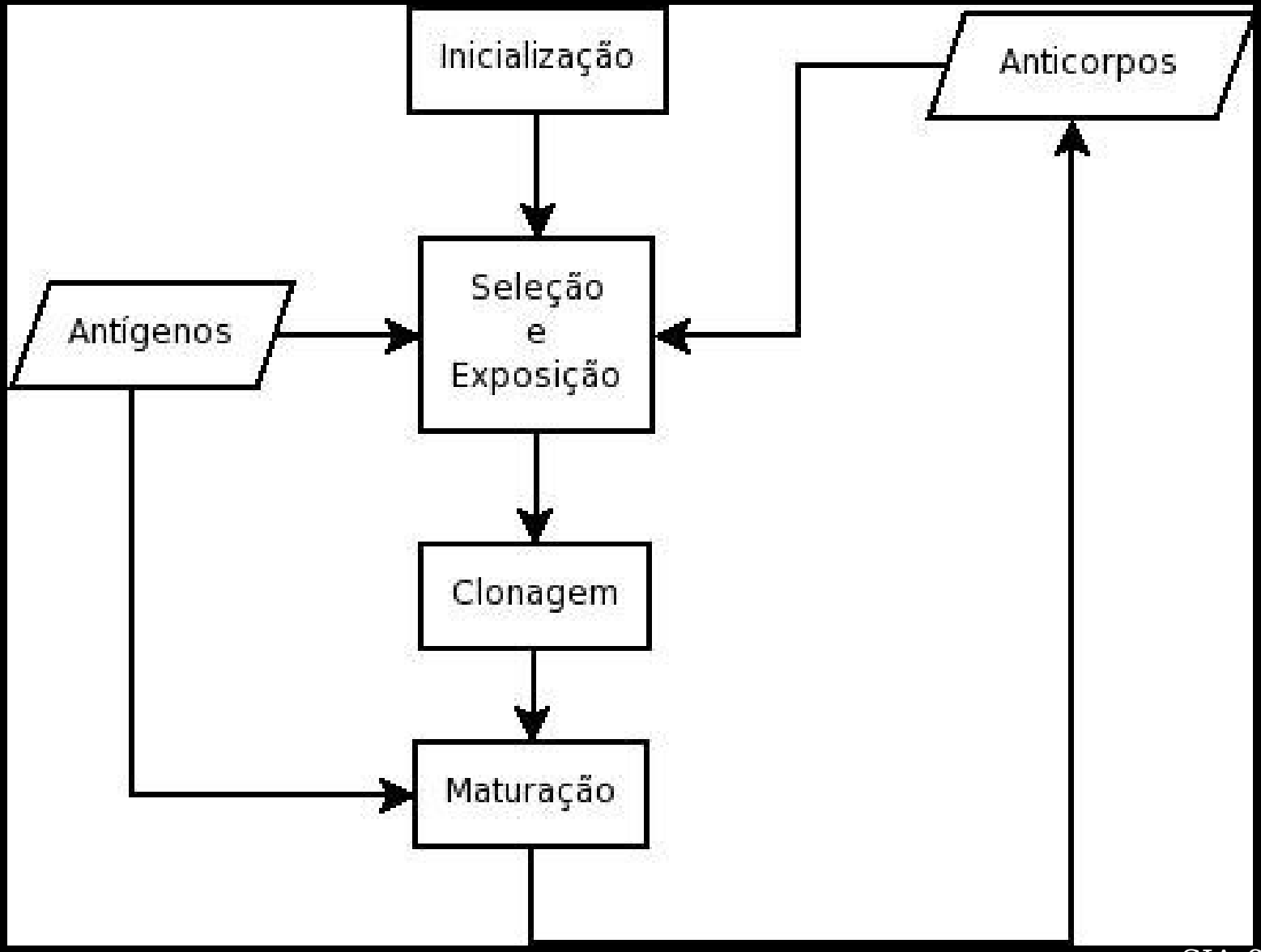
SIA - Seleção Clonal

O algoritmo de seleção clonal é inspirado no processo de seleção clonal, realizado durante a resposta imune adaptativa do sistema imunológico [De Castro et alli, 2000].

Objetivos:

- Criação de novos Anticorpos, cada vez mais reativos aos Antígenos que os originou.
- A cada iteração, os Anticorpos com menor afinidade são substituídos por Anticorpos com maior afinidade.

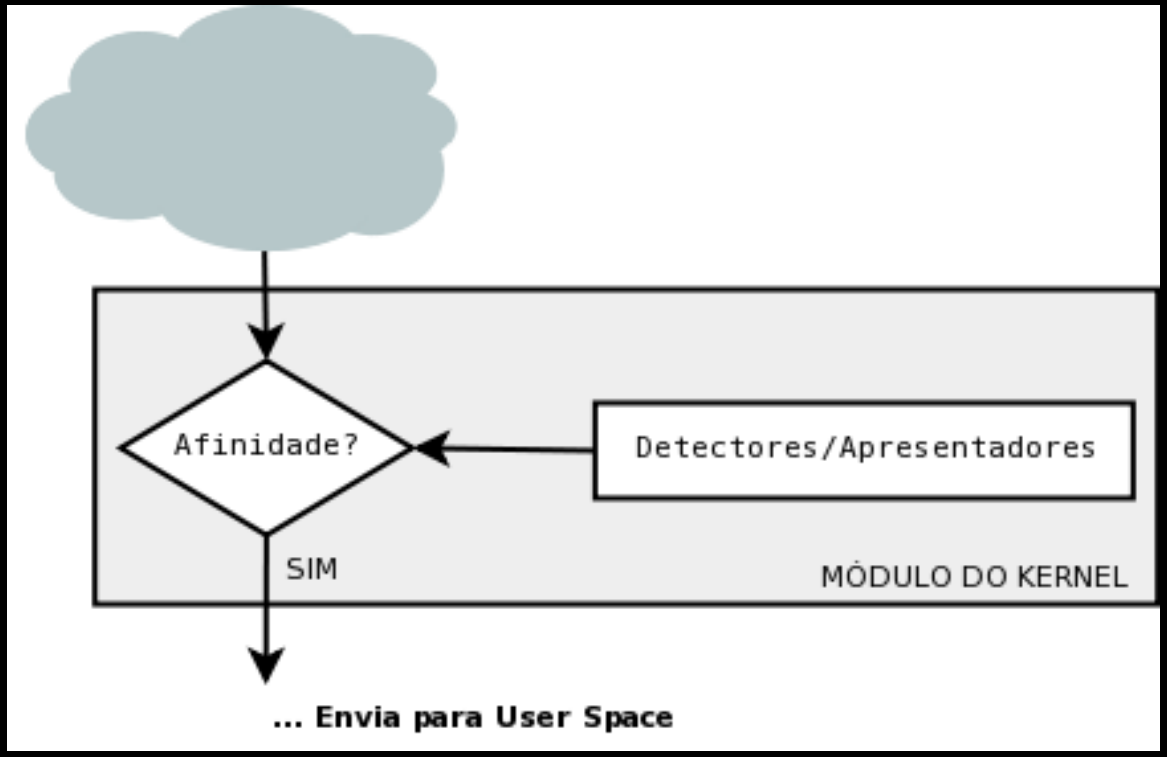
SIA - Seleção Clonal



Implementação - Definições

Módulo do Kernel (detecção)

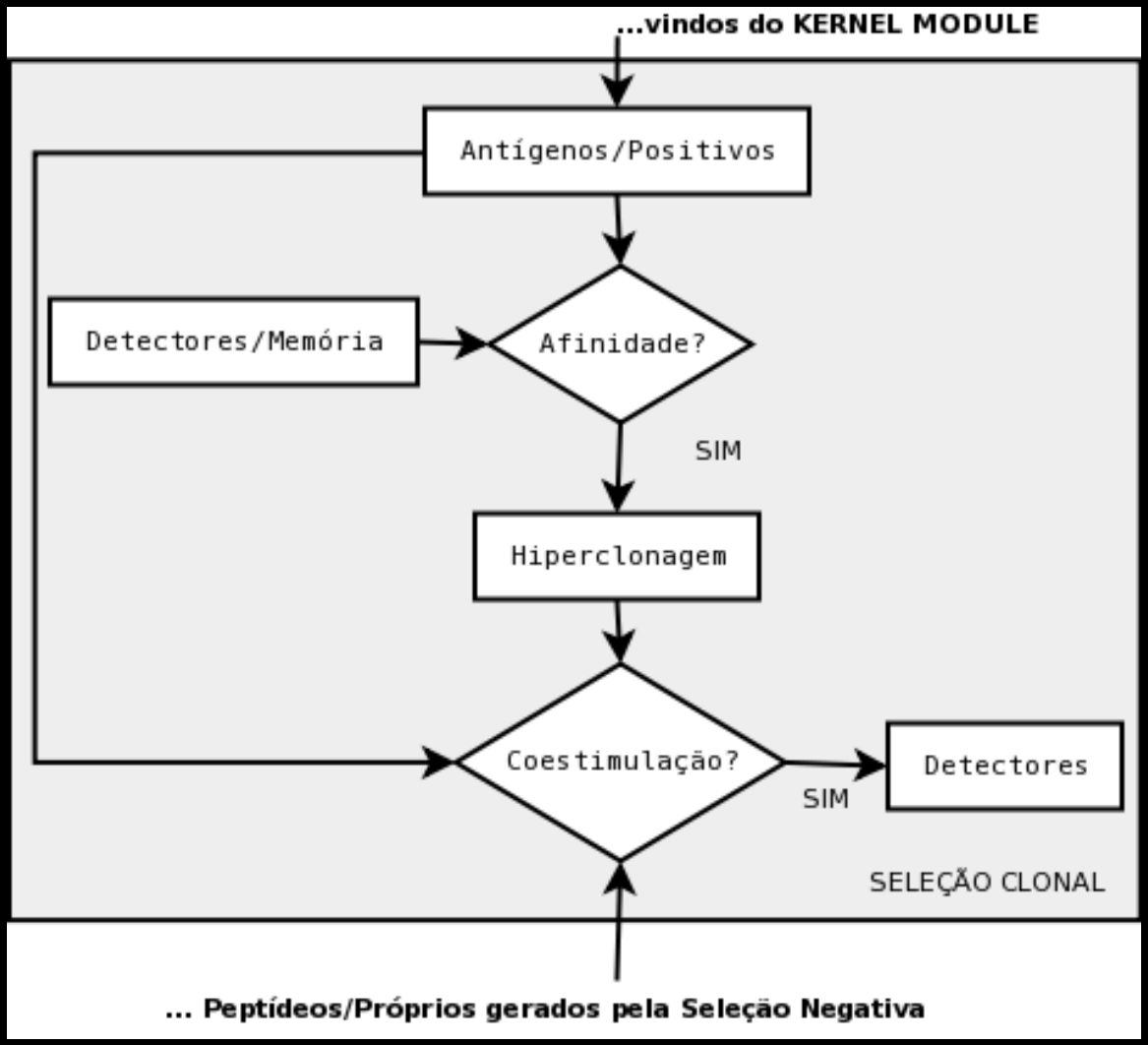
- utiliza *hooks* do Netfilter para capturar os pacotes da rede;
- o pacote *(possível Antígeno)* capturado é comparado com uma população de detectores *(Anticorpos)*;



- pacotes com grande afinidade com os detectores são enviados para análise;

Implementação - Definições

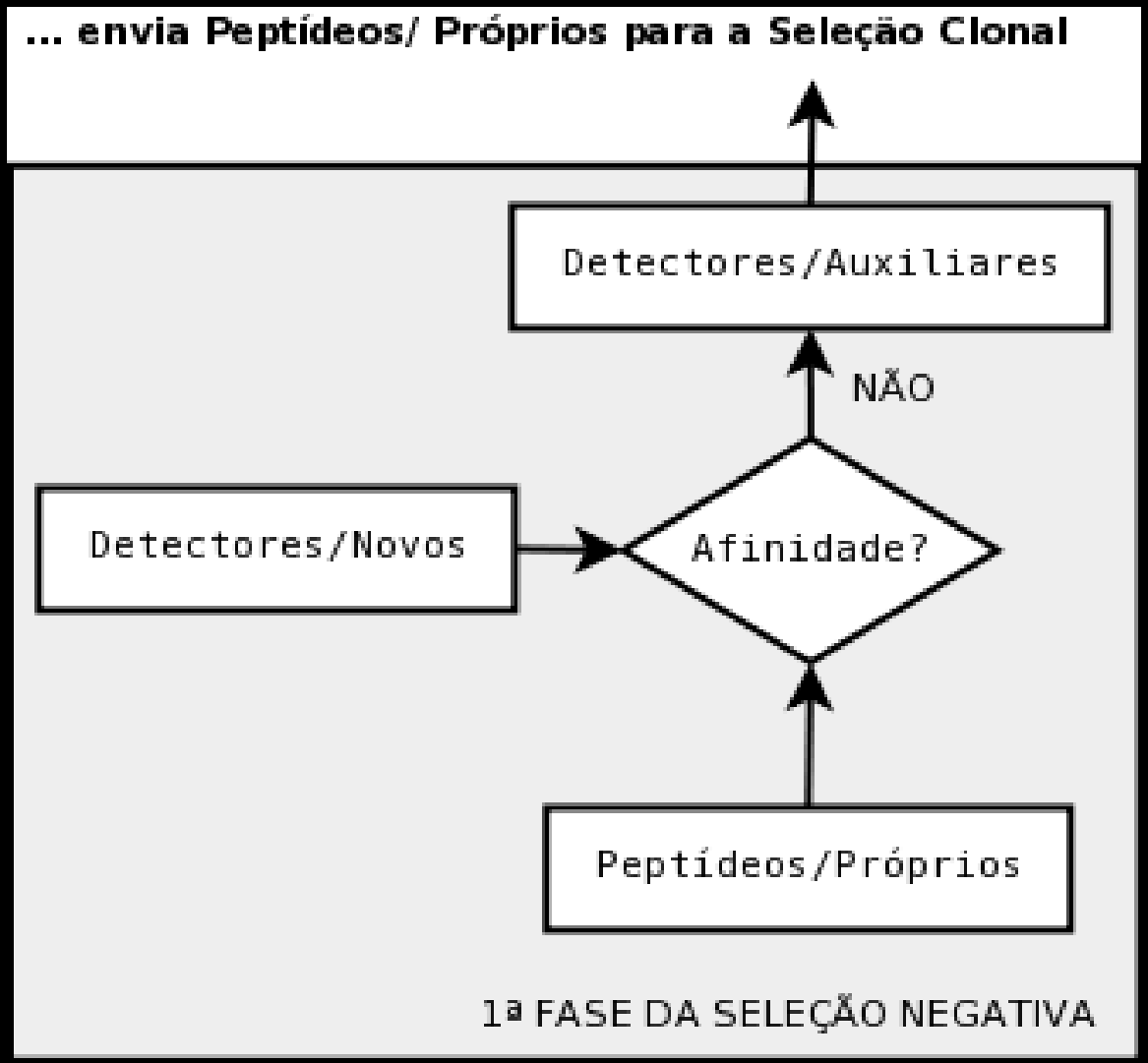
Programa Usuário (análise e geração)



- pacotes identificados como patógenos pelo módulo são analisados pelo algorítmo de Seleção Clonal;

Implementação - Definições

Programa Usuário (geração)



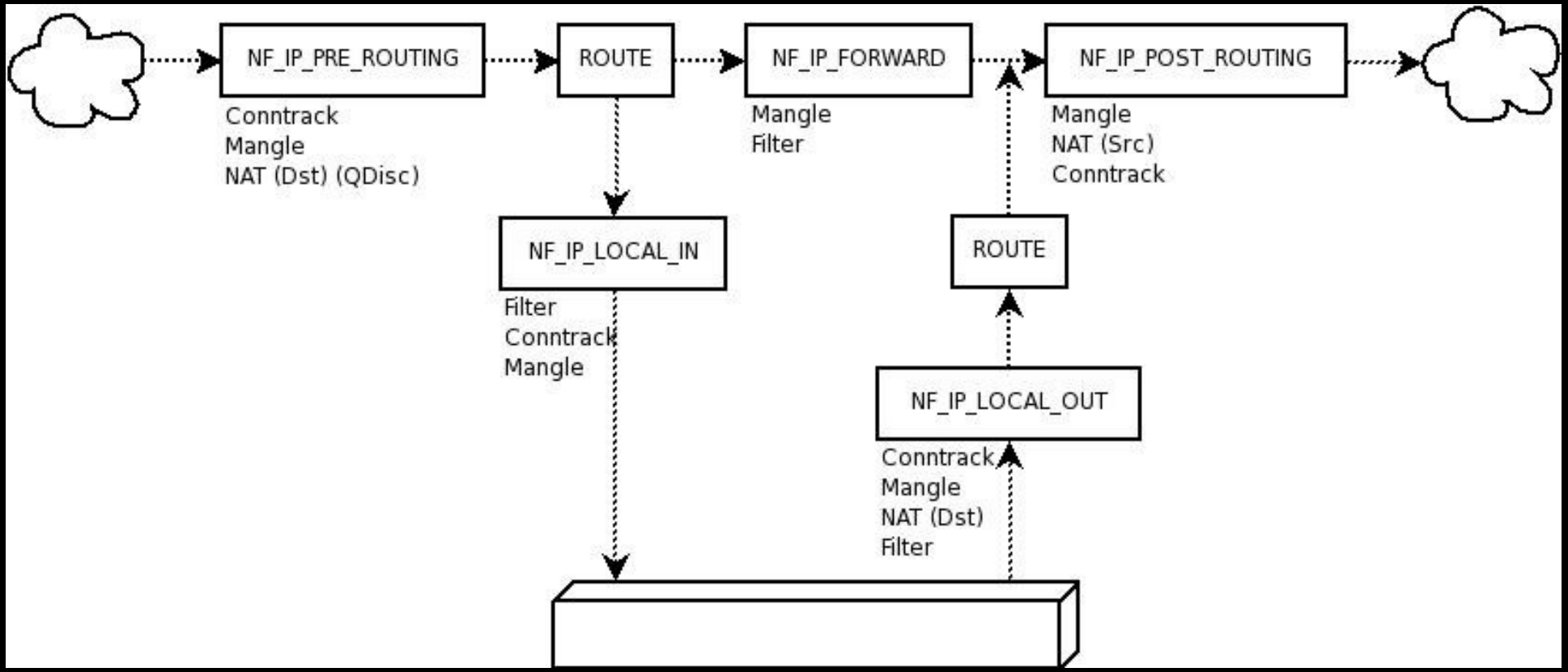
- o algorítmo de Seleção Negativa gera novos detectores com menor afinidade com o tráfego próprio.

Implementação - Módulos

- Extendem as funcionalidades do Kernel sem necessidade de reinicialização;
- Possibilitam Kernel's mais enxutos;

```
/* Inicia o modulo */
int __init init (void)
{
    int ret;
    fops.read = device_read;
    nfhin.hook = hook_in;
    nfhin.hooknum = NF_IP_PRE_ROUTING;
    ret = register_chrdev(MAJOR_NUM, DEVICE_NAME, &fops);
    if (ret < 0) {
        printk(KERN_ALERT "Falha ao registrar o char device - %d\n", ret);
        return ret;
    }
    nf_register_hook(&nfhin);
    return 0;
}
```

Implementação - Netfilter hooks



Implementação - Netfilter hooks

- **NF_ACCEPT**: Deixa o pacote passar.
- **NF_DROP**: Descarta o pacote.
- **NF_STOLEN**: “Esquece” do pacote.
- **NF_QUEUE**: Envia o pacote para o *userspace*.
- **NF_REPEAT**: Chama este *hook* novamente.

Implementação - Character Device

- São Devices que aceitam a escrita de um ou mais caracteres;
- Devices permitem a comunicação entre o kernel e os programas;

```
/* Funcao para escrever no char device */
static ssize_t device_write (struct file *file, const char *buffer,
                             size_t length, loff_t * offset)
{
    int bw; // bytes_write
    for (bw = 0; bw < length && bw < BUFFER_LEN; bw++)
         get_user (Mensagem[bw], buffer + bw);
    Mensagem_Ptr = Mensagem;
    return bw;
}
```

Implementação - IOCTL

- Permite o envio de comandos para um módulo através de um device file;

```
/* Escreve uma string no Char Device */
int Ioctl::WriteDevice(string &mensagem)
{
    int file_desc, ret_val;
    file_desc = open(nome, 0);
    if (file_desc < 0) {
        printf("Falha ao abrir o device: %s\n", nome);
        return 0;
    }
    ret_val = ioctl(file_desc, IOCTL_SET_MSG, mensagem.c_str());
    if (ret_val < 0) {
        printf("Falha ao escrever no device: %d\n", ret_val);
        return 0;
    }
    close(file_desc);
    return 1;
}
```

Simulações

- Port Scanner em servidor HTTP(S)

  nmap XMAS e NULL scans

- Seleção Negativa

  População inicial: Próprios

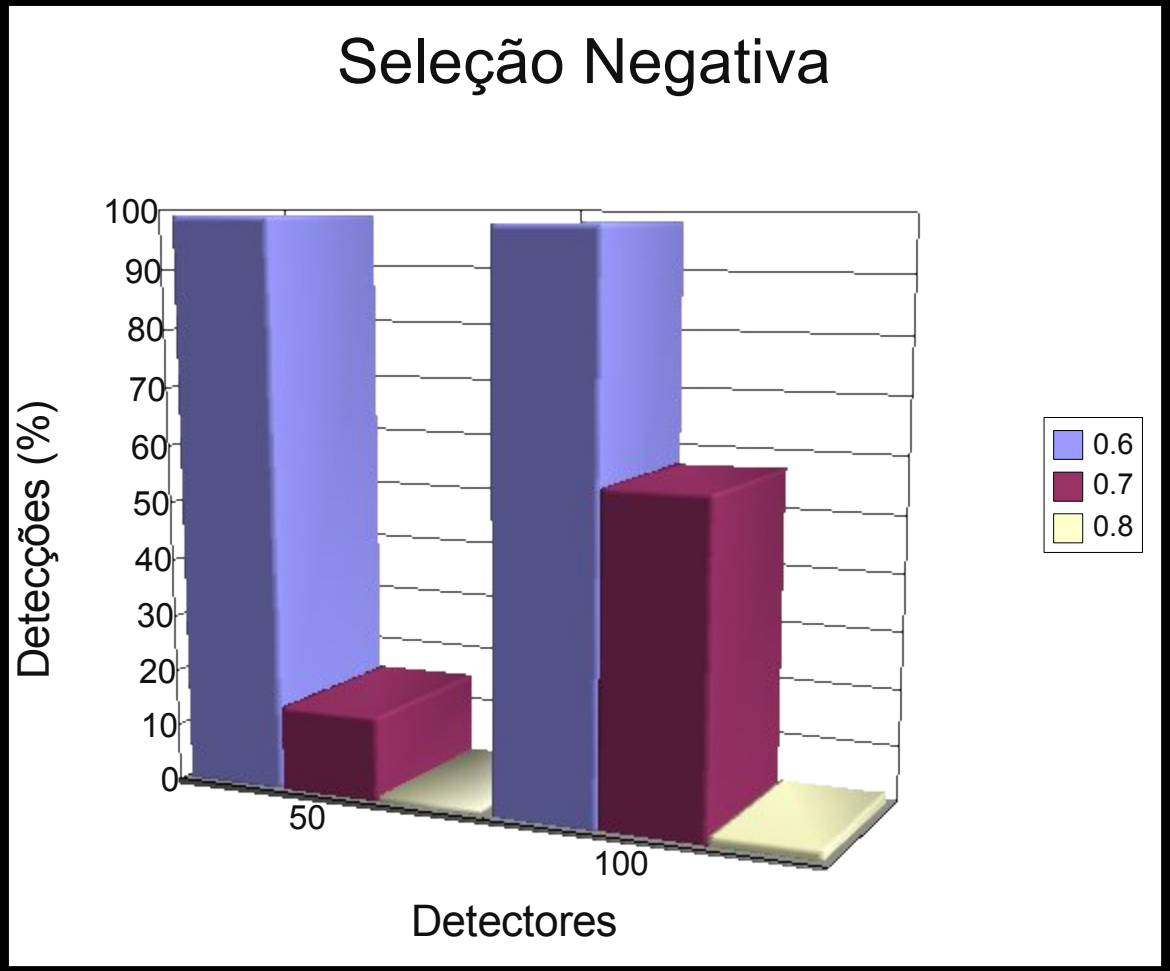
- Seleção Clonal

  População Inicial: Memória

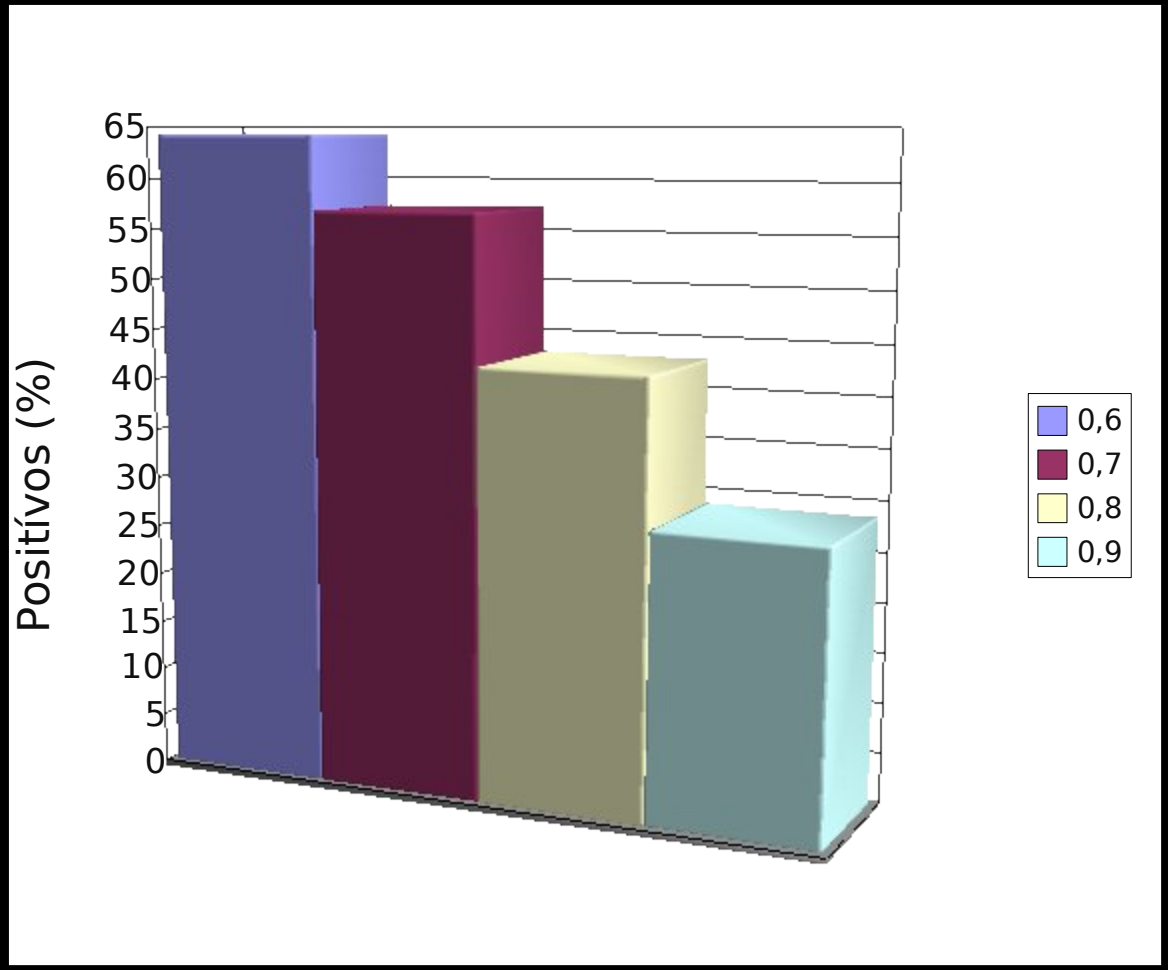
- Seleção Clonal com Seleção Negativa

  Populações Iniciais: Próprios e Memória

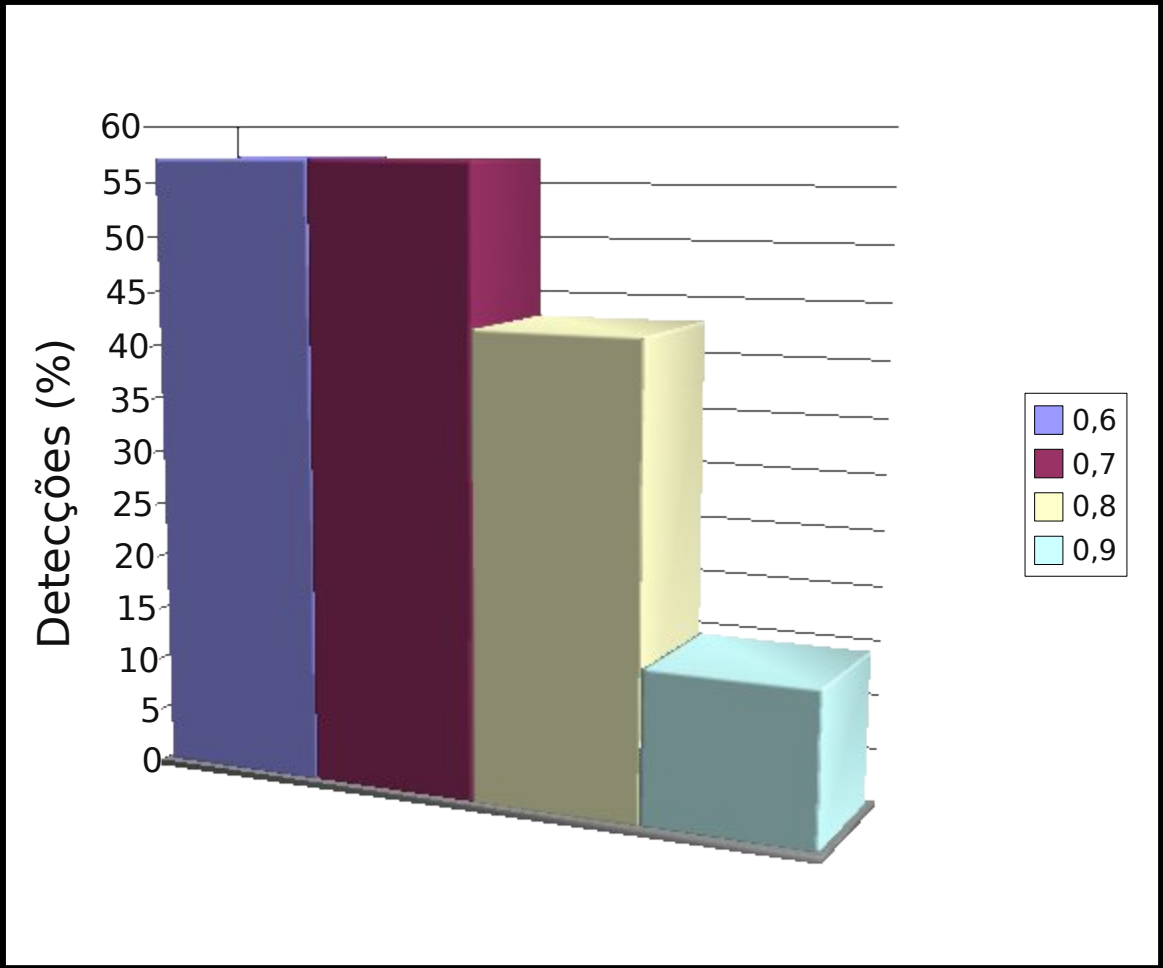
Simulações – Seleção Negativa



Simulações – Seleção Clonal



Simulações – Clonal com Negativa



Limitações

- Bibliotecas ANSI e outras?

  Infelizmente não. Podem ser utilizadas funções do kernel (/proc/kallsyms)

- rmmod?

  Habilitar o "-f" ao compilar o kernel.

- Debug?

  Ferramentas exóticas para debug.

- Outras?

  Memória...

Trabalhos Futuros

- Considerar conexão completa;
- Melhorar comunicação Programa -> Módulo;
- Utilizar threads e crontab no Programa;
- Implementar reações do IDS (reiniciar conexão, bloquear tráfego, etc...);
- Outras combinações dos Algoritmos Imunológicos;

Bibliografia

- [DE CASTRO et alli, 2000] DE CASTRO, L. N., VON ZUBEN, F. J. **The Clonal Selection Algorithm with Engineering Applications**. GECOO'00, Workshop on Artificial Immune System and Their Applications, pgs. 36-37; 2000.
- [D'HAESELEER et alli, 1996] D'haeseleer, P., Forrest, S., Helman, P. **An Immunological Approach to Change Detection: Algorithms, Analysis, and Implications**. In Proceedings of the 1996 IEEE Symposium on Computer Security and Privacy; 1996.
- [FORREST et alli, 1994] Forrest, S., Perelson, A.S., Allen, L., Cherukuri, R. **Self-Nonself Discrimination in a Computer**. In Proceedings of the 1994 IEEE Symposium on Research in Security and Privacy, Los Alamitos, CA: IEEE Computer Society Press; 1994.
- [FORREST et alli, 1997] Forrest, S., Hofmeyr, S.A., Somayaji, A. **Computer immunology**. Communications of the ACM, vol. 40, nº. 10, pgs. 88-96; 1997.

Bibliografia

- [HOFMEYR, 2000] HOFMEYR, S. **An Interpretative Introduction to the Immune System**. Department of Computer Science, University of New Mexico; Albuquerque, 2000.
- [JONES, 2006] JONES, M., T. **Access the Linux kernel using the /proc filesystem**, http://www-128.ibm.com/developerworks/linux/library/l-proc.html, IBM Developerworks Linux Zone, Março 2006.
- [JUNGWON, 2002] JUNGWON, K. W. **Integrating Artificial Immune Algorithms for Intrusion Detection**, PhD Thesis, Department of Computer Science, University College; London, 2002.
- [MUKHERJEE et alli, 1994] MUKHERJEE, B., HEBERLEIN, L. T., LEVITT, K. N. **Network Intrusion Detection**. IEEE Network Magazine, pgs. 26-41; Maio/Junho de 1994.

Bibliografia

- **Cross-Referencing Linux,** http://lxr.linux.no/
- **FreeBSD and Linux Kernel Cross-Reference**, http://fxr.watson.org/
- **KernelNewbies,** http://kernelnewbies.org/
- **Linux HeadQuarters**, http://www.linuxhq.com/
- **Linux/M32R Home Page**, http://www.linux-m32r.org/
- **The Linux Documentation Project**, http://www.tldp.org/
- **The netfilter.org project**, http://netfilter.org/
- **people.netfilter.org**, http://people.netfilter.org/
- **PHRACK**, http://www.phrack.org/archives/
- **whatisthekernel**, http://whatisthekernel.blogspot.com/
- **Eclipse CDT**, http://www.eclipse.org/cdt/

Dúvidas?! Sugestões?!

keep on hacking in the free world! ;)

oscarcosta@gmail.com